

*IBM API Connect 2018.x*

**IBM**





---

# Tables of Contents

<b>Welcome</b>	1
<b>API Connect overview</b>	2
What's new in the latest release (Version 2018.4.1.24)	3
What's new in Version 2018.4.1.20	4
What's new in Version 2018.4.1.19	4
What's new in Version 2018.4.1.17	4
What's new in Version 2018.4.1.16	5
What's new in Version 2018.4.1.15	5
What's new in Version 2018.4.1.13	6
Version 2018.4.1 known limitations	6
Available deployment options of API Connect	11
API Connect concepts	12
Packaging strategy and terminology in API Connect	12
Understanding rate limits for APIs and Plans	15
API Connect components	16
API Connect: End-to-end solution example	18
API Connect gateway types	20
API Connect user roles	20
API Connect support	24
API Connect glossary	25
Accessibility features for IBM API Connect	27
Legal information	28
License updates	28
Tracking API volume for auditing and compliance	28
Notices	29
Terms and conditions for information centers	30
IBM API Connect Considerations for GDPR Readiness	31
Essential reading	33
<b>Tutorials</b>	34
<b>Installing and maintaining your IBM API Connect cloud</b>	35
Kubernetes	36
Installing and upgrading on Kubernetes	36
IBM API Connect Version 2018 software product compatibility requirements	36
Estimating internal storage space for Analytics	37
Working with certificates	39
Certificate management: Read This First	39
Setting and managing certificates	40
Setting default certificates	40
Setting custom certificates	42
Replacing custom certificates	44
Setting common certificates	45
Setting the encryption-secret for the management database	45
Clearing certificates	46
Reference for certificates, commands, and validations	46
Certificate reference	46
Command reference	50
Validation reference	52
Tips and tricks for using APICUP	53
Deploying to a Kubernetes environment	56
Requirements for a Kubernetes deployment	56
Requirements for deploying IBM API Connect into a Kubernetes environment	56
Deployment overview for endpoints and certificates	59
Firewall requirements on Kubernetes	60
Kubernetes ingress controller prerequisites	63
Load balancer configuration in a Kubernetes deployment	64
Configuring logging for a Kubernetes deployment	66
Installing API Connect into a Kubernetes environment	66
First steps for installing API Connect: Upload files to registry	67
Installing the Management subsystem into a Kubernetes environment	69
Installing the Analytics subsystem into a Kubernetes environment	73
Installing the Developer Portal subsystem into a Kubernetes environment	77
Defining multiple portal endpoints for a Kubernetes environment	80
Installing the Gateway subsystem into a Kubernetes environment	80
Deploying pods to specific worker nodes in a multi-node cluster	83
Manually creating a CustomResourceDefinition in a Kubernetes environment	85
Creating an extra values file in a Kubernetes environment	86
Configuring user-defined policies on the API Gateway in a Kubernetes deployment	88
Installing the toolkit	90
Upgrading API Connect in a Kubernetes environment	91
Requirements for upgrading	92

Requirements for upgrading from v2018.4.1.8 or later	92
Requirements for upgrading from v2018.4.1.5, v2018.4.1.6, or v2018.4.1.7	93
Requirements for upgrading from Version 2018.4.1.4 or earlier	94
Upgrading API Connect subsystems	95
Adding a monitor pod during gateway upgrade	97
Verifying Portal upgrade	98
Troubleshooting the upgrade	98
Deleting the API Connect deployment in a Kubernetes runtime environment	100
Maintaining a Kubernetes deployment	100
Backing up and restoring	101
Backing up the management database in a Kubernetes environment	101
Restoring the management database in a Kubernetes environment	103
Troubleshooting restoration of management database	105
Backing up and restoring the Developer Portal in a Kubernetes environment	109
List of the Developer Portal exec commands	111
Backing up and restoring the analytics database	112
Adding certificates for Analytics for back-up and restore	115
Using APICUP to reconfigure	116
Converting installation mode	117
Converting dev mode to standard mode on Kubernetes	118
Converting standard mode to dev mode on Kubernetes	119
Increasing the memory allocation for the management database	119
Enabling Analytics ingestion-only on Kubernetes	120
Disabling Analytics ingestion-only on Kubernetes	121
Configuring maximum size of client requests to the Management subsystem	121
Increasing memory allocation for Cassandra operator on OpenShift	122
Setting rate limits for public APIs on the management service for a Kubernetes environment	122
Enabling high performance peering for DataPower API Gateway on Kubernetes	123
Dynamically re-registering and reconfiguring a Gateway service in a Kubernetes deployment	124
Monitoring and logging a Kubernetes deployment	125
Checking cluster health on Kubernetes	126
Determining status of a cluster on Kubernetes	126
Obtaining simple health check data of Developer Portal sites by using a REST API call	128
Gathering logs for a Kubernetes environment	129
Changing logging levels	129
Disabling the Analytics subsystem on Kubernetes	130
VMware	131
Installing and upgrading on VMware	132
IBM API Connect Version 2018 software product compatibility requirements	132
Estimating internal storage space for Analytics	132
Working with certificates	134
Certificate management: Read This First	135
Setting and managing certificates	135
Setting default certificates	136
Setting custom certificates	137
Replacing custom certificates	139
Setting common certificates	140
Setting the encryption-secret for the management database	141
Clearing certificates	141
Reference for certificates, commands, and validations	142
Certificate reference	142
Command reference	146
Validation reference	148
Tips and tricks for using APICUP	149
Deploying to a VMware environment	151
Requirements for deploying on VMware	152
Deployment overview for endpoints and certificates	152
Firewall requirements on VMware	154
Firewall enabled ports for clustered OVA deployments	156
Load balancer ports for clustered OVA deployments	157
Requirements for initial deployment on VMware	157
Load balancer configuration in a VMware deployment	160
Configuring remote logging for a VMware deployment	163
First steps for deploying in a VMware environment	164
Deploying the Management subsystem in a VMware environment	165
Deploying the Analytics subsystem in a VMware environment	171
Deploying the Developer Portal in a VMware environment	176
Defining multiple portal endpoints for a VMware environment	182
Access the Cloud Manager and begin API Connect Cloud Configuration	183
Configuring API Connect subsystems in a cluster on VMware	183
Installing the IBM License Metric Tool for VMware	187
Adding a static route on a virtual machine	187

Configuring use of an external NTP server	188
Deploying DataPower Gateway	189
Installing DataPower Gateway	189
Configuring DataPower Gateway for API Connect	190
Configuring DataPower API Gateway	190
Configuring DataPower Gateway (v5 compatible)	193
Sample configuration for multiple peering objects on gateway services external to Kubernetes	195
Installing the toolkit	90
Upgrading in a VMware environment	200
Requirements for upgrading on VMware	201
Requirements for upgrading from v2018.4.1.8 or later on VMware	201
Requirements for upgrading from v2018.4.1.7 on VMware	202
Requirements for upgrading from v2018.4.1.5 or v2018.4.1.6 on VMware	203
Requirements for upgrading from v2018.4.1.4 or earlier on VMware	204
Upgrading API Connect subsystems in a VMware environment	206
Control planes needed for upgrading to earlier fix packs	210
Troubleshooting the API Connect upgrade on VMware	214
Upgrading DataPower Gateway Service	214
Maintaining a VMware deployment	215
Backing up and restoring	216
Backing up the management subsystem in VMware environments	216
Restoring a management subsystem in a VMware environment	218
Troubleshooting restoration of management database on VMware	220
Backing up and restoring the Developer Portal in a VMware environment	224
List of the Developer Portal exec commands	226
Backing up and restoring the analytics database	227
Adding certificates for Analytics for back-up and restore on VMware	230
Disabling the Analytics subsystem on VMware	231
Using VM snapshots for infrastructure backup and disaster recovery	232
Using APICUP to reconfigure	233
Converting installation mode	234
Converting dev mode to standard mode on VMware	235
Converting standard mode to dev mode on VMware	236
Increasing the memory allocation for the management database	236
Enabling Analytics ingestion-only on VMware	237
Disabling Analytics ingestion-only on VMware	238
Setting rate limits for public APIs on the management service for a VMware environment	238
Dynamically re-registering and reconfiguring a Gateway service in a VMware deployment	239
Adding disk space to a VMware appliance	240
Running a filesystem check on a VMware root partition	241
Managing an appliance data disk	241
Troubleshooting the API Connect upgrade on VMware	214
Checking cluster health on VMware	243
Determining status of a cluster on VMware	243
Obtaining simple health check data of Developer Portal sites by using a REST API call	246
Monitoring the network with SNMP	246
Gathering logs for a VMware environment	247
Changing logging levels	248
IBM Cloud Private	249
Installing and upgrading on IBM Cloud Private	249
Deployment overview for endpoints and certificates	250
Deploying to an IBM Cloud Private environment	251
Deploying with the IBM Cloud Private catalog	251
Deploying with the Install Assist tool	254
Migrating an IBM Cloud Private catalog install to apicup	256
Installing the toolkit	90
Upgrading with IBM Cloud Private Catalog	258
Upgrading from Version 2018.4.1 with the IBM Cloud Private Catalog	258
Upgrading Version 2018.3.7 with the IBM Private Cloud Catalog	259
Backing up and restoring an IBM Cloud Private catalog deployment	259
Backing up and restoring the management database on IBM Cloud Private	259
Backing up and restoring the Developer Portal on IBM Cloud Private	260
OpenShift	261
Installing API Connect on OpenShift	261
Disabling the Analytics subsystem on OpenShift	263
IBM Cloud Pak for Integration	264
Backing up and restoring on Cloud Pak for Integration	265
Disabling the Analytics subsystem on Cloud Pak for Integration	265
Migrating a Version 5 deployment	266
Using the API Connect operations command line interface	266
Identify Services State	267
<b>Configuring and managing your server environment</b>	<b>267</b>

Activating your Cloud Manager user account	268
Accessing the Cloud Manager user interface	269
Defining your topology	270
Configuring the Management service	272
Creating an Availability Zone	272
Registering a gateway service	273
Registering an analytics service	275
Registering a portal service	276
Associating an analytics service with a gateway service	277
Setting visibility for a service	278
Managing authentication and security	279
User registries overview	279
Configuring an Authentication URL user registry	281
Configuring an LDAP user registry in the Cloud Manager	282
Using the CLI to configure a shared LDAP user registry	284
Configuring a Local User Registry	290
Configuring an OIDC user registry	291
Setting visibility for a user registry	295
Deleting a user registry	296
Removing a user from a user registry	297
Removing yourself from a user registry	297
Removing another user from a user registry	297
TLS profiles overview	298
Viewing TLS Profiles, Keystores, and Truststores	299
Creating a TLS Server Profile	300
Creating a TLS Client Profile	301
Setting visibility for a TLS Client Profile	302
Defining elliptic curve cryptographic schemes for a TLS client profile	302
Creating a Keystore	304
Creating a Truststore	304
Viewing certificate details and adding certificates to a keystore or truststore	305
Generating a self-signed certificate using OpenSSL	306
Generating a PKCS#12 file for Certificate Authority	307
Binding a TLS server profile to a gateway service	308
Updating the PKCS#12 certificate for a TLS server profile	308
OAuth Provider overview	309
Configuring a native OAuth provider	310
Configuring basic settings for a native OAuth provider	311
Configuring scopes for a native OAuth provider	312
Configuring user security for a native OAuth provider	313
Configuring tokens for a native OAuth provider	314
Configuring token management and revocation for a native OAuth provider	315
Configuring introspection for a native OAuth provider	316
Configuring metadata for a native OAuth provider	316
Configuring the OIDC parameters for a native OAuth provider	317
Editing a native OAuth provider by using the API Editor	318
Configuring a third-party OAuth provider	319
Setting the visibility for OAuth providers	320
OAuth concepts for API Connect	320
OAuth user scenario	321
OAuth introspection for third-party OAuth providers	321
OAuth metadata URL and authentication URL	323
Scope	326
Tokens	330
Refresh tokens	330
OAuth revocation URL	331
Authenticating and authorizing through a redirect URL	333
Token management with the DataPower Gateway (v5 compatible)	335
Token management with the DataPower API Gateway	337
Authentication URL	338
Custom forms for user security	339
Creating a custom HTML login form for user security	340
Creating a custom HTML authorization form for user security	341
Securing an API with a JSON Web Token	342
Troubleshooting OAuth	342
Configuring the cloud settings	343
Using the CLI to modify cloud settings	344
Change the name of your cloud	345
Configuring an email server for notifications	345
Setting up notifications	346
Customizing email notification templates	347
Configuring invitation timeouts	348
Configuring the password-reset notification and timeout	348
Selecting user registries for Cloud Manager and API Manager	349

Setting a default user registry for Cloud Manager and API Manager	350
Configuring timeouts for access tokens and refresh tokens	350
Viewing platform and UI endpoints	351
Configuring the default gateway services for Catalogs	351
Administering provider organizations	352
Creating a provider organization	352
Editing a provider organization	353
Deleting a provider organization	354
Viewing current provider organizations	354
Administering members and roles	355
Creating roles in the admin organization	356
Editing roles in the admin organization	356
Deleting roles in the admin organization	357
Viewing members and roles	357
Adding members to the admin organization	358
Assigning roles to members	359
Deleting a member	359
Role Defaults overview	360
Managing role defaults for provider organizations	360
Managing role defaults for consumer organizations	361
Monitoring the cloud	362
Counting total API calls for your Analytics service	362
Creating a script to count API calls	364
Customizing Analytics data with filters	365
Sample filters for customizing Analytics data	366
Configuring data retention and index rollover time periods	368
Configuring analytics offload for API Connect	369
Supported event types for analytics offload	370
Supported third-party systems for analytics offload	371
Providing a custom certificate for analytics offload	372
Configuring output plugins for analytics offload	373
Configuring the analytics message queue	376
Configuration files for analytics offload customization	376
Disabling access to analytics event data in API Connect	377
Troubleshooting your analytics offload	378
Configuring the audit log to track API calls	379
Configure the audit settings	381
Changing your Cloud Manager password and profile information	382
Resolving login problems by increasing HTTP header size	383
Extending the Gateway server behavior	384
Gateway extension guidelines - DataPower Gateway (v5 compatible)	384
Gateway extension guidelines - DataPower API Gateway	385
Configuring your Gateway server extensions	387
Cloud Manager Tutorials	388
Tutorial: Configuring the Cloud	389
Tutorial: Creating a Provider Organization	398
<b>Developing your APIs and applications</b>	<b>402</b>
Working with the toolkit	403
Installing the toolkit	90
Logging in from API Designer	404
Switching clouds from API Designer	406
Working offline with API Designer	406
Using the developer toolkit command-line tool	407
Overview of the command-line tool	407
Logging in to a management server	409
Cloud administration commands	410
API development and management commands	414
Creating APIs and applications	420
Publishing APIs and applications	421
Managing API Products	422
Working with Drafts	425
Reading input from the command line	425
Scripting with the toolkit commands	427
Working with OpenAPI extensions	428
Viewing application performance metrics	432
Viewing the application metrics dashboard	433
Viewing application metrics using third-party consoles	434
[Technical Preview] Searching for items in API Manager	436
Working with API definitions	439
Creating an API definition	439
Creating a REST proxy API from a target service	441
Creating a REST proxy API from an existing OpenAPI service	441
Creating a REST proxy API from an existing WSDL service	443
Creating a SOAP proxy API from an existing WSDL service	444
Creating a new REST OpenAPI definition	446

Adding a REST API by importing an OpenAPI definition file	446
Adding a SOAP API by importing a zip file	448
Editing an API definition	449
Defining Paths for a REST API	451
Configuring an operation	453
Organizing your APIs and Products into categories	454
Activity logging with the DataPower API Gateway	454
The Assemble view	455
Including components in your assembly	456
Adding components to your assembly	457
Handling errors in the assembly	457
Error cases supported by assembly catches	458
OpenAPI and assembly components	458
The behavior of an assembly	459
Creating a new version of an API definition	459
Specifying the gateway type for an API definition	460
Converting an API definition for deployment to the DataPower API Gateway	461
API properties	461
Setting API properties	466
Variable references in API Connect	467
Activating an API	470
Testing an API with the assembly test tool	471
Testing an API with the Local Test Environment	472
Staging an API	475
Publishing an API	476
Downloading an API definition	477
Deleting an API definition	478
Using an options file when importing a WSDL service	478
API policies and logic constructs	479
Built-in policies	480
Activity Log	481
Client Security	482
GatewayScript	483
GatewayScript code examples	484
Generate JWT	487
Validate JWT	489
Invoke	490
JSON to XML	494
Log	495
Map	496
The Map policy structure	496
Configuring the Map policy in the user interface	500
Map policy examples	503
OAuth	506
Example - using multiple OAuth policies in an OAuth provider assembly	508
Parse	511
Proxy	512
Rate Limit	514
Configuring a rate or burst limit on the DataPower API Gateway	514
Redaction - DataPower API Gateway	515
Constructing JSONata expressions to redact fields	516
Redaction - DataPower Gateway (v5 compatible)	518
Constructing XPath expressions to redact fields	519
Set Variable	521
User Security	522
Validate - DataPower API Gateway	524
Validate - DataPower Gateway (v5 compatible)	525
Validate Username Token	526
XML to JSON	528
XSLT	529
XSLT policy examples	530
Implementation code examples	533
Logic Constructs	538
if	538
operation-switch	539
switch	540
Using the switch policy condition editor	540
throw	543
Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway	543
Tags in API Connect	546
Configuring API security	547
Creating a security definition	548
Creating a basic authentication security definition	548

Creating an API key security definition	549
Creating an OAuth security definition	550
Applying security definitions to an API	551
Applying security definitions to an API operation	552
Enabling CORS support for an API	553
LDAP authentication	553
Authentication URL user registry	555
Working with Products	556
Creating a draft Product	557
Downloading a draft Product	559
Importing a draft Product	559
Editing a draft Product	560
Defining rate limits for an API operation	562
Staging a draft Product	563
Publishing a draft Product	564
Creating a new version of your Product	565
Specifying the gateway type for a Product	566
Deleting a draft Product	566
Creating and validating API and Product definitions by using the CLI	567
Creating an OpenAPI definition file	567
IBM extensions to the OpenAPI specification	568
execute	569
activity-log	571
client-security	572
gatewayscript	573
if	574
invoke	575
json-to-xml	578
jwt-generate	579
jwt-validate	581
log	582
map	583
operation-switch	587
oauth	589
Example - using multiple OAuth policies in an OAuth provider assembly	508
parse	593
proxy	595
ratelimit	596
redact - DataPower API Gateway	597
redact - DataPower Gateway (v5 compatible)	598
set-variable	600
switch	601
Writing switch condition scripts - DataPower API Gateway	602
Writing switch condition scripts - DataPower Gateway (v5 compatible)	604
throw	605
user-security	606
validate - DataPower API Gateway	608
validate - DataPower Gateway (v5 compatible)	610
validate-username token	611
xml-to-json	612
xslt	613
catch	613
properties	614
catalogs	615
activity-log	615
Specifying a gateway type for an API definition	616
Referring to an extension in an API definition	617
Using \$ref to reuse code fragments in your OpenAPI files	618
Creating a Product definition file	619
Product definition schema	620
Completing the information section of your Product description	627
Specifying a gateway type for your Product	629
Converting a Product YAML file to use DataPower API Gateway	629
Specifying the visibility of your Product	630
Referencing the APIs for your Product	631
Describing Plans in your Product	632
An example YAML representation of a Product	634
Using x-ibm-languages to create multilingual API and Product documentation	635
Using x-example to control the examples displayed in the Developer Portal	638
Using x-embedded-doc to add additional documentation to products and APIs	638
Using x-pathaliases to give consistent URLs for products and APIs	641
Validating the YAML or JSON definition of an API or Product	642
Creating and using API and Product definitions templates	642
OpenAPI 3.0 support in IBM API Connect	643

Working with global policies	644
Reference	646
API Connect context variables	646
OAuth context variables	649
API and Product definition template examples	652
Template variables for API and Product definitions	658
<b>Managing your APIs</b>	<b>659</b>
Activating your API Manager user account	660
Accessing the API Manager user interface	661
[Technical Preview] Searching for items in API Manager	662
Working with Catalogs	665
Creating and configuring Catalogs	666
Configuring sub paths for Developer Portal sites	669
Managing Catalog membership	670
Importing a user-defined policy into a Catalog	671
Using syndication in API Connect	672
Enabling Spaces in a Catalog	673
Creating, modifying, and deleting Spaces	674
Working with Spaces	674
Managing Products in a Space	675
Managing subscription requests in a Space	675
Managing application subscriptions in a Space	676
Managing Space membership	676
Managing user access in a Space	677
Managing Gateways in a Space	677
Administering Consumer organizations	678
Creating a Consumer organization	678
Editing a Consumer organization	679
Deleting a Consumer organization	680
Working with Consumer organization groups	680
Managing developer applications	681
Security and authentication	682
Creating a TLS client profile	682
Creating a Keystore	683
Creating a Truststore	684
Generating a PKCS#12 file for Certificate Authority	685
Generating a self-signed certificate using OpenSSL	686
Viewing certificate details and adding certificates to a keystore or truststore	687
Defining elliptic curve cryptographic schemes for a TLS client profile	687
Authenticating by using your enterprise user registry	688
Working with user registries	689
Creating an LDAP user registry in API Manager	689
Using the CLI to create an organization-specific LDAP user registry	692
Creating an Authentication URL user registry	696
Creating a Local User Registry	698
Creating an OIDC user registry	698
Modifying the configuration details for a user registry	703
Password lockout criteria	703
Configuring a native OAuth provider	703
Configuring basic settings for a native OAuth provider	705
Configuring scopes for a native OAuth provider	706
Configuring user security for a native OAuth provider	707
Configuring tokens for a native OAuth provider	708
Configuring token management and revocation for a native OAuth provider	709
Configuring introspection for a native OAuth provider	709
Configuring metadata for a native OAuth provider	710
Configuring the OIDC parameters for a native OAuth provider	711
Editing a native OAuth provider by using the API Editor	711
Configuring a third-party OAuth provider	712
OAuth concepts for API Connect	320
OAuth user scenario	321
OAuth introspection for third-party OAuth providers	321
OAuth external URL and authentication URL	323
Scope	326
Tokens	330
Refresh tokens	330
Token revocation	331
Authenticating and authorizing through a redirect URL	333
Token management with the DataPower Gateway (v5 compatible)	335
Token management with the DataPower API Gateway	337
Authentication URL user registry	338
Custom forms for user security	339
Creating a custom HTML login form for user security	340
Creating a custom HTML authorization form for user security	341



Securing an API with a JSON Web Token	342
Troubleshooting OAuth	342
Working with Products in the API Manager	736
The Product lifecycle	736
Managing your Products	737
Publishing a Product	738
Publishing a new Product	739
Replacing a Product with another Product	740
Superseding a Product with another Product	741
Changing the availability of a Product	743
Deprecating a Product	743
Retiring a Product	744
Re-staging a Product	745
Updating the gateway services for a Product	746
Removing a Product from a Catalog	746
Approving Product lifecycle and subscription requests	747
Managing the APIs in a Product	748
API Analytics	749
Catalogs, Spaces, and Analytics	749
Accessing analytics	752
The applications landing page	752
Dashboard application page	753
Default dashboards	755
Cloning a dashboard	756
Creating custom dashboards	756
Importing dashboards	757
Editing dashboards	758
Exporting dashboards	759
Backing up and restoring dashboards	759
Deleting custom dashboards	760
Visualization application page	760
Visualizations	762
Applying filters to change the sampling of data displayed in visualizations	763
Creating visualizations	765
Configuring visualizations	766
Editing visualizations	767
Importing visualizations	768
Exporting visualizations	768
Backing up and restoring visualizations	769
Deleting custom visualizations	769
Working with searches (Discover)	770
Creating searches	771
Editing searches	773
Exporting searches	776
Importing searches	777
Backing up and restoring searches	779
Deleting searches	779
Exporting API event data	780
Using the UI to export API event data	781
Exporting aggregated analytics data from a visualization	781
Using REST APIs to export events data to JSON or YAML	783
API event record fields	783
Troubleshooting Analytics start-up	790
Administering user access	791
Viewing your user information	791
Adding provider organization users and assigning roles	791
Creating custom roles	792
Removing a user from a provider organization	793
Changing your API Manager password	793
Resolving login problems by increasing HTTP header size	794
Reference	795
API gateway response codes	795
API Manager tutorials	795
Tutorial: Creating a proxy REST API definition	796
Tutorial: Creating a SOAP API	805
Tutorial: Creating a REST API definition that invokes an existing SOAP service	811
Tutorial: Mapping JSON Content	818
Tutorial: Importing an API	836
Tutorial: Supersede a Product	841
Tutorial: Implementing OAuth Security	849
Tutorial: Implementing OpenID Connect Security	866
Tutorial: Generate a JSON Web Token (JWT)	878
Tutorial: Validate a JSON Web Token (JWT)	888
Tutorial: Creating a Client Application	899

<b>Authoring policies</b>	<b>903</b>
Authoring policies for the DataPower Gateway (v5 compatible)	904
Describing your policy	904
Creating a new user-defined policy	907
Implementing your policy	908
Packaging and importing your policies into IBM API Connect	910
Reference	911
Implementation code examples	911
Authoring policies for the DataPower API Gateway	915
Defining and packaging your user-defined policy for the DataPower API Gateway	915
Publishing your user-defined policy to the DataPower API Gateway	917
<b>Developer Portal: socialize your APIs</b>	<b>918</b>
Using the Developer Portal	918
Exploring APIs and Products in the Developer Portal	919
Testing an API by using the Developer Portal test tool	919
Calling an API	920
Calling an API by using CORS	921
Applications in the Developer Portal	922
Registering an application	922
Managing applications	923
Editing an application	924
Deleting an application	924
Changing an application image	924
Verifying an application client secret	925
Analytics in the Developer Portal	925
Application analytics in the Developer Portal	925
Organization analytics in the Developer Portal	926
Consumer organizations in the Developer Portal	926
Adding a Consumer organization from within the Developer Portal	926
Editing the name of a Consumer organization	927
Switching Consumer organizations	927
Adding users to a Consumer organization	927
Removing a user from a Consumer organization	928
Changing the ownership of a Consumer organization	928
Deleting a Consumer organization	929
User accounts, passwords, and support in the Developer Portal	930
Creating a new developer account	931
Changing your account settings	931
Deleting your developer account	931
Configuring the Developer Portal site	932
Concepts in the Developer Portal	933
Getting started configuring the Developer Portal site	934
Appearance	936
Controlling general appearance elements in the Developer Portal	936
Changing the shortcut icon	936
Changing the site email address	937
Changing the site logo	937
Changing the site name or site slogan	938
Changing the visibility of site branding	938
Adding custom JavaScript to a custom theme	939
Applying a modified content type template	939
Creating a sub-theme	940
Installing additional themes	942
Deleting themes	942
Content	943
Adding content elements in the Developer Portal	943
Adding and configuring meta tags	944
Disabling external search engine indexing	944
Adding custom pages	945
Integrating Twitter data into the social block	945
Editing the social block	946
Adding content in multiple languages	946
Adding custom pages to APIs and Products	947
Adding new frequently asked questions	948
Applying an image to an API or Product	948
Attaching a documentation file to APIs	949
Configuring blogs	949
Configuring the taxonomy menu block	950
Customizing the URL alias for a specific API or Product page	950
Embedding multimedia in site content	951
Linking from one piece of site content to another	951
Linking to social media sites	951
Managing tags in the Developer Portal	952

Tagging APIs based on their lifecycle phase	952
Posting a poll	953
Adding images to site content	953
Configuring and restricting content in the Developer Portal	954
Creating content types	954
Adding fields to content types	955
Configuring documentation upload limitations	955
Configuring image upload limitations	956
Configuring the appearance of ratings in the Developer Portal	956
Customizing the privacy policy statement	957
Customizing the terms of use statement	957
Editing a site comment	958
Editing sample content	958
Editing tags for a specific item	959
Turning content on or off in the Developer Portal	959
Deleting a site comment	960
Deleting blocks	960
Disabling blogs	960
Deleting forums from the front page	961
Turning comments for specific content types on or off in the Developer Portal	962
Turning comments off for an individual item	962
Turning off ratings for specific content types in the Developer Portal	963
Turning the ability to tag specific content types off in the Developer Portal	963
Structure	963
Configuring the front page	964
Disabling blogs	960
Deleting forums from the front page	961
Adding a menu	966
Adding and changing the blocks displayed on Developer Portal pages	967
Changing the front page banner block	968
Changing the front page Featured Content block	969
Changing the items in the main menu	969
Changing the ratings display	970
Displaying APIs and Products in categories	972
Enabling dynamic category creation	972
Implementing an image carousel	973
Making your application's image public	974
Providing navigation by tag hierarchy	974
Using the Views module in the Developer Portal	975
Creating views in the Developer Portal	975
Configuring the search results view in the Developer Portal	976
Configuring the default number of items in a view list in the Developer Portal	977
Configuration	977
Configuring content moderation	977
Editing, reviewing, and publishing moderated content	980
General configuration tasks	980
Adding a page load progress indicator	981
Adding custom fields to user records	982
Checking the site status report	982
Clearing the server caches	983
Configuring a proxy	983
Configuring cron to run scheduled tasks	984
Configuring search indexes and servers	984
Configuring the date and time	985
Configuring the settings of a Consumer organization	985
Configuring the site default timezone	986
Configuring the site error handling	986
Configuring which buttons are displayed in the WYSIWYG rich text editor	987
Configuring which languages are available	987
Customizing user account settings	988
Disabling languages	988
Disabling test tool restrictions	990
Enabling a cookie compliance banner	990
Enabling code languages for code snippets	991
Enabling code snippets for SOAP APIs	991
Forcing new users to accept terms and conditions	992
Hiding the admin registry on the login form	993
Hiding the certificate in the header for APIs secured with mutual TLS	994
Importing and exporting taxonomies	994
Restricting access by IP address	996
Restricting or preventing access from search engines	998
Toggling the site in and out of maintenance mode	999

Viewing available updates	999
Managing Developer Portal security	1000
Configuring CAPTCHA	1000
Configuring reCAPTCHA	1001
Configuring session timeout and limit	1002
Configuring the timeout length for the password reset link	1002
Configuring your site password policy	1003
Disabling CORS warnings	1003
Disabling live testing of APIs	1004
How to enable penetration testing, and information about Developer Portal cookies	1004
How to manage IP security in the Developer Portal	1006
Managing banned IP addresses	1007
Using flood control for login security	1007
Login security	1008
Using the Security kit	1008
Using Honeypot for spam protection	1009
People	1010
Working with roles in the Developer Portal	1010
Creating administrator users for the Developer Portal	1010
Assigning users to a role	1011
Assigning permissions to a role	1011
Creating a new role	1012
Blocking and unblocking specific users	1012
Controlling access to Developer Portal content	1013
Creating a developer account to customize API properties	1013
Forums	1014
Creating a new forum	1015
Creating new forum containers	1015
Configuring the creation of new forums for each API	1015
Locking forum topics	1016
Marking content in a forum as sticky	1016
Removing forum posts	1016
Turning off forums in the Developer Portal	1017
Reports	1017
Extend	1018
Custom module development: an introduction to Drupal tools for PHP development	1018
Custom module development: background and prerequisites	1019
Custom module development: creating a module skeleton	1020
Custom module development: using hooks	1022
Installing custom modules	1024
Deleting custom modules	1024
Disabling modules	1025
Troubleshooting guide for the Developer Portal	1026
Setting the security mode of the Portal Director	1029
Migrating your Developer Portal from Version 5 to Version 2018	1030
Developer Portal tutorials	1032
Tutorial: Creating the Developer Portal	1032
Tutorial: Creating a private Developer Portal site	1036
Tutorial: Creating Accounts and Applications on the Developer Portal	1037
Tutorial: Creating a custom theme for the Developer Portal	1044
Tutorial: Using avatars	1046
Tutorial: Using image optimization	1049
Tutorial: Adding a Back to top button	1051
Tutorial: Adding a zoom effect	1052
Tutorial: Creating a drop-down menu link	1054
Tutorial: Displaying blog posts on the front page	1058
Tutorial: Displaying tweets on the front page	1061
Tutorial: Changing the front page of your Developer Portal	1067
Tutorial: Grouping products by category	1073
Tutorial: Adding a field to the sign up form	1077
Tutorial: Adding validation to a field on the sign-up form	1079
Tutorial: Adding a field group to group fields in a content type	1081
Tutorial: Adding a custom block to the front page	1083
Tutorial: Adding a custom block to a page other than the front page	1086
Tutorial: Configuring the RobotsTxt file	1088
Tutorials for using custom modules in the Developer Portal	1090
Tutorial: Changing the profile page to display firstname lastname, instead of username	1090
Tutorial: Adding validation to a field on the sign-up form	1079
Tutorial: Using a custom weighting sort order on the product list page	1094
<b>Reference information</b>	<b>1097</b>
Toolkit command-line tool reference	1097
apic activations	1098
apic activations:clear	1098
apic activations:delete	1098

apic activations:get	1099
apic activations:list	1099
apic analytics	1100
apic analytics:create	1100
apic analytics-services	1101
apic analytics-services:clear	1101
apic analytics-services:create	1102
apic analytics-services:delete	1102
apic analytics-services:get	1103
apic analytics-services:list	1103
apic analytics-services:update	1104
apic api-keys	1104
apic api-keys:create	1104
apic api-keys:delete	1105
apic api-keys:get	1105
apic api-keys:list	1106
apic apis	1106
apic apis:clone	1107
apic apis:document	1107
apic apis:get	1108
apic apis:list	1108
apic apis:list-all	1109
apic apis:update	1109
apic apis:wSDL	1110
apic apps	1110
apic apps:clear	1111
apic apps:create	1111
apic apps:delete	1112
apic apps:get	1112
apic apps:list	1112
apic apps:update	1113
apic associates	1113
apic associates:get	1114
apic associates:list	1114
apic availability-zones	1115
apic availability-zones:clear	1115
apic availability-zones:create	1116
apic availability-zones:delete	1116
apic availability-zones:get	1117
apic availability-zones:list	1117
apic availability-zones:update	1118
apic billings	1118
apic billings:clear	1119
apic billings:create	1119
apic billings:delete	1119
apic billings:get	1120
apic billings:list	1120
apic billings:update	1121
apic catalog-settings	1121
apic catalog-settings:get	1122
apic catalog-settings:update	1122
apic catalogs	1122
apic catalogs:clear	1123
apic catalogs:create	1123
apic catalogs:delete	1124
apic catalogs:get	1124
apic catalogs:list	1125
apic catalogs:transfer-owner	1125
apic catalogs:update	1126
apic client-creds	1126
apic client-creds:clear	1126
apic client-creds:list	1127
apic client-creds:set	1127
apic cloud-settings	1128
apic cloud-settings:about	1128
apic cloud-settings:audit-endpoint-test-connection	1129
apic cloud-settings:designer-credentials-list	1129
apic cloud-settings:get	1130
apic cloud-settings:info	1130
apic cloud-settings:mail-server-configured	1130
apic cloud-settings:oauth2-certs	1131
apic cloud-settings:toolkit-credentials-list	1131
apic cloud-settings:topology	1132

apic cloud-settings:update	1132
apic config	1132
apic config:clear	1133
apic config:delete	1133
apic config:get	1134
apic config:list	1135
apic config:set	1136
apic configured-api-user-registries	1137
apic configured-api-user-registries:clear	1137
apic configured-api-user-registries:create	1138
apic configured-api-user-registries:delete	1138
apic configured-api-user-registries:get	1139
apic configured-api-user-registries:list	1139
apic configured-billings	1140
apic configured-billings:clear	1140
apic configured-billings:create	1141
apic configured-billings:delete	1141
apic configured-billings:get	1141
apic configured-billings:list	1142
apic configured-catalog-user-registries	1142
apic configured-catalog-user-registries:create	1143
apic configured-catalog-user-registries:delete	1143
apic configured-catalog-user-registries:get	1144
apic configured-catalog-user-registries:list	1144
apic configured-catalog-user-registries:search	1145
apic configured-gateway-services	1145
apic configured-gateway-services:clear	1146
apic configured-gateway-services:create	1146
apic configured-gateway-services:delete	1146
apic configured-gateway-services:get	1147
apic configured-gateway-services:list	1147
apic configured-oauth-providers	1148
apic configured-oauth-providers:clear	1148
apic configured-oauth-providers:create	1149
apic configured-oauth-providers:delete	1149
apic configured-oauth-providers:get	1150
apic configured-oauth-providers:list	1150
apic configured-tls-client-profiles	1151
apic configured-tls-client-profiles:clear	1151
apic configured-tls-client-profiles:clear-all	1152
apic configured-tls-client-profiles:create	1152
apic configured-tls-client-profiles:delete	1153
apic configured-tls-client-profiles:get	1153
apic configured-tls-client-profiles:list	1154
apic configured-tls-client-profiles:list-all	1154
apic consumer-org-settings	1155
apic consumer-org-settings:delete	1155
apic consumer-org-settings:get	1155
apic consumer-org-settings:update	1156
apic consumer-orgs	1156
apic consumer-orgs:clear	1157
apic consumer-orgs:create	1157
apic consumer-orgs:delete	1158
apic consumer-orgs:get	1158
apic consumer-orgs:list	1159
apic consumer-orgs:transfer-owner	1159
apic consumer-orgs:update	1160
apic create	1160
apic create:api	1161
apic create:product	1162
apic credentials	1162
apic credentials:clear	1163
apic credentials:create	1163
apic credentials:delete	1164
apic credentials:get	1164
apic credentials:list	1165
apic credentials:reset-client-secret	1165
apic credentials:reset	1166
apic credentials:update	1166
apic credentials:verify-client-secret	1167
apic draft-apis	1167
apic draft-apis:clear-all	1168
apic draft-apis:clear	1168

apic draft-apis:clone	1169
apic draft-apis:create	1169
apic draft-apis:delete	1170
apic draft-apis:document	1170
apic draft-apis:get	1170
apic draft-apis:list	1171
apic draft-apis:list-all	1171
apic draft-apis:update	1172
apic draft-apis:validate	1172
apic draft-apis:wSDL	1173
apic draft-products	1173
apic draft-products:clear-all	1174
apic draft-products:clear	1174
apic draft-products:clone	1175
apic draft-products:create	1175
apic draft-products:delete	1175
apic draft-products:document	1176
apic draft-products:get	1176
apic draft-products:list-all	1177
apic draft-products:list	1177
apic draft-products:publish	1178
apic draft-products:update	1178
apic draft-products:validate	1179
apic drafts	1179
apic drafts:clear	1180
apic drafts:list	1180
apic entries	1180
apic entries:clear	1181
apic entries:create	1181
apic entries:delete	1182
apic entries:get	1182
apic entries:list	1183
apic entries:update	1183
apic extensions	1184
apic extensions:clear-all	1184
apic extensions:clear	1185
apic extensions:clone	1185
apic extensions:create	1186
apic extensions:delete	1186
apic extensions:document	1187
apic extensions:get	1187
apic extensions:list	1188
apic extensions:list-all	1188
apic extensions:update	1189
apic gateway-extensions	1189
apic gateway-extensions:create	1190
apic gateway-extensions:delete	1190
apic gateway-extensions:get	1191
apic gateway-extensions:implementation	1191
apic gateway-extensions:update	1192
apic gateway-services	1192
apic gateway-services:clear	1192
apic gateway-services:create	1193
apic gateway-services:delete	1193
apic gateway-services:get	1194
apic gateway-services:list	1194
apic gateway-services:reset-oauth-secret	1195
apic gateway-services:update	1195
apic global-policies	1196
apic global-policies:clear	1196
apic global-policies:clear-all	1197
apic global-policies:create	1197
apic global-policies:delete	1198
apic global-policies:document	1198
apic global-policies:get	1199
apic global-policies:list	1199
apic global-policies:list-all	1200
apic global-policies:update	1200
apic global-policy-posthooks	1201
apic global-policy-posthooks:create	1201
apic global-policy-posthooks:delete	1201
apic global-policy-posthooks:get	1202

apic global-policy-posthooks:update	1202
apic global-policy-prehooks	1203
apic global-policy-prehooks:create	1203
apic global-policy-prehooks:delete	1204
apic global-policy-prehooks:get	1204
apic global-policy-prehooks:update	1205
apic groups	1205
apic groups:clear	1206
apic groups:create	1206
apic groups:delete	1207
apic groups:get	1207
apic groups:list	1207
apic groups:update	1208
apic identity-providers	1208
apic identity-providers:list	1209
apic integrations	1209
apic integrations:clear	1210
apic integrations:create	1210
apic integrations:delete	1211
apic integrations:get	1211
apic integrations:list	1212
apic integrations:list-all	1212
apic integrations:update	1212
apic invitations	1213
apic invitations:clear	1213
apic invitations:create	1214
apic invitations:delete	1214
apic invitations:get	1215
apic invitations:list	1215
apic invitations:update	1216
apic jobs	1216
apic configured-billings:clear	1140
apic jobs:delete	1217
apic jobs:get	1218
apic jobs:list	1218
apic jobs:retry	1218
apic keystores	1219
apic keystores:clear	1219
apic keystores:create	1220
apic keystores:delete	1220
apic keystores:get	1221
apic keystores:list	1221
apic keystores:update	1222
apic lb4	1222
apic licenses	1222
apic log-spec	1223
apic log-spec:get	1223
apic log-spec:update	1224
apic login	1224
apic logout	1225
apic mail-servers	1225
apic mail-servers:clear	1226
apic mail-servers:create	1226
apic mail-servers:delete	1226
apic mail-servers:get	1227
apic mail-servers:list	1227
apic mail-servers:test-connection	1228
apic mail-servers:update	1228
apic me	1229
apic me:change-password	1229
apic me:delete	1229
apic me:get	1230
apic me:update	1230
apic member-invitations	1231
apic member-invitations:clear	1231
apic member-invitations:create	1232
apic member-invitations:delete	1232
apic member-invitations:get	1233
apic member-invitations:list	1233
apic member-invitations:update	1234
apic members	1234
apic members:clear	1235
apic members:create	1235



apic members:delete	1236
apic members:get	1236
apic members:list	1237
apic members:update	1237
apic notification-templates	1238
apic notification-templates:get	1238
apic notification-templates:list	1239
apic notification-templates:list-all	1239
apic notification-templates:update	1240
apic oauth-providers	1240
apic oauth-providers:clear	1240
apic oauth-providers:create	1241
apic oauth-providers:delete	1241
apic oauth-providers:get	1242
apic oauth-providers:list	1242
apic oauth-providers:update	1243
apic org-settings	1243
apic org-settings:get	1244
apic org-settings:update	1244
apic orgs	1244
apic orgs:clear	1245
apic orgs:create	1245
apic orgs:delete	1246
apic orgs:get	1246
apic orgs:list	1247
apic orgs:transfer-owner	1247
apic orgs:update	1247
apic payment-methods	1248
apic payment-methods:create	1248
apic payment-methods:delete	1249
apic payment-methods:get	1249
apic payment-methods:list	1250
apic payment-methods:update	1250
apic permissions	1251
apic permissions:get	1251
apic permissions:list	1252
apic permissions:list-all	1252
apic policies	1253
apic policies:clear	1253
apic policies:clear-all	1254
apic policies:clone	1254
apic policies:create	1255
apic policies:delete	1255
apic policies:document	1256
apic policies:get	1256
apic policies:implementation	1257
apic policies:list	1257
apic policies:list-all	1258
apic policies:update	1258
apic portal-services	1259
apic portal-services:clear	1259
apic portal-services:create	1260
apic portal-services:delete	1260
apic portal-services:get	1260
apic portal-services:list	1261
apic portal-services:update	1261
apic portal-services:update-credentials	1262
apic primary-events	1262
apic primary-events:get	1263
apic primary-events:list	1263
apic products	1264
apic products:clear	1264
apic products:clear-all	1265
apic products:clone	1265
apic products:delete	1266
apic products:document	1266
apic products:execute-migration-target	1267
apic products:get	1267
apic products:list	1268
apic products:list-all	1268
apic products:migrate-subscriptions	1269
apic products:publish	1269
apic products:replace	1270

apic products:set-migration-target	1270
apic products:supersede	1271
apic products:update	1271
apic properties	1272
apic properties:clear	1272
apic properties:create	1272
apic properties:delete	1273
apic properties:get	1273
apic properties:list	1274
apic properties:update	1274
apic registrations	1275
apic registrations:clear	1275
apic registrations:create	1276
apic registrations:delete	1276
apic registrations:get	1276
apic registrations:list	1277
apic registrations:update	1277
apic role-defaults	1278
apic role-defaults:clear	1278
apic role-defaults:create	1279
apic role-defaults:delete	1279
apic role-defaults:get	1280
apic role-defaults:list	1280
apic role-defaults:list-all	1281
apic role-defaults:update	1281
apic roles	1282
apic roles:clear	1282
apic roles:create	1283
apic roles:delete	1283
apic roles:get	1284
apic roles:list	1284
apic roles:update	1285
apic services	1285
apic services:clear	1286
apic services:clear-all	1286
apic services:create	1287
apic services:delete	1287
apic services:get	1287
apic services:list	1288
apic services:list-all	1288
apic services:update	1289
apic space-settings	1289
apic space-settings:get	1290
apic space-settings:update	1290
apic spaces	1291
apic spaces:clear	1291
apic spaces:create	1292
apic spaces:delete	1292
apic spaces:get	1293
apic spaces:list	1293
apic spaces:transfer-owner	1294
apic spaces:update	1294
apic subscriber-events	1294
apic subscriber-events:get	1295
apic subscriber-events:list	1295
apic subscriptions	1296
apic subscriptions:clear	1296
apic subscriptions:create	1297
apic subscriptions:delete	1297
apic subscriptions:get	1298
apic subscriptions:list	1298
apic subscriptions:update	1299
apic task-queues	1299
apic task-queues:get	1300
apic task-queues:list	1300
apic tasks	1301
apic tasks:get	1301
apic tasks:list	1302
apic tasks:update	1302
apic tls-client-profiles	1303
apic tls-client-profiles:clear	1303
apic tls-client-profiles:clear-all	1304
apic tls-client-profiles:create	1304

apic tls-client-profiles:delete	1305
apic tls-client-profiles:get	1305
apic tls-client-profiles:list	1305
apic tls-client-profiles:list-all	1306
apic tls-client-profiles:update	1306
apic tls-server-profiles	1307
apic tls-server-profiles:clear	1307
apic tls-server-profiles:clear-all	1308
apic tls-server-profiles:create	1308
apic tls-server-profiles:delete	1309
apic tls-server-profiles:get	1309
apic tls-server-profiles:list	1309
apic tls-server-profiles:list-all	1310
apic tls-server-profiles:update	1310
apic truststores	1311
apic truststores:clear	1311
apic truststores:create	1312
apic truststores:delete	1312
apic truststores:get	1313
apic truststores:list	1313
apic truststores:update	1313
apic user-registries	1314
apic user-registries:clear	1314
apic user-registries:create	1315
apic user-registries:delete	1315
apic user-registries:execute	1316
apic user-registries:get	1316
apic user-registries:list	1317
apic user-registries:search	1317
apic user-registries:test-connection	1317
apic user-registries:update	1318
apic user-registry-settings	1318
apic user-registry-settings:get	1319
apic user-registry-settings:update	1319
apic users	1320
apic users:clear	1320
apic users:create	1320
apic users:delete	1321
apic users:get	1321
apic users:list	1322
apic users:request-password-reset	1322
apic users:search-provider	1323
apic users:update	1323
apic validate	1324
apic version	1324
apic webhooks	1325
apic webhooks:get	1325
apic webhooks:list	1326
apic webhooks:update	1326
apic wsdl	1326
apic wsdl:introspect	1327
API Connect REST APIs	1327
Example: Using the platform REST APIs to publish a product containing a SOAP API	1327
Example Product file: globalweatherproduct_1.0.0.yaml	1329
Example API definition: globalweather_1.0.0.yaml	1330
Example WSDL file: globalweather.wsdl	1334

**Note:** A later version of IBM API Connect is available.  
For details, see the [IBM API Connect Version 10 product documentation](#).

---

## Popular Topics



### Install and get started

- [Overview of IBM API Connect](#)
- [Installation - toolkit, portal, API Connect](#)
- [Key concepts](#)
- [Configure and manage your server environment](#)

[See More](#)



### Develop APIs

- [Start creating APIs](#)
- [Define paths for a REST API](#)
- [Convert an API definition for deployment to the new DataPower API Gateway](#)
- [Example](#)

[See More](#)



### Secure and manage

- [Work with Catalogs and Spaces](#)
- [Manage authentication](#)
- [Configure API security and authentication](#)

[See More](#)



### Discover and use APIs

- [Register your applications](#)
- [Browse the Catalog and sign up to APIs](#)
- [Customize your Developer Portal](#)

[See More](#)

### Additional Reading

- [Recommendations for an API Economy Center of Excellence](#)
- [Creating A Digital Ecosystem – Past, Present, and Future](#)
- [Why Become a Digital Business?](#)
- [Providing APIs or Managing APIs – There is a Big Difference](#)
- [Principles for API Security - White Paper](#)
- [API Connect 2018.4.1.x WhitePaper](#)
- [What is API Management?](#)
- [What is an API? and What is the API Economy?](#)
- [Istio Service Mesh and APICConnect/DataPower Gateway integration](#)

## Getting help

Be sure to check out these extra help resources.

### Blogs & Recipes

Find in-depth perspectives and examples from experts in API management.

[View the API Connect blog on developerWorks](#)

[Search for community-created recipes.](#)

### Forum

Ask a question in the IBM API Connect Forum, or search the Forum history to see if it's been asked before.

[Join the discussion](#)

### Support

For additional support, contact IBM Support for API Connect and API Manager.

[Contact IBM Support for API Connect and API Manager](#)

For more information, including expert advice and pricing plans, visit the [IBM API Connect product page](#)

Copyright IBM Corporation 2012, 2020. All Rights Reserved.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API Connect overview

IBM® API Connect is an integrated API management offering, with capabilities and tooling for all phases of the API lifecycle. Key steps of the API lifecycle include create, secure, manage, socialize, and analyze. IBM API Connect Version 2018 delivers enhanced capabilities for the market-leading IBM API management solution. In addition to the ability to deploy in complex, multi-cloud topologies, this version provides enhanced experiences for developers and cloud administrators in your organization.

Use of inclusive terminology: While IBM values the use of inclusive language, terms that are outside of IBM's direct influence are sometimes required for the sake of maintaining user understanding. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the product interface to reflect those changes.

IBM API Connect has two main focuses: the first is providing best in class API Management tooling, and the second is having a cloud native solution. This allows users to create, manage, and secure applications that are deployed across a variety of on-premises and cloud environments.

The following table explains the key phases in the API lifecycle in more detail.

Table 1. Key phases of the API lifecycle

Lifecycle Phase	Description
Create	Develop and write API definitions from an API development environment, eventually bundling these APIs into consumable products, and deploying them to production environments. For tutorials, walk-throughs, and in-depth guides for building, testing, and deploying APIs and Products in API Connect, see <a href="#">Tutorials</a> , and <a href="#">Developing your APIs and applications</a> .
Secure	Leverage the best-in-class API Gateway, gateway policies, and more, to manage access to your APIs and back-end systems. To learn more about adding security to your API, see <a href="#">Configuring API security</a> .  To learn more about how to add API Gateway policies to your API, see <a href="#">API policies and logic constructs</a> .
Manage	Governance structures are built in to the entire API lifecycle, from managing the view/edit permissions of APIs and Products being deployed, to managing what application developers can view and subscribe to when APIs are deployed. To understand and leverage API Connect management and governance controls along the API lifecycle, see <a href="#">Managing your APIs</a> .
Socialize	API Connect comes with an advanced Developer Portal that streamlines the onboarding process of application developers, and can be completely customized to an organization's marketing standards. To understand more about using the Developer Portal, see <a href="#">Developer Portal: Socialize your APIs</a> .
Analyze	Developers and Product Managers alike are given the tooling in API Connect to understand their API traffic patterns, latency, consumption, and more to make data driven insights into their API initiatives. To learn more about how to leverage API Connect analytics tooling, see <a href="#">API Analytics</a> .

API Connect has four major components: API Manager, Analytics, Developer Portal, and Gateway. These four components can be deployed in a variety of hybrid and multi-cloud topologies. The API Connect infrastructure can either be deployed and managed by an IBM team in an IBM Cloud environment, or it can be deployed and managed by the customers in their own dedicated environment or third-party cloud. There is also the option for having hybrid scenarios, for example, with the API Connect Reserved Instance Offering, users are able to have their API Manager and Developer Portal running in the IBM Cloud, but then place remote gateways next to their back-end services.

Configuration of customer deployed API Connect clouds is done through the Cloud Manager. For in-depth guides and instructions see [Configuring and managing your server environment](#).

For information about the API Connect components that provide these capabilities, and details about the strategy for packaging and publishing APIs for use by API consumers, see:

- [Packaging strategy and terminology in API Connect](#)
- [API Connect components](#)

Note: For a comprehensive technical guide to best practices, considerations, and deployment options for API Connect, see the [API Connect 2018.4.1.x Whitepaper](#).

- **What's new in the latest release (Version 2018.4.1.24)**  
IBM API Connect 2018.4.1.24 contains fixes and the following enhancements. Note that fix packs 2018.4.1.21, 2018.4.1.22, and 2018.4.1.23 were not released. These fix packs were skipped to resume synchronized release numbering with IBM DataPower®.
- **Version 2018.4.1 known limitations**  
This page describes known limitations for IBM API Connect Version 2018.4.1.
- **Available deployment options of API Connect**  
IBM API Connect is offered as several deployment options, depending on your needs.
- **API Connect concepts**  
To help you get started, read about the API Connect concepts and obtain a high level understanding of the API management solution.
- **API Connect gateway types**  
IBM API Connect provides two gateway types, DataPower API Gateway and DataPower Gateway (v5 compatible).
- **API Connect user roles**  
The IBM API Connect solution provides an infrastructure, tools, and facilities that allows users to create, manage, and stage APIs. The ability to perform tasks in the API Connect user interfaces is controlled through user roles, and the permissions that are assigned to those roles.
- **API Connect support**  
If you experience a problem with IBM API Connect that you cannot resolve, you can check the IBM Support website for the most recent technical bulletins and fixes. Otherwise, you can contact IBM Support.
- **API Connect glossary**  
The IBM API Connect and Cloud Manager glossary of terms and definitions.
- **Accessibility features for IBM API Connect**  
Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.
- **Legal information**  
Notices, and terms and conditions for information centers.
- **IBM API Connect Considerations for GDPR Readiness**  
Information about features of IBM API Connect that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness.
- **Essential reading**  
These articles by IBM API Connect product specialists provide a wealth of supporting information on APIs and the API economy.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## What's new in the latest release (Version 2018.4.1.24)

IBM® API Connect 2018.4.1.24 contains fixes and the following enhancements. Note that fix packs 2018.4.1.21, 2018.4.1.22, and 2018.4.1.23 were not released. These fix packs were skipped to resume synchronized release numbering with IBM DataPower®.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

VMware only: You must upgrade to 2018.4.1.20 before upgrading to 2018.4.1.24

On VMware, the only upgrade that is supported for version 2018.4.1.24 is from 2018.4.1.20; you cannot upgrade directly from older releases.

Ability to disable the Portal web endpoint check

When you create or register a Developer Portal service, the Portal subsystem checks that the Portal web endpoint is accessible. However sometimes, for example due to the complexity of public and private networks, the endpoint cannot be reached. In this instance, you can disable the Portal web endpoint check so that the Developer Portal service can be created successfully.

To disable the endpoint check, complete one of the following updates depending on your platform:

On Kubernetes

Add the following section to the Portal custom resource (CR) template:

```
spec:
  template:
    - containers:
      - env:
        - name: PORTAL_SKIP_WEB_ENDPOINT_VALIDATION
          value: "true"
        name: admin
        name: www
```

On VMware

In your `apicup` project, create a file called `ptl-extra-values.yaml` (or edit the file if one already exists), and add the following section:

```
spec:
  template:
    - containers:
      - env:
        - name: PORTAL_SKIP_WEB_ENDPOINT_VALIDATION
          value: "true"
        name: admin
        name: www
```

For more information, see [Registering a Portal service](#).

- [What's new in Version 2018.4.1.20](#)  
IBM API Connect 2018.4.1.20 contains fixes and the following enhancements.
- [What's new in Version 2018.4.1.19](#)  
IBM API Connect 2018.4.1.19 includes the following enhancements. Note: 2018.4.1.18 was not released. This fix pack was skipped to resume synchronized release numbering with IBM DataPower.
- [What's new in Version 2018.4.1.17](#)  
IBM API Connect Version 2018.4.1.17 includes the following enhancements.
- [What's new in Version 2018.4.1.16](#)  
IBM API Connect Version 2018.4.1.16 includes the following enhancements.
- [What's new in Version 2018.4.1.15](#)  
IBM API Connect Version 2018.4.1.15 includes the following enhancements.
- [What's new in Version 2018.4.1.13](#)  
IBM API Connect Version 2018.4.1.13 includes the following enhancements.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.20

IBM® API Connect 2018.4.1.20 contains fixes and the following enhancements.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

Upgrading to 2018.4.1.20 from 2018.4.1.19 on VMware requires intermediate control plane files

Upgrading to 2018.4.1.20 from 2018.4.1.19 on VMware requires intermediate control plane files as explained in the topic, [Upgrading API Connect subsystems in a VMware environment](#).

Upgrading to 2018.4.1.20 on OpenShift requires upgrading API Connect first

If you are upgrading from 2018.4.1.19 or earlier to 2018.4.1.20 on OpenShift, you must upgrade API Connect before upgrading OpenShift to version 4.10.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.19

IBM® API Connect 2018.4.1.19 includes the following enhancements. Note: 2018.4.1.18 was not released. This fix pack was skipped to resume synchronized release numbering with IBM DataPower®.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

Drupal 9 upgrade compatibility check now added to the Developer Portal upgrade process

A compatibility check has been added to the Developer Portal upgrade process to verify that any additional contributed modules, custom modules, or sub-themes, declare that they are compatible with Drupal 9. Any Developer Portal sites that contain modules or sub-themes that don't have this version declaration will fail to upgrade. For information about how to make the Drupal 9 version declaration in custom modules or sub-themes, see [Installing Drupal 8 based custom modules or sub-themes into the Drupal 9 based Developer Portal](#). For more information about Drupal 9, see [About Drupal 9](#) on the drupal.org website.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.17

IBM® API Connect Version 2018.4.1.17 includes the following enhancements.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

Prevent the `Access-Control-Allow-Credentials` header from being returned in a CORS response

By default, when CORS is enabled for an API, every response to a CORS request contains the following header:

```
Access-Control-Allow-Credentials: true
```

However, the inclusion of this header does not provide optimal security. A new `x-ibm-gateway-cors-allow-credentials-when-no-cors-policy` API property is provided to allow you to prevent the return of this header. For more information, see [API properties](#).

Upgrade from Drupal 8 to Drupal 9 content management system

The Developer Portal is now based on the Drupal 9 content management system. This is an in-place upgrade, so no specific steps need to be taken. However, if you are using additional contributed modules, or custom modules or sub-themes, you should check that they are compatible with Drupal 9, and not using any deprecated APIs for example. There are tools available for checking your custom code, such as [drupal\\_check](#) on GitHub, which checks Drupal code for

deprecations. For information about how to make the core version of custom modules or sub-themes compatible with Drupal 9, see [Installing Drupal 8 based custom modules or sub-themes into the Drupal 9 based Developer Portal](#). For more information about Drupal 9, see [About Drupal 9](#) on the drupal.org website.

Timestamp format changed for Developer Portal restore operations

The timestamp format that is used in Developer Portal restore operations has changed from `YYYY-MM-DD HH:MM:SS` to `YYYYMMDD.HHMMSS`. For more information, see:

- [Backing up and restoring the Developer Portal in a Kubernetes environment](#)
- [Backing up and restoring the Developer Portal in a VMware environment](#)
- [Backing up and restoring the Developer Portal on IBM Cloud Private](#)

New IP security command available in the Developer Portal

IP security is enabled by default, but you can now use an internal Developer Portal command to toggle the IP security between enabled and disabled at the Portal service level. For more information, see [How to manage IP security in the Developer Portal](#).

Directly navigate to specific APIs and operations in the Open API Explorer Documentation

You can now use the URL to navigate directly to specific APIs and operations in the OpenAPI Explorer Documentation for the API Connect REST APIs. You can also select the APIs for a specific version by using the drop-down menu in the header bar. See [Open API Explorer Documentation for Version 2018](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.16

IBM® API Connect Version 2018.4.1.16 includes the following enhancements.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.15

IBM® API Connect Version 2018.4.1.15 includes the following enhancements.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

Planning to upgrade to IBM API Connect V10?

Be sure to review the [Upgrade requirements and limitations](#) in the V10 documentation before you begin.

Helm v3 is required

For v2018.4.1.15, Kubernetes deployments must use Helm v3. For new installations, see [Requirements for deploying API Connect into a Kubernetes environment](#).

When upgrading from a prior version of API Connect, you must upgrade from Helm v2 to Helm v3. See [Upgrading API Connect subsystems](#).

Adding additional documentation to products and APIs

You can embed additional documentation pages within the api swagger documentation. These pages can be base64 encoded strings or markdown strings of the html content of the page. These pages show up in the navigation, on the left side, underneath the overview for that product or API, and above the paths. Any styling that is needed to theme the documentation can be included in the Developer Portal custom theme. For more information, see [Using x-embedded-doc to add additional documentation to products and APIs](#).

Ability to send API Manager metadata from the Developer Portal

For content created or updated after Version 2018.4.1.13, the Developer Portal propagates more metadata to API Manager for **consumerorgs**, **users**, and **applications**.

This metadata might be values that the user provided by adding extra fields to content types, or added programmatically by using custom modules.

The benefit of this is that when you add custom fields to **consumerorgs**, **users**, and **applications**, this data is now also stored as metadata in API Manager. Storing this data allows the Developer Portal to correct or repopulate the custom fields from a webhook or a snapshot.

Detect illegal XML characters in API request headers

A new `x-ibm-gateway-inspect-request-headers` API property enables the inspection of the HTTP headers in the API request to check for characters in the header values that are illegal XML characters. By default, there is no inspection, and such characters cause the API request to fail with an HTTP 500 Internal Server Error, but with this property you can choose to replace these characters with `?`, or to have the API request fail with an HTTP 400 Bad Request if any such characters are found. For more information, see [API properties](#).

Badgerfish support for handling of empty XML elements by the map policy

The `x-ibm-gateway-map-xml-empty-element` API property provides new options that enable empty XML input element values to be placed into JSON badgerfish value properties. For more information, see [API properties](#).

New Invoke policy properties for HTTP support

With the DataPower® API Gateway, the Invoke policy now provides the following new properties:

- Restrict to HTTP 1.0: HTTP transactions are restricted to version 1.0.
- Allow chunked uploads: Chunked-encoded documents are sent to the server.

For more information, see [Invoke](#).

Log policy performance enhancements

Rather than sending event data to the analytics server individually, the Log policy now buffers event data and sends it to the analytics server in batches according to the time interval configured for the Analytics Endpoint on the DataPower API Gateway. For more information on the Log policy, see [Log](#).



Change to the way in which the sending of client ID and scope to a third party OAuth provider is controlled

A new **suppress-parameter** header enables you to suppress the sending of client ID and scope to a third party OAuth provider; by default these parameters are now sent. For more information, see [OAuth introspection for third-party OAuth providers](#)

When activating an account through a local user registry, the specified email address must match the invitation email address

When a user activates a Cloud Manager or API Manager account through an API Connect local user registry, the email address that the user enters on the sign up page must match the email address to which the invitation email was sent, otherwise the account activation fails. Previously, a different email address could be supplied.

For other user registry types, the user supplies their existing credentials when activating their account and, upon authentication, the API Connect user record is updated from the backend identity provider.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## What's new in Version 2018.4.1.13

IBM® API Connect Version 2018.4.1.13 includes the following enhancements.

Note: You can access the latest files from [Fix Packs Available for IBM API Connect v2018.x](#).

A tutorial on how to create a drop-down menu link

You can create and use drop-down menus, or nested menus, to enhance the use of your Developer Portal.

For more information, see [Creating a drop-down menu link](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Version 2018.4.1 known limitations

This page describes known limitations for IBM® API Connect Version 2018.4.1.

Note: The limitations that are documented on this page will be removed as they are addressed in the IBM API Connect product. For the most up-to-date list of product limitations, visit the English version of this page.

You cannot upgrade to API Connect 10.0.6.0 or later from any version of 2018.

A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.


VMware only: You must upgrade to 2018.4.1.20 before upgrading to 2018.4.1.24

On VMware, the only upgrade that is supported for version 2018.4.1.24 is from 2018.4.1.20; you cannot upgrade directly from older releases.

Override plan rate limits are not displayed in the Endpoint tab

Any override plan rate limits that have been added to your API for individual operations are not displayed in the API Endpoint tab in the UI. Only the plan rate limit is displayed.

Updates to members in the UI might not automatically refresh

Updates to members in the Cloud Manager and API Manager UIs might not automatically refresh. However, you can click the Refresh icon  to manually refresh your browser.

APIC repair job fails when streaming-connection-idle-timeout is set to 0

The recommended timeout value is 1 hour. The value is set in the kubelet config file. For information on configuring this setting, see the Kubernetes kubelet documentation at <https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/>.

Viewing multiple dashboards in a non-incognito browser is not supported

Either an incognito (private) browser tab, a separate browser instance, or a full page refresh is required for viewing dashboards in multiple Analytics services simultaneously.

Default vanity API endpoints are not displayed for a Catalog with Spaces enabled

If you enable Spaces in a Catalog then, on the Settings > API Endpoints page for the Catalog, the default vanity API endpoints are not displayed.

Limitations in the analytics record of **request\_body**

Logging the API **request\_body** in Analytics is subject to the following limitations:

- A non-xml request body is not supported in the analytics record
- A non-json request body is not supported in the analytics record
- A multipart request body is not supported in the analytics record

An API returns a **Content-Type** header of **unknown**

If an API contains a GatewayScript policy that executes an **apim.setvariable** function call that manipulates **message.body**, the **Content-Type** header is set to **unknown**. This might cause issues with invoke policy target servers, or the API client. To resolve this problem, you must immediately follow the **apim.setvariable** function call with an **apim.output** function call that specifies the content type of the payload that was written to **message.body** by **apim.setvariable**.

User interfaces behave incorrectly after upgrading to a new fix pack

After upgrading to new fix pack release, the user interfaces might exhibit incorrect behavior such as missing icons, errors when attempting an action, and looping situations. To resolve this problem, first clear your browser cache before using a user interface, then if the problem persists disable your browser cache.

An OpenAPI definition that contains regular expression syntax fails validation

IBM API Connect supports the [GO regular expression syntax](#). When you import an OpenAPI definition into the API Designer or API Manager user interfaces, or validate one with the `apic validate`, the validation will fail if the OpenAPI source contains unsupported regular expression syntax, with errors that include `Does not match format 'regex'`; for example:

```
- Must validate at least one schema (anyOf) (context:
(root).paths./example/types.post.parameters.0.schema.properties.items, line: 0, col: 0)
- Must validate one and only one schema (oneOf) (context: (root).paths./example/types.post.parameters.0, line: 46164, col:
21)
- paths./example/types.post.parameters.0.schema.properties.items.properties.pattern Does not match format 'regex'
(context: (root).paths./example/types.post.parameters.0.schema.properties.items.properties.pattern, line: 0, col: 0)
```

When scaling down a Kubernetes cluster of DataPower gateways, the cluster might enter an unrecoverable state if the peering primary instance is lost

If you are scaling down a Kubernetes cluster of DataPower gateways, you must first complete the following steps to ensure that the peering primary instance is not lost:

1. List the pods in the cluster; for example:

```
$ kubectl get pods -n default --selector app=dynamic-gateway-service
NAME                                READY   STATUS    RESTARTS   AGE
rbeb67b8620-dynamic-gateway-service-0  1/1     Running   0           3h31m
rbeb67b8620-dynamic-gateway-service-1  1/1     Running   0           3h32m
rbeb67b8620-dynamic-gateway-service-2  1/1     Running   0           3h34m
```

2. Enter the DataPower admin CLI for gateway 0, by entering the following command and logging in at the prompt; for example:

```
$ kubectl attach -n default -it rbeb67b8620-dynamic-gateway-service-0
```

3. Switch to the API Connect domain and enter config mode:

```
idg# switch apiconnect
idg[apiconnect]# config
Global mode
```

4. Show the gateway peering status; for example:

```
idg[apiconnect] (config)# show gateway-peering-status
```

Address	Configuration name	Pending updates	Replication offset	Link status	Primary
1.2.3.1	gwd	0	18331896	ok	no
1.2.3.1	rate-limit	0	3091219	ok	no
1.2.3.1	subs	0	3150127	ok	no
1.2.3.1	tms	0	3063575	ok	no
1.2.3.2	gwd	0	18330948	ok	no
1.2.3.2	rate-limit	0	3091067	ok	no
1.2.3.2	subs	0	3149975	ok	no
1.2.3.2	tms	0	3063257	ok	no
1.2.3.3	gwd	0	18330948	ok	yes
1.2.3.3	rate-limit	0	3091067	ok	yes
1.2.3.3	subs	0	3149975	ok	yes
1.2.3.3	tms	0	3063257	ok	yes

5. For each of the configuration names, set gateway 0 to be the peering primary instance:

```
idg[apiconnect] (config)# gateway-peering-switch-primary gwd
The instance is now the primary in the peer group.
idg[apiconnect] (config)# gateway-peering-switch-primary rate-limit
The instance is now the primary in the peer group.
idg[apiconnect] (config)# gateway-peering-switch-primary subs
The instance is now the primary in the peer group.
idg[apiconnect] (config)# gateway-peering-switch-primary tms
The instance is now the primary in the peer group.
```

6. Show the gateway peering status to confirm the changes; for example:

```
idg[apiconnect] (config)# show gateway-peering-status
```

Address	Configuration name	Pending updates	Replication offset	Link status	Primary
1.2.3.1	gwd	0	19067953	ok	yes
1.2.3.1	rate-limit	0	3218172	ok	yes
1.2.3.1	subs	0	3277032	ok	yes
1.2.3.1	tms	0	3189160	ok	yes
1.2.3.2	gwd	0	19067953	ok	no
1.2.3.2	rate-limit	0	3218172	ok	no
1.2.3.2	subs	0	3277200	ok	no
1.2.3.2	tms	0	3189160	ok	no
1.2.3.3	gwd	0	19069241	ok	no
1.2.3.3	rate-limit	0	3218172	ok	no
1.2.3.3	subs	0	3277508	ok	no
1.2.3.3	tms	0	3189312	ok	no

Truststore and keystore settings in a shared TLS client profile are not reflected in API Manager

If you create a shared TLS client profile in the Cloud Manager user interface and assign a truststore and a keystore, these truststore and keystore settings are not reflected in that TLS client profile in the API Manager user interface and cannot be set there. Furthermore, the TLS client profile list views display the value shared for the truststore and keystore rather than the correct values. To workaround this problem, create the truststore, keystore, and TLS client profile separately for each provider organization in the API Manager user interface, then enable the TLS client profile in your Catalogs as required.

An invited Space owner cannot work with consumer organization groups

If you enable Spaces in a Catalog and invite a user with Space Owner permission, the invited user cannot create, edit, or manager consumer organization groups, and cannot add consumer organizations to an existing consumer organization group.

Install Assist utility not supported on Windows when upgrading to v2018.4.1.10

A known limitation prevents successful use of the Windows platform version of the Install Assist utility, when upgrading to v2018.4.10. The workaround is to use a Linux or macOS version of the binary to complete the upgrade.

The `apic cloud-settings:topology` command doesn't return a topology object

If you use the `apic`

`cloud-settings:topology` command without a `--format` parameter, it returns only the string `CloudTopology` rather than the cloud topology. To retrieve the topology object, include the `--format` parameter, with a value of either `yaml` or `json` as required.

The `apic cloud-settings:mail-server-configured` command response is incorrect

If you use the `apic`

`cloud-settings:mail-server-configured` without a `--format` parameter, it returns only the string `MailServerConfigured` rather than a boolean value to indicate whether an email server has been configured in the cloud settings. To retrieve the correct response, include the `--format` parameter, with a value of either `yaml` or `json` as required.

API Designer user interface might not open correctly if there are badly formed OpenAPI YAML files in the startup directory

If, when launching the API Designer user interface, the startup directory that you select contains one or more badly formed OpenAPI YAML files, the interface might fail to load correctly, with an error such as the following:

```
Cannot read property 'type' of null
```

To resolve this problem, remove any invalid OpenAPI YAML files from the startup directory, then restart the API Designer user interface.

Pagination setting is global across the API Connect user interfaces

If you set the Items per page value on any page in either the Cloud Manager or API Manager user interface, that setting is then applied to all pages in both user interfaces in the same browser session. If you want to set the value separately for a specific page, open it in a private browser window. Such a setting in a private browser window is specific to that window and is lost when the window is closed.

In Catalina, testing the WSDL-SOAP flow in designer assembly editor has no dependencies

In Catalina, for designer, if you want to use test functions inside the assembly editor for the WSDL-SOAP flow, the Invoke button is disabled. Also, you cannot generate the parameter body.

To resolve this situation, you can directly publish the WSDL-SOAP APIs to the API Manager UI. Then, use the test function inside the API Manager UI - Assembly Editor. However, when you generate the body you must remove the comments `<-- . . . -->`, as in some cases the comments are not processed correctly.

The User Security policy in API Designer does not make user registries available for selection

If, in the API Designer user interface, you add a User Security policy to an API assembly, the User registry list in the policy configuration window does not contain any user registries to select from. To work around this problem, add the user registry to the policy directly on the Source tab; define the following property in the `user-security` policy configuration:

```
user-registry: user_registry_name
```

where `user_registry_name` is the value of the `name` property of the required user registry.

Ingress timeout settings might need adjustment to enable improved synchronization in Version 4.1.8

API Connect Version 4.1.8 contains a set of core updates designed to improve synchronization, and to strengthen stability and data integrity. Due to the changes, we recommend increasing the ingress time-out settings if you see 504 or 409 errors when trying to create or delete a provider organization or catalog. The increased stability also means that APIs and Products take longer to process, compared to versions prior to 2018.4.1.8.

To change ingress timeout settings, see the section "Kubernetes/ingress-nginx ingress controller config.map settings" in the topic [Kubernetes ingress controller prerequisites](#).

Upgrade of management server to Version 2018.4.1.8 can take longer than during previous upgrades

The upgrade of the management server from prior versions to Version 2018.4.1.8 may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data.

Manual backup required when upgrading to Version 2018.4.1.8 iFix 2 or later

When upgrading to Version 2018.4.1.8 iFix 2 or later, you must complete a manual backup just prior to starting the upgrade. The manual backup is required because the upgrade can take an extended period of time. See [Requirements for upgrading on VMware](#) and [Requirements for upgrading](#) on Kubernetes.

Manual backup required when increasing memory allocation for the management database on Version 2018.4.1.8 iFix 2 or later

When changing memory allocation for the management database on Version 2018.4.1.8 iFix 2 or later, you must complete a manual backup just prior to changing the allocation. See [Increasing the memory allocation for the management database](#) on VMware and [Increasing the memory allocation for the management database](#) on Kubernetes.

If you are using API Connect v2018.4.1.8 or later, there can be a temporary delay in data sync between Management server and other services

Because of changes to improve stability, certain conditions can cause a temporary delay in the syncing of data between the Management server and other services such as the Gateway or Developer Portal. For example, there can be a temporary delay before a newly published Product runs on the Gateway, or is displayed on the Developer Portal. No action is needed to resolve the delay. Wait a few minutes and the Gateway and Developer Portal should be back in sync.

Deletion of organizations, catalogs, and applications can fail

The deletion of organizations, catalogs, or applications can fail, if they or resources inside them (like draft APIs inside provider organizations) were created with Version 2018.4.1.6 or earlier. This problem does not exist if organizations, catalogs or applications, and resources inside them were created with Version 2018.4.1.7 or later. For Version 2018.4.1.6 or earlier, internal locks can remain in place and prevent successful deletion. If you encounter this problem, please contact IBM Support and open a support ticket. IBM Support can help you to clear the locks so that the deletions complete successfully.

Management of refresh tokens not available in the UI

You must use the command line interface (CLI) to enable and manage refresh tokens. The use of refresh tokens is supported in API Manager, API Designer, the Developer Portal, and the CLI Toolkit, provided that the user logs in with a non-OIDC Provider user registry. If the registry type used for user login is OIDC, then the refresh setting is ignored. When the token expires, the user is always redirected to the login page. For more information, see [Specifying cloud settings for refresh tokens](#).

The Applications page doesn't indicate how many applications there are in the list

On the Applications page in a Catalog, there is no indication of how many applications there are in total. Therefore, if the total number of applications exceeds the Items per page setting, the list is incomplete with no indication that there are further applications that aren't displayed. To ensure that you view all applications, set the Items per page setting to the maximum value, then use the next page arrow to scroll through the complete list.

The API republish icon isn't enabled after an API update

If you make a change to an API definition in the Assemble view and save your changes, the API republish icon might not be enabled automatically, preventing you from republishing your updates. In such situations, enable the republish icon by refreshing the page, or by navigating to another view and then returning to the Assemble view.

Upgrade of DataPower API Gateway to Version 2018.4.1.7 or later requires installation of the DataPower monitor pod

API Connect Version 2018.4.1.7 introduces a new monitor pod for DataPower API Gateway. The monitor pod is required for all deployments of Version 2018.4.1.7 on Kubernetes. Installation of the monitor pod is required when upgrading DataPower Gateway from prior versions to version 2018.4.1.7 or later. For upgrade

instructions, see [Upgrading API Connect in a Kubernetes environment](#).

Enablement of DataPower Gateway high performance peering on a currently deployed system requires interruption to API transaction processing.

API Connect Version 2018.4.1.7 introduces high performance peering to DataPower API Gateway. If you do not enable high performance peering during initial installation, you can reconfigure to enable it later. Note that the reconfiguration process requires an interruption to API transaction processing, so you should plan accordingly. For Kubernetes, see [Enabling high performance peering for DataPower API Gateway on Kubernetes](#). For VMware environments, see [Enabling multiple peering objects on gateway services external to Kubernetes](#).

Setting resource limits in Kubernetes can cause containers to behave oddly at run time

In Kubernetes, you can set resource limits for each container. However, limiting CPU means that the container processes might slow down if they try to use too much. Limiting memory means that the Kubernetes code kills processes that try to allocate memory that would take the container over its limit. These limits can result in odd behavior such as the containers continually trying to start a daemon process, such as nginx or node, and exiting with success immediately.

Field validation for the Client security policy is incorrect

When configuring a Client Security policy in an API assembly in the API Designer or API Manager user interfaces, there is the following incorrect validation behavior:

- The ID Name field is required, but the API definition can be saved without entering a value in the field.
- The Secret Name field is required only when the Secret Required option is selected, but the user interface indicates that the Secret Name field is required regardless. Furthermore, when the field is required, the API definition can be save without entering a value.
- If the Authenticate Client Method is set to Third party, the User Registry Name field is required, but the API definition can be saved without entering a value in this field.

The title of a WSDL API can't be changed

If, when editing a WSDL API definition, you attempt to change the Title field, an error is returned. You can, however, change the title by modifying the `title` property in the OpenAPI source on the Source tab.

A WSDL API cannot be saved as a new version

If, when editing a WSDL API definition, you attempt to change the Version field, or use the Create new version option, an error is returned.

User interface URL returns 404 error on OpenShift

If IBM API Connect is deployed to an OpenShift environment, attempts to open the API Manager user interface at the URL `https://api_manager_ui_endpoint` returns a 404 error because the redirect to the full URL fails. To avoid this problem, use the full URLs for the user interfaces, as follows:

- API Manager user interface: `https://api_manager_ui_endpoint/manager`
- Cloud Manager user interface: `https://cloud_admin_ui_endpoint/admin`

User interface does not provide the option to assign roles when inviting Space members

When you invite a user to join as a member of a Space by using the API Manager user interface, it is not possible to assign roles to the user. This means that when the user receives the invitation, activates their account, and logs in, they have no permissions. To resolve this problem, assign the required roles to the user after they have activated their account, by selecting the roles alongside the user on the Members page in the Space.

You can, however, assign roles when adding a member to a Space, and when adding or inviting members to a Catalog.

Requests that take longer than the value of the nginx timeout cause a 409 error

Requests that take longer than the value of the `nginx` timeout can cause a 409 error, such as:

```
Another request is operating on portal-subsystem-03729230-4df7-4419-94c9-e7691b616382 with the same name, please try again.
```

The error occurs because the ingress controller is sending a repeat request when the previous one timed out. Your request will still continue processing in the background.

In some scenarios, applications crash due to lack of sufficient memory for Cassandra

Some applications that consume extensive amounts of memory can crash because the memory requirements exceed the memory allocation for Cassandra. For example, a large number of subscription tables, or upgrade requirements. Sample error from Cassandra logs:

```
r7430a60641-apiconnect-cc-0.log:io.netty.util.internal.OutOfDirectMemoryError: failed to allocate 16921 byte(s) of direct memory (used: 67100185, max: 67108864)
```

**Workaround:** Choose one of the following:

- Allocate more memory on Cassandra. See [Increasing the memory allocation for the management database](#).
- Scale up Cassandra by adding a node, which increases the memory in the cluster

Adding members to a provider organization might fail

When adding members to a provider organization, the Cloud Manager user interface lists the current owner and members as well. If existing members are selected, the add action will fail.

Upgrading with new certificates breaks analytics ingestion

If the analytics subsystem certificates are modified after installation ( by using the `apicup certs set` command , followed by the `apicup install` command to deploy the change to the cluster), the analytics ingestion ingress will no longer be recognized by the Gateway. The analytics service must be re-registered in the Cloud Manager interface so that the new certificates are recognized.

Publishing a large Product might result in an out of memory error

Publishing a Product that contains large numbers of APIs might result in an out of memory error. To resolve this problem, divide the Product into multiple Products with fewer APIs, or increase or tune the memory settings for API Connect/Cassandra.

Users are not logged out of the user interfaces after tokens expire

If an API Manager or Cloud Manager user interface session times out, the user does not get redirected to the login page and might see an error. To redirect to the login page, refresh the browser window.

User interfaces are not supported in Microsoft Edge

The API Manager and Cloud Manager user interfaces are not supported in the Microsoft Edge web browser. To work in the user interfaces, use a different browser.

New gateways cannot connect to existing gateways after the gateway peering certificates are changed

Changing the gateway peering SSL certificates after gateway installation causes new gateways to be unable to connect to existing gateways, resulting in an outage. A full restart of the gateway StatefulSet (by deleting all the StatefulSet pods) is required for the gateways to come back online.

Changes to an OAuth provider are not reflected in the consuming APIs

If you modify the endpoints, scopes, or grants for an OAuth provider, the APIs that consume that OAuth provider will be affected. You might need to update the consuming APIs accordingly.

Deletion of a non-empty provider organizations, Catalogs, or Spaces might fail

The deletion of provider organization, Catalog, or Space that contains a large number of resources might fail. Before deleting provider organizations, Catalogs, or Spaces, first delete the contained resources.

No option to bulk delete APIs or Products

In the API Designer and API Manager user interfaces, there is currently no option to delete multiple APIs or Products in a single operation; in the user interfaces, APIs and Products must be deleted individually. However, you can bulk delete APIs and Products by using the REST API or CLI interfaces.

Cannot publish an imported API that was generated with IBM API Connect Version 5.0 LoopBack®

If you import an API YAML file that was generated with IBM API Connect Version 5.0 LoopBack, attempts to publish the API fail. Before publishing, remove the catalogs section, and the invoke policy named `tls-invoke-profile`, from the YAML source of the API definition; for example:

```
catalogs:
  apic-dev:
    properties:
      runtime-url: $(TARGET_URL)
  sb:
    properties:
      runtime-url: 'http://localhost:4001'
```

Can't publish an API that has duplicate security definition entries

The API Designer and API Manager user interfaces allow you to add duplicate security definitions to an API. However, attempts to publish the API will fail with OpenAPI validation errors. Ensure that the security definitions in an API are unique.

API import can fail if there are validation errors

Validation errors in the OpenAPI definition of an API can cause the importing of the API to fail. The errors must be corrected in the OpenAPI source before the API can be imported successfully. You can use the `apic validate` CLI command to validate the OpenAPI definition before importing.

Kubernetes might excessively evict pods

In Kubernetes, if insufficient memory, CPU, or storage resources are provided then pods are evicted randomly. To avoid this problem, allocate sufficient resources to the pods. In addition, there are known issues with log rotation in Kubernetes that are documented here: <https://github.com/kubernetes/kubernetes/issues/59902>. The lack of log rotation can cause logs to grow, limited only by the amount of storage on the node, eventually causing pods to restart. Therefore ensure that appropriate log levels are configured for all API Connect components. You may also follow the instructions described here <https://success.docker.com/article/how-to-setup-log-rotation-post-installation> to set the maximum size of the log before it is rolled (max-size) and the maximum number of log files that can be present (max-files). More information about the Docker JSON File logging driver is available here: <https://docs.docker.com/config/containers/logging/json-file/>.

A Product that was defined in API Connect Version 5 might fail to publish

If you import a Product and it's referenced APIs, and the Product and APIs were defined in API Connect Version 5, attempts to publish the Product fail because the API references are lost from the Product. To resolve this situation, add the APIs to the Product, by using either the Product Design page or Source page, before publishing again.

API Manager pods might restart due to an out of memory condition

It is possible that during heavy workloads over long periods of time the API Manager pods might reach an out-of-memory condition causing them to restart. The multiple node design of the environment will handle the load while the pod restarts.

Changing the analytics association for a Gateway service might result in an error 504

If you change the analytics association for a Gateway service in the Cloud Manager user interface, and the Gateway service is enabled in multiple Catalogs, the user interface operation might timeout with an error 504. However, the operation itself completes successfully.

The migration of many subscriptions in a single operation might fail

If you use any of the following CLI commands, or their API Manager user interface equivalents where available, and the resulting number of application subscriptions that need to be migrated is large, the operation might fail:

- `apic products:replace`
- `apic products:execute-migration-target`
- `apic products:migrate-subscriptions`

You can resolve the problem in either of the following ways:

- Use the `apic products:migrate-subscriptions` to migrate the subscriptions in small batches. If you are replacing a Product, repeat the `apic products:replace` command after migrating all the subscriptions.
- Set a migration target, by using either the `apic products:set-migration-target` command or the Set migration target operation on the Product in the API Manager user interface. to allow users to migrate their subscriptions individually.

The deletion of many subscriptions in a single operation might fail

If you use any of the following CLI commands, or their API Manager user interface equivalents where available, and they result in many consumer organizations needing to be deleted in turn, the operation might fail:

- `apic orgs:delete`
- `apic catalogs:delete`
- `apic catalogs:clear`
- `apic spaces:delete`
- `apic spaces:clear`
- `apic consumer-orgs:clear`

The operations might fail due to the large number of child resources that need to be deleted in conjunction with the resources you are trying to delete. As a workaround, delete the child resources, such as the consumer organizations, individually prior to deleting the parent resource.

Logging in to the API Connect user interfaces fails when using the Safari web browser

If you are using the Safari web browser and a Basic Authorization header exists for the same DNS domain in which API Connect is running, attempts to log in to the API Connect user interfaces, or to sign up by using an activation link, fail. To avoid this problem, use an alternative web browser.

**Endpoints cannot be changed by using APICUP after they have been configured in Cloud Manager**

The changing of endpoints with APICUP after they have been configured in Cloud Manager is not supported. Any such endpoint changes will not be propagated to a running deployment.

If the name of an API is changed, Products that contain the API fail to publish

If you change the `x-ibm-name` field of an API in the Source tab, or you change the Title field of an API in the Design tab, thereby causing the `x-ibm-name` field to be updated, any Products that contain the API fail to publish because the API name reference in the Product is not updated. To resolve this problem, open the Source tab for the Product and update the API `name` references to match the new name.

If an API is deleted from a Product, it is not deleted from Plans in the Product

If an API is deleted from a Product, it is not deleted from any Plans in the Product that contain that API. After deleting an API from a Product you must manually delete the API from any Plans to which it was previously added.

The `apicup` utility appears to hang after executing a backup or restore command

On Version 2018.4.1, the `apicup` utility might appear to hang after you run a `backup` or `restore` command. You can verify that `apicup` is still running, or succeeded:

- For a backup command, such as `apicup subsys exec backup`:
  - Verify that the backup job is created. Enter `kubect1 get jobs -n <namespace>` and verify that the job **successful** status is **1**.
  - Verify that the backup pod is created. Enter `kubect1 get pods -n <namespace>` `-a` and verify that the pod status is either still running or **completed**.
- For a restore command, such as `apicup subsys exec restore`:
  - Verify that the restore job is created. Enter `kubect1 get jobs -n <namespace>` and verify that the job **successful** status is **1**.
  - Verify that the restore pod is created. Enter `kubect1 get pods -n <namespace>` `-a` and verify that the pod status is either still running or **completed**.

Unenforced APIs are not supported in the user interfaces

The setting of the Enforced API option to false, meaning that the API is to run on a gateway other than an API Connect gateway, is not currently supported in the API Designer or API Manager user interfaces. Attempts to publish, from the user interfaces, a Product that contains a non-enforced API will fail. To publish such a Product, use the developer toolkit CLI, or the REST APIs.

The API Designer user interface cannot be used with an OIDC user registry

You cannot use an OIDC user registry for logging in to the API Designer user interface. Either configure a non-OIDC registry for API Designer use, or use the equivalent capability in the API Manager user interface.

API Designer limitation when running on IBM Cloud

API Designer cannot establish a connection to the management server in order to create a cloud connection when the management server is running on the IBM Cloud platform.

Terms of use cannot be enforced for OIDC Developer Portal users.

You can configure terms of use for users of the Developer Portal so that the user must accept the terms and conditions before they can register or log in. This feature does not work if the user is using OIDC.

For more information, see [Forcing new users to accept terms and conditions](#).

An OAuth provider fails if the resources that it references aren't enabled in the Catalog

If you enable an OAuth provider in a Catalog then any resources that it references, such as API user registries or TLS client profiles, must be enabled in the same Catalog; if not, then although the OAuth provider might publish successfully it will fail at run time. For information on enabling resources in a Catalog, see [Creating and configuring Catalogs](#).

API Connect does not support resizing storage capacity or modifying storage class of the Cassandra database

If you updated `cassandra-volume-size-gb` or `storage-class` in `apiconnect-up.yml` prior to upgrading the management server, the upgrade completes successfully but the Cassandra servers are not upgraded. You can correct this problem by following the troubleshooting steps in [Upgrading in a Kubernetes environment](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Available deployment options of API Connect

IBM® API Connect is offered as several deployment options, depending on your needs.

### V2018 Reserved Instance

---

IBM API Connect V2018 Reserved Instance offers an individual API Connect instance that runs on IBM-managed infrastructure. V2018 Reserved Instance provides value by balancing the flexibility of a shared V2018 Reserved Instance infrastructure with the isolation of a single-tenant deployment based on the topology and functionality of API Connect.

V2018 Reserved Instance offers the following key features:

- Common log-in with other IBM services that use IBMid.
- Isolation from other users of the IBM Cloud public service.
- Managed, monitored, and operated by the API Connect Operations team.
- Deployed across multiple pods within the datacenter-zone for resilience.
- Optionally deployed as a High Availability (HA) deployment with a 99.95% SLA, for an additional cost.

For more information, see the [API Connect V2018 Reserved Instance](#) documentation.

### IBM Cloud Private

---

The IBM Cloud Private format provides some of the advantages of the public cloud, but with more security and control. The IBM Cloud Private deployment includes the following key features:

- Cloud environment that is set up on a private server.
- Security precautions can be added, including a private firewall.
- Control the timing of updates.

### On premises

---

The on premises format provides an installed version of API Connect. The on premises deployment includes the following key features:



- Highest level of security, as it is installed on your company server that can be behind a firewall.
- Control the timing of updates.

## Comparison table

The following table provides a convenient comparison of the different formats:

Feature	V2018 Reserved Instance	IBM Cloud Private	On-premises
Managed by IBM operations	Yes	No	No
Additional P-Orgs	By request	Yes	Yes
Shared gateways	No	No	No
Single Tenant gateways	Yes	Yes	Yes
Remote gateways	Yes	Yes	Yes
VPN connectivity	No	Yes	Yes
Single-DC HA	Yes	Yes	Yes
Multi-DC HA	Available at an extra cost	Yes	Yes
Custom branding	Yes	Yes	Yes
CMC access	No	Yes	Yes
API Manager access	Yes	Yes	Yes
DataPower UI access	No	Yes	Yes
User-defined policies	Yes <sup>1</sup>	Yes	Yes
Gateway log offload	Yes <sup>1</sup>	Yes	Yes
Analytics offload	Yes <sup>1</sup>	Yes	Yes
Custom extensions	Yes <sup>1</sup>	Yes	Yes

<sup>1</sup> If any custom code is required for these features, IBM agrees to apply code (supplied by you) to your reserved instance of API Connect; however, IBM cannot write or maintain custom code in the SaaS environment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API Connect concepts

To help you get started, read about the API Connect concepts and obtain a high level understanding of the API management solution.

- [Packaging strategy and terminology in API Connect](#)  
API Connect uses a proprietary packaging strategy for creating and publishing collections of APIs.
- [API Connect components](#)  
The API Connect components provide a unified user experience across the API lifecycle. Changes in one stage of the API lifecycle are automatically reflected in the other components of API Connect.
- [API Connect: End-to-end solution example](#)  
This example summarizes the concepts relating to the creation and use of APIs in the API Connect on-premises solution. It depicts the workflow and highlights some of the default roles for the tasks completed during the API lifecycle.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Packaging strategy and terminology in API Connect

API Connect uses a proprietary packaging strategy for creating and publishing collections of APIs.

The packaging strategy supports API providers in meeting the requirements of the API consumers. An understanding of the concepts and terminology behind the packaging strategy is required before developing and deploying APIs using IBM® API Connect.

The following sections describe the concepts and terminology behind the packaging strategy for IBM API Connect:

- [APIs](#)
- [Plans and Products](#)
- [Catalog and Spaces](#)
- [Organizations and users](#)
- [LoopBack® applications](#)
- [Sample provider organization with two Catalogs](#)

## APIs

An Application Programming Interface (API) is an industry-standard software technology. An API is a set of routines, protocols, and tools for building software applications. An API specifies how software components interact and provides quick access to common assets and processes. APIs can be public (such as offered on

GitHub), can require client credentials, or can be kept private within an application. Thus, APIs are classified as external (public), partner (protected), or internal (private), based on how they are consumed.

An API is composed of operations, called methods, which are offered in one of the following styles in API Connect:

- A REST API is structured according to the principles of Representational State Transfer. REST APIs use HTTP or HTTPS requests to PUT, GET, POST, and DELETE data (also referred to as CRUD operations). REST identifies resources using URIs. Data can be described in a variety of formats (XML, HTML, JSON, TXT, etc.), with JSON being the popular choice. REST APIs specify MIME (Multipurpose Internet Mail Extensions) types. REST is platform- and language-independent and works across firewalls using HTTPS. REST APIs leverage HTTP standards for security, caching, and status codes. HTTP clients and servers are available for all major programming languages and operating system/hardware platforms. REST implementations are easily scaled due to use of HTTP and browsers as a uniform interface.
- A SOAP (Simple Object Access Protocol) API is a web service that is exposed as an API. SOAP interfaces are described in WSDL (Web Services Description Language) format. The WSDL is an XML document describing the structure for headers, messages, URL endpoints, and datatypes used to access a web service. SOAP is considered more secure than REST as it supports WS-Security as well as SSL. SOAP also contains WS-Reliable Messaging for reliability, rather than relying on retrying the operation (as does REST). SOAP requires a client application and is better suited for enterprise applications that require secure transactions.

APIs can be versioned and packaged into multiple Products for distribution to API consumers on the Developer Portal. For information about creating and managing APIs, see [Developing your APIs and applications](#) and [Managing your APIs](#).

## Plans and Products

Plans and Products are proprietary packaging constructs that are unique to API Connect. API providers use Products to offer one or more APIs to the application developers who will consume the APIs (API consumers). The providers use Plans to control access to APIs and to manage API usage. Products are packages that contain both the APIs and the accompanying Plans. See [Working with Products](#).

To make an API available to an application developer, it must be included in at least one Product and at least one Plan.

Plans perform the following functions:

- Control which APIs an application developer can use
- Make available a collection of operations from one or more APIs
- Apply rate limits to APIs to differentiate between offerings
- Implement different rate limits to specify how many requests a consuming application is allowed to make during a specified time interval

Plans can use differing rate limits to provide different levels of service to API consumers. For example, a "Demo Plan" might enforce a rate limit of ten calls per minute, while a "Full Plan" might permit up to 1000 calls per second.

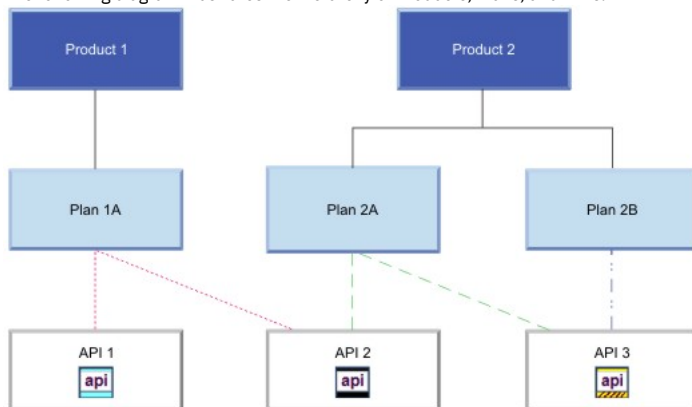
A Product in API Connect bundles a set of APIs and Plans into one offering that is intended for a particular use. You can create Plans only within Products, and these Products are then published in a Catalog. The following rules apply for the relationship between Products and Plans:

- A Plan can belong to only one Product.
- A Product can have multiple Plans that each contain a different set of APIs.
- A Plan in one Product can share APIs with Plans from any other Product.

Multiple Plans within a single Product provide different levels of performance for the same offering. For example, a Product can include a "Demo Plan" that makes a single API available, and a "Full Plan" that makes several APIs available.

Products are used to manage the lifecycle of the APIs they contain. The states in the Product lifecycle include draft, staged, published, deprecated, and retired. A Product in draft state is moved to the staged state when it is staged to a *Catalog*. A Product moves to the published state when the Product is published. The APIs in a Product become accessible to API consumers when the Product is in the published state and they then become visible on a *Developer Portal*. After a Product is published, application developers can use the Developer Portal to gain access to its APIs by registering applications to one or more Plans in the Product.

The following diagram illustrates the hierarchy of Products, Plans, and APIs.



For more information about managing the lifecycle of Products, see [Working with Products in the API Manager](#).

## Catalogs and Spaces

A Catalog contains a collection of Products. Catalogs are staging targets through which Products (together with the accompanying Plans and APIs) are published on a Developer Portal. Catalogs are used to separate Products and APIs for testing before publishing them on a Developer Portal. In a typical workflow, an API provider uses a development Catalog when developing and testing APIs, and uses a production Catalog for publishing APIs that are ready for external use. Each Catalog has an associated Developer Portal for exposing the published Products. A Catalog includes runtime capability through an associated gateway service that handles any API requests for the APIs in that Catalog.

API Connect includes a syndication feature that enables API providers to partition a Catalog into multiple staging targets (or *Spaces*) for API development purposes. Each API provider development team can use its own dedicated Space to manage its Products independently of other teams. A Space has its own set of capabilities relating



specifically to the Products and APIs that are created and published to that Space. Products and APIs in all Spaces in a given Catalog are published to the same Developer Portal. Spaces are not visible on the Developer Portal. Application developers who consume the APIs on the Developer Portal are unaware of the Space configuration used by the API Developers. On the Developer Portal, the APIs are seen as a coordinated offering within a Catalog.

For more information about Catalogs and Spaces, see [Working with Catalogs](#) and [Using syndication to partition Catalogs into Spaces](#).

## Organizations and users

In the context of API Connect, there are two types of organizations: provider and consumer. An organization can encompass a project team, department, or division.

A provider organization owns APIs, and associated Plans and Products, and can additionally own *provider* applications that are called by APIs. To complete the various functions in the API lifecycle, a provider organization assigns responsibilities for certain tasks. Some standard responsibilities within a provider organization include:

- A provider organization owner who has the full set of access permissions to API Connect functions, and can also commission APIs and track their business adoption.
- API developers who design and develop APIs and applications for the provider organizations to which they belong.
- An administrator who manages the lifecycles of APIs and publishes APIs for discovery and use.
- A "community" manager who manages the relationship between the provider organization and application developers, provides information about API usage, and provides support to application developers.

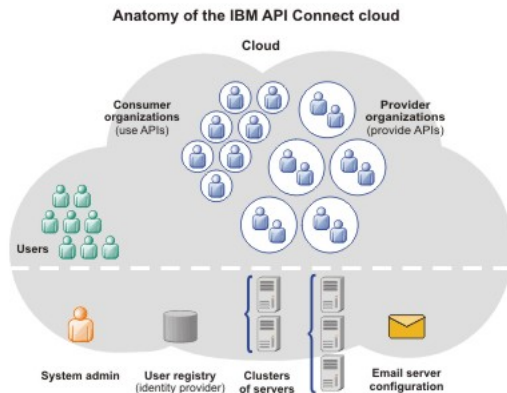
A consumer organization owns only *developer* applications, and consumes the APIs and applications produced by the provider organization. Standard responsibilities within a consumer organization include:

- A consumer organization owner who adds application developers to the consumer organization, views the Products and APIs that the provider organization has made available on the Developer Portal, and subscribes to APIs to use them in applications.
- Application developers who view the Products and APIs that the provider organization has made available on the Developer Portal, and subscribe to use these APIs in applications.

The provider and consumer organization responsibilities map to *roles* within API Connect. Some roles are independent of an organization; for example, an administrator who manages the cloud infrastructure and keeps the system running. For information about the full set of defined roles and access permissions, see [API Connect user roles](#).

Users have an existence in the API Connect ecosystem that is independent of an organization. A user can be a member of more than one provider or consumer organization.

There can be multiple provider organizations in one API Connect cloud, to provide an API development environment for each line of business of an enterprise. The API Connect cloud is a collection of servers that comprise an API Connect installation, including the configuration information and metadata that they contain. The cloud infrastructure is shared by all organizations, and managed independently of them (by a cloud or system administrator). The following diagram shows the relationship between the provider organizations, consumer organizations, and users. The clusters shown are logical groupings of servers with the same capability.



For more information about organizations, see [Administering provider organizations](#) and [Administering consumer organizations](#).

## LoopBack applications

In addition to APIs, provider organizations can also create applications (with associated APIs), which are built using Node.js and Java technology. When published, these APIs and applications are called by developer applications. The developer applications contain client code that accesses APIs to interact with a service, system, or content. The developer applications are typically mobile or web applications that use the HTTP protocol.

For information about creating LoopBack applications, see [Creating APIs and applications](#).

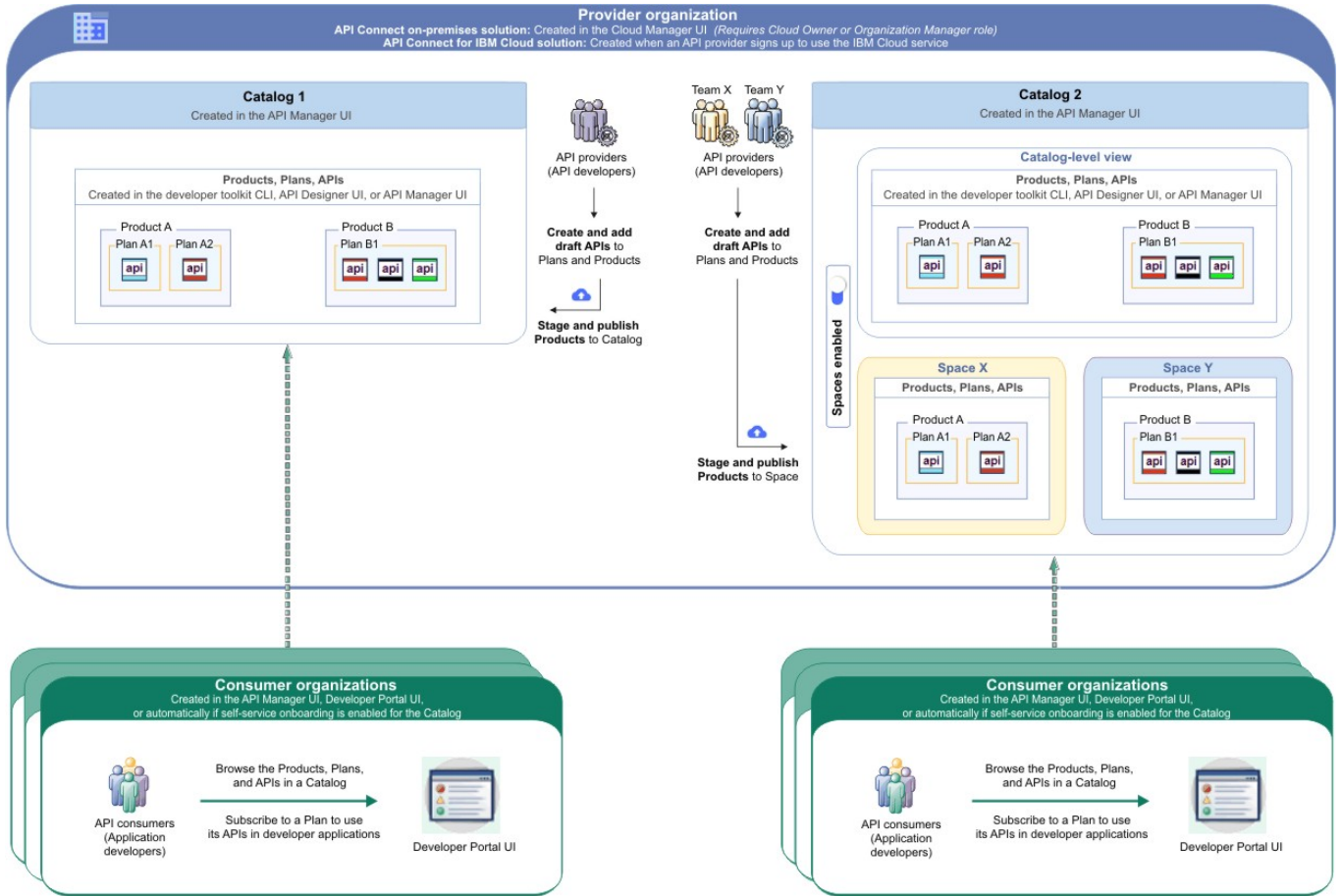
## Sample provider organization with two Catalogs

The following diagram shows an example for how APIs, Plans, Products, Catalogs, and Spaces fit within provider and consumer organizations. In this example, two Catalogs are created in the provider organization to act as staging targets for different sets of requirements.

- Catalog 1 does not require separation of the Products for use by individual provider development teams, and so does not have Spaces enabled. API developers with access to this Catalog create draft APIs, and stage and publish them as Products to the Catalog. Several consumer organizations are granted permission to explore,

discover, and use the published Products and APIs. In each of the consumer organizations, the published Products and APIs from Catalog 1 are exposed on a single Developer Portal.

- Catalog 2 is partitioned into two Spaces (X and Y) so that two provider development teams can manage their Products independently. These teams stage and publish their APIs (as Products) separately to the individual Spaces, to make them accessible to application developers in multiple consumer organizations. In each of the developer organizations, the published Products and APIs from both Spaces are exposed in the same Developer Portal, and application developers who access this portal will see the APIs from both Spaces as a coordinated offering.



- **Understanding rate limits for APIs and Plans**

In API Connect, you can configure rate limiting on APIs and Plans to manage network traffic and API usage.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Understanding rate limits for APIs and Plans

In API Connect, you can configure rate limiting on APIs and Plans to manage network traffic and API usage.

### What are rate limits and burst limits?

A rate limit is the maximum number of calls you want to allow in a particular time interval. Setting rate limits enables you to manage the network traffic for your APIs and for specific operations within your APIs. API Connect supports two types of rate limiting:

#### Rate limit

A specified number of calls to be accepted within a defined time period; for example, 100 calls per minute. In API Connect, rate limits can be defined as unlimited, or with a specified number of calls per second, minute, hour, day, or week. When the limit is reached, no more calls are accepted from that customer until the beginning of the next time period. For example, you might want to permit a total of 1000 calls per hour (rate limit). If a customer makes 1000 calls in the first 10 minutes, they cannot complete any more calls until the hour has expired.

#### Burst limit

A rate limit that is applied to a very small time period. In API Connect, burst limits are defined for multiples of seconds or minutes. When a burst limit is reached, calls are not accepted until the time period has expired. Once that pause is over, calls continue to be accepted until the rate limit is reached. Configuring a burst limit prevents usage spikes and ensures that the rate limit is evenly spread across its overall time period. For example, you might want to permit a total of 1000 calls per hour (rate limit) and a maximum spike of 50 calls per second (burst limit).

### Where can you apply rate limits?

You can configure rate limiting at several levels of an API and the Plan containing it. Use the different settings to fine-tune your rate limits for your customers and environment. The more granular settings on APIs override the settings defined in a Plan that contains that API. For example, if you set a rate limit to 100 calls per minute on a Plan, that limit only affects APIs in the Plan that don't already have a more granular rate limit defined. The following list explains where you can configure rate limits, in decreasing levels of granularity:

1. API assembly

You can define a rate limit within the API's assembly (its process flow). This is useful when you want to dynamically set the rate limit based on a GatewayScript action; for example, you might want to increase the base rate limit only for certain users, IP addresses, or server names. For more information on configuring a rate limit on an API assembly, see the following documentation:

- [Rate Limit policy](#)
- [GatewayScript policy](#)
- [Using context variables in a GatewayScript policy](#)

2. Plan: API path and operation

You can define a rate limit for a specific path and operation of an API within a Plan; for example, you might allow unlimited calls to a GET operation but enforce a limit on PUT operations for the same API. You can even define multiple rate limits on the same path and operation. For more information on configuring rate limits on API paths and operations within a plan, see the following topics:

- [Defining rate limits for an API operation](#)
- [Describing Plans in your Product](#)

3. Plan: all APIs

You can additionally define a general rate limit at the Plan level. This limit is applied as a default for each API in the plan. Any rate limit defined at a more granular level overrides the general Plan-level limit. For more information, see [Editing a draft Product](#).

Note: If the subscriber changes to a different Plan, only the Plan-level rates limit are reset. If there is a rate limit specified in the API's definition or assembly, that rate limit is still enforced under the new Plan.

---

## How can you test rate limits?

When you're ready to test rate limits on an API or Plan, make sure you deploy it in a test Catalog where automatic subscriptions are not enabled. Execute a larger number of calls than specified by your rate limit, within a shorter period of time. Then verify that calls exceeding the limit were not accepted. If you enabled the use of a hard limit, each API call that is over the limit is returned with an error message.

The built-in test application that is used by the API Manager and API Designer Test tool is not subject to rate limits if you enabled automatic subscriptions for the Catalog where you are testing. To ensure that your rate limits are applied as intended, create a new test Catalog that requires manual subscriptions and test your API and Plan there. For more information on creating and configuring Catalogs, see [Working with Catalogs](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## API Connect components

The API Connect components provide a unified user experience across the API lifecycle. Changes in one stage of the API lifecycle are automatically reflected in the other components of API Connect.

- [Cloud Manager](#)
- [The developer toolkit](#)
- [API Manager](#)
- [API Gateways](#)
- [Runtime](#)
- [Developer Portal](#)
- [API Analytics](#)
- [Typical tasks per interface component](#)
- [API Connect server requirements](#)

---

## Cloud Manager

The API Connect Cloud Manager component is used to manage the API Connect on-premises cloud. The Cloud Administrator uses this UI to:

- Define the cluster of *Management servers*, *Gateway servers*, and *containers* that are required in the cloud, and configure the topology. For information about Management servers and Gateway servers, see [API Connect server requirements](#). For information about containers, see [Runtime](#).
- Manage (modify, move, remove, restart, reboot) the servers in the cloud.
- Monitor the health of the cloud.
- Define and manage the provider organizations that develop APIs. (Assigned managers or owners of provider organizations can also complete this task.)
- Define additional cloud administrators, or set up users with roles that enable access to specific capabilities.
- Add user registries for authenticating users and securing APIs, and configure the secure transmission of data (for example, through websites).

For more information about the Cloud Manager, see [Managing your cloud](#).

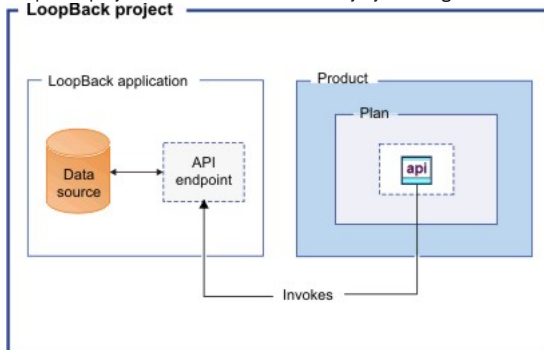
---

## The developer toolkit

The developer toolkit provides the tools for modeling, developing, and testing APIs and LoopBack® applications. The developer toolkit includes a command line interface (CLI). It also incorporates LoopBack, an open source Node.js framework.

API developers use the API management functions in the API Manager or the CLI to create draft API definitions for REST and SOAP APIs, or for OAuth provider endpoints that are used for OAuth 2.0 authentication. The API definitions can be configured to add the API to a Product, add a policy assembly flow (to manipulate requests/responses), and to define security options and other settings. APIs can then be tested locally prior to publishing, to ensure they are defined and implemented correctly.

Using LoopBack, an API developer can create a Node.js application, connect to a data source such as a back-end database or a REST API to be consumed, and then expose the application as a REST API by creating a model definition. A LoopBack model defines the application data, validation rules, data access capabilities, and business logic for an API, and provides a REST API by default. This REST API can then be used by a REST API definition that was created using the API Manager or CLI and exposed to your users. The API and its associated application, which are implemented as a LoopBack project, must both be published to enable the project to be run. LoopBack projects can also be tested locally by creating a local runtime environment. The following diagram illustrates the LoopBack project architecture:



Draft APIs (in their containing Products) that are created by using the API Manager, CLI, or LoopBack are published to Catalogs. Applications created using LoopBack are published to containers, from where they run when called. (For information about containers, see [Runtime](#).)

The developer toolkit is installed locally, for offline API and application development. For more information about the developer toolkit, see [Developing your APIs and applications](#). For more information about LoopBack, see [LoopBack: The Node.js API Framework](#).

## API Manager

The API Manager provides a user interface that facilitates promotion and tracking of APIs that are packaged within Products and Plans. API providers can move the Products through their lifecycle, and manage the availability and visibility of APIs and Plans.

Catalogs and Spaces are created in the API Manager to act as staging targets through which APIs, Plans, and Products are published to consumer organizations. API providers can stage their Products to Catalogs or Spaces, and then publish them to make the APIs in those Products visible on a *Developer Portal* for external discovery.

To control access to the available API management functions, users in the provider organization can be set up in the API Manager UI with assigned roles and permissions. API providers can also use the UI to manage the consumer organizations that sign up to access their APIs and Plans. Developer *communities* can additionally be created as a way of grouping together a collection of consumer organizations to whom a particular set of Products and Plans can be made available.

The API Manager UI also includes functions to manage the security of the API environment, and provides access to analytics information about API invocation metrics within customizable dashboard views.

For more information about the API Manager, see [Managing your APIs](#).

## API Gateways

Gateways enforce runtime policies to secure and control API traffic, provide the endpoints that expose APIs to the calling applications, and provide assembly functions that enable APIs to integrate with various endpoints. They also log and report all API interactions to the API Connect analytics engine, for real-time and historical analytics and reporting. The following Gateway is available for use in API Connect:

- The DataPower® Gateway is an enterprise API Gateway that is built for departments and cross-enterprise usage. This Gateway provides a comprehensive set of API policies for security, traffic management, mediation, acceleration, and non-HTTP protocol support. The DataPower Gateway is deployed on a virtual or physical DataPower appliance and supports multiple Catalogs per instance or cluster. The DataPower Gateway can handle enterprise level complex integration, and supports containers for flexible runtime management.

Your API Connect offering (or *edition*) can include a virtual DataPower Gateway, and support for a physical DataPower Gateway is also available, subject to certain conditions.

## Runtime

You can run applications and API implementations in API Connect in a containerized runtime.

### Containerized runtime

A containerized runtime environment provides a lightweight deployment location for APIs and applications. A container wraps an application in a complete file system that includes everything it needs to run, such as code, runtime, system tools, and system libraries. You can use Docker Swarm or Kubernetes containers to run your APIs and applications being managed by API Connect.

## Developer Portal

The Developer Portal provides a customizable self-service web-based portal to application developers to explore, discover, and subscribe to APIs.

When API providers publish APIs in the API Manager, those APIs are exposed in the Developer Portal for discovery and usage by application developers in consumer organizations. Application developers can access the Developer Portal UI to register their applications, discover APIs, use the required APIs in their applications (with access approval where necessary), and subsequently deploy those applications.

The Developer Portal provides additional features, such as forums, blogs, comments, and ratings, for socialization and collaboration. API consumers can also view analytics information about the APIs that are used by an application, or used within a consumer organization. For more information, see [Developer Portal: socialize your APIs](#).

## API Analytics

API Connect provides the capability to filter, sort, and aggregate your API event data. This data is then presented within correlated charts, tables, and maps, to help you manage service levels, set quotas, establish controls, set up security policies, manage communities, and analyze trends. API analytics is built on the Kibana V5.5.1 open source analytics and visualization platform, which is designed to work with the Elasticsearch real-time distributed search and analytics engine.

## API Connect server requirements

From an on-premises cloud, you can create, promote, use, and track APIs. An on-premises cloud is composed of various appliances, where each appliance is a server of a specific type. The collection of servers defines your cloud and determines how to distribute the work of managing, analyzing, routing, and storing data.

Your on-premises cloud can be a combination of new and existing physical appliances and virtual appliances, or can be entirely composed of virtual appliances. The type and quantity of servers in an API Connect environment are determined by the individual needs of each enterprise, but the minimum requirement is one Management server, one Analytics server, one Gateway server, and one server to host the Developer Portal.

The API Connect on-premises cloud includes the following server types:

- **Management server.** Stores all of the cloud configuration, and controls communication between the other servers within API Connect. Manages the operations of the various servers in the API Connect cloud and provides the tools to interface with the various servers. The Cloud Manager and API Manager user interfaces run on the Management server.
- **Analytics server.** Provides analytic functions that collect and store information about APIs and API users.
- **Gateway server.** Processes and manages security protocols and stores relevant user and appliance authentication data. The Gateway server also provides assembly functions that enable APIs to integrate with various endpoints, such as databases or HTTP-based endpoints.
- **Developer Portal server.** Provides a customizable social developer portal with a full-featured content management system, and includes clustering capability. Enables API providers to build portals for their application developers, and provides the interface for application developers to discover APIs and subscribe to usage Plans contained in the published Products for use in their applications.

Note: All Management appliances in an API Connect cloud must run at the same firmware level as each other. Gateway appliances can run on different firmware levels to each other, but it is recommended that all of the Gateway appliances run on the same level as each other.

## Typical tasks per interface component

API Connect offers both command line and graphical user interfaces. Provider and consumer organizations use different interfaces for completing typical tasks. Refer to the following table to locate the interface that corresponds to a specific task.

Table 1. API Connect Tasks per interface component

Organization Type	Interface Component	Tasks
API Provider	Command Line Interface (CLI)	Create APIs, Plans, and Products
	API Manager UI	Create APIs, Plans, and Products
	API Manager UI	Create Catalogs and Spaces; Create Consumer Organizations
	Cloud Manager UI	Create Provider Organizations
API Consumer (application developer)	Developer Portal	Access APIs to create and run applications; Create Consumer Organizations

- If self-service onboarding is enabled for a Catalog, a consumer organization is automatically created when an application developer signs up or is invited by the API provider to a Developer Portal, and the application developer then becomes the owner of that consumer organization.

## Related concepts

- [Packaging strategy and terminology in API Connect](#)

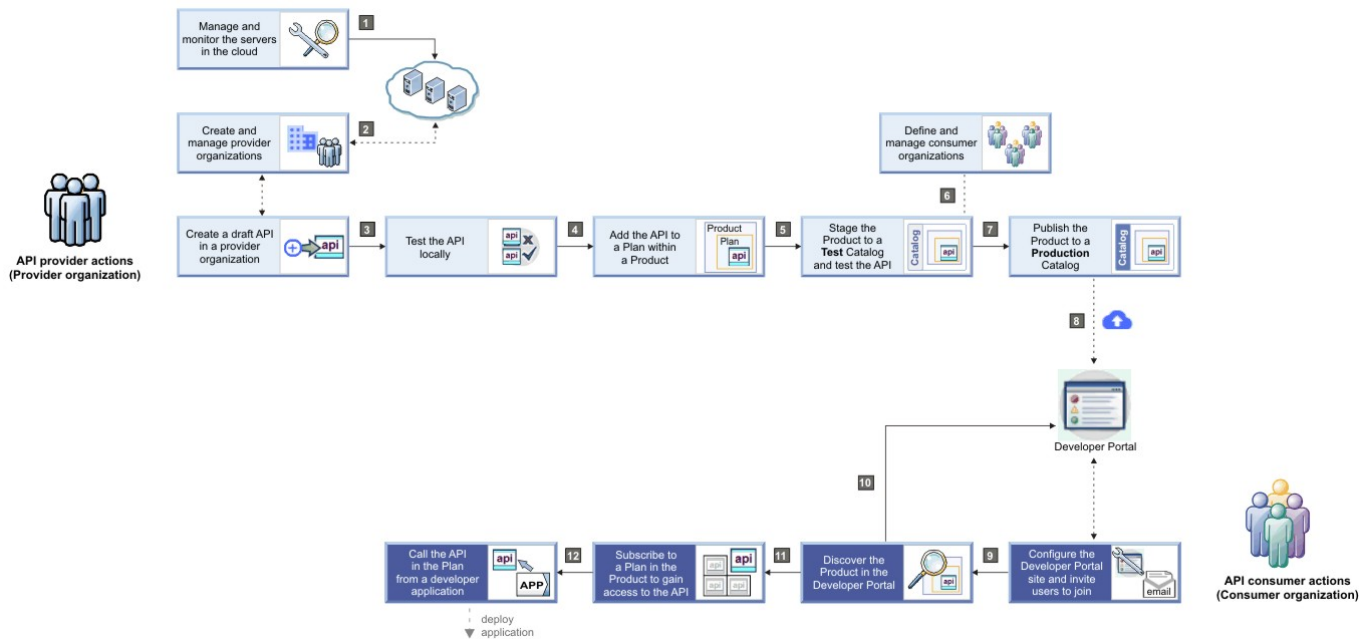
**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API Connect: End-to-end solution example

This example summarizes the concepts relating to the creation and use of APIs in the API Connect on-premises solution. It depicts the workflow and highlights some of the default roles for the tasks completed during the API lifecycle.

The following diagram shows an example of the workflow steps that are completed by the provider and consumer organizations.



## Action: 1

**Cloud Owner** **Cloud Administrator** The minimum requirements for an on-premises API Connect solution consist of one Management server to manage APIs, one Gateway server to direct API traffic, one Analytics server to analyze the APIs, and a server to host the Developer Portal. As a Cloud Owner or Cloud Administrator, you gather a collection of Management, Analytics, Gateway, and Developer Portal servers to create *clusters* to load balance and isolate traffic. A cluster has a single network address through which you can access its capabilities.

## Action: 2

**Organization Manager** **Organization Owner** With the infrastructure in place, Organization Managers and Organization Owners can manage *organizations* of users who create APIs, provider applications, and associated Products. Users belong to one or more provider organizations and individually or collectively work on the APIs or applications that belong to the organization. Project teams, departments, and company divisions are all examples of groups of users that might be members of the same provider organization in API Connect.

## Action: 3 4 5

**API Developer** Once defined as a user in a provider organization and assigned access permissions, API Developers (who might be assigned more than one role) can design, develop, and test APIs, and associate them with Plans and Products. As an API Developer, you specify policy settings to limit the usage of the APIs exposed by the Plan. You can define a single quota policy that applies to all the API resources accessed through the Plan, or you can define separate quota policies for specific API resources. You can also define policies on API resources to configure capabilities such as security, logging, routing of requests to target services, and transformation of data from one format to another. Such policies control aspects of processing in the Gateway during the handling of an API invocation, and are the building blocks of assembly flows. While developing and maintaining APIs, you can also create separate deployment targets called *Catalogs* for testing and production. Each Catalog is associated with a specific Developer Portal and endpoints. If you have administrative privileges, you can restrict deployment access to a Catalog and require actions, such as approving deployment of new API versions.

## Action: 6

**Product Manager** To control access to APIs that are ready for publication and ready to be included in applications, a Product Manager defines and manages *organizations* of users who own developer applications and call published APIs from these applications. A consumer organization is assigned an owner, and might represent a business partner, or a group of internal or external developers. Consumer organizations can also be grouped into *communities* to which one or more APIs (in their containing Plans and Products) can be collectively published. As a Product Manager, you manage access to APIs, manage the relationship between the provider organization and consumer organizations, provide support to application developers when needed, and analyze API usage.

## Action: 7 8

**API Administrator** After APIs are created and successfully tested, an API Administrator publishes one or more *Products* to expose the APIs on the Developer Portal for discovery and use. APIs are included in a *Plan*, which is contained in a *Product*, before being published, and can be published to one or more consumer organizations, thereby restricting visibility of the API. Only application developers in the specified organizations can see the API on the Developer Portal and obtain application keys to access it. The API Administrator is also responsible for managing the lifecycle of Products and their associated APIs, and uses analytics to track API usage and determine whether an API is fulfilling its intended purpose.

## Action: 9

**Consumer Organization Owner** After a consumer organization is created, its designated Consumer Organization Owner can invite other users to join the consumer organization so that they can access the Developer Portal and use the APIs that have been made available to the consumer organization. The Consumer Organization Owner, or another

user with relevant access, can also configure the Developer Portal site; for example, customize its appearance, create and control forums, post blog entries, and configure blogs.

## Action: 10 11 12

**App Developer** After a Product is published, authorized App Developers gain access to its APIs by registering applications to access the Plans in that Product. An application developer uses the Developer Portal to browse for a required API, subscribe to its associated Plan, and then includes the API in an application that can subsequently be deployed to a device.

When the API is invoked from the deployed application on a device, a sample request/response flow of the API Connect runtime interactions might be as follows:

1. The device user opens the application, which then issues the API request.
2. The request is handled by the Gateway (which performs load balancing and security validation for all API requests) and the API runtime:
  - a. The Gateway validates access policies with the API Manager and invokes the API.
  - b. The API runtime executes the API and obtains the data payload from the back-end system.
  - c. The API response is sent back to the Gateway.
  - d. The Gateway forwards the response to the calling application.
  - e. The Gateway reports analytics data to the Analytics server.

All members of the consumer organization can optionally view API analytics information relating to individual applications or the entire organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API Connect gateway types

IBM® API Connect provides two gateway types, DataPower® API Gateway and DataPower Gateway (v5 compatible).

### DataPower API Gateway

The DataPower API Gateway has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible). Where DataPower Gateway (v5 compatible) was built for flexibility, DataPower API Gateway is built specifically for the API use case, with resulting performance benefits.

DataPower API Gateway was built and optimized for the cloud. Use this gateway if you are running applications in a public or private cloud and want to expose them as APIs.

### DataPower Gateway (v5 compatible)

DataPower Gateway (v5 compatible) provides compatibility with the IBM DataPower Gateway that was provided with IBM API Connect Version 5 and earlier releases.

Consider using DataPower Gateway (v5 compatible) if you are an existing DataPower user and want to utilize your DataPower resources and knowledge.

## Gateway comparison

Table 1. Comparison of the DataPower Gateway (v5 compatible) and DataPower API Gateway

Feature	DataPower Gateway (v5 compatible)	DataPower API Gateway
Native policies	No	Yes
OAuth provider	Full OAuth 2.0 Support	Full OAuth 2.0 Support
OAuth policy	No	Yes
OpenID Connect	Supported through a template	Supported natively
Invoke policy	Yes	Yes
Custom policies	Yes	Yes
Conditional policies	if, operation-switch, switch	switch
Activity logging	Implicitly executed at the end of API assembly	Configured in the API design, outside of the API assembly.
Parse policy (threat detection)	No	Yes
Gateway extensions	Yes	Yes
Support for mutual TLS (mTLS)	Yes	Yes

## Related information

- [Registering a gateway service](#)
- [API policies and logic constructs](#)
- [Updating the gateway type for an API](#)
- [Updating the gateway type for a Product](#)
- [Specifying a gateway type for an API definition](#)
- [Specifying a gateway type for your Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## API Connect user roles

The IBM® API Connect solution provides an infrastructure, tools, and facilities that allows users to create, manage, and stage APIs. The ability to perform tasks in the API Connect user interfaces is controlled through user roles, and the permissions that are assigned to those roles.

The roles described here are the default API Connect roles. In the API Manager user interface, you can create custom roles; for more information, see: [Creating custom roles](#). You can also create custom roles in the Developer Portal user interface.

The following sections describe the roles and permissions for each of the API Connect user interfaces:

- [User roles in the Cloud Manager UI](#).
- [User roles in the API Manager UI](#).
- [User roles in the Developer Portal UI](#).

## User roles and permissions in the Cloud Manager UI

The following table describes the Cloud Manager UI user permissions as configured in the base product. Certain roles can be edited as indicated in Table 2, and custom roles can be created. For instructions on how to create custom roles for the Admin organization (Cloud Manager users), see [Creating roles in the admin organization](#).

Table 1. Cloud Manager UI permissions

Permission	Action	Meaning
Cloud Settings	View	View all items on the Cloud Manager > Settings menu (except Roles)
	Manage	Add, update, and delete all items on the Cloud Manager > Settings menu (except Roles)
Member	View	View members on the members list located at Cloud Manager > Members
	Manage	Add and invite members from Cloud Manager > Members
Org	View	Org:View is a permission assigned to all Roles in Cloud Manager. It does not provide access to any functionality. It allows a user to activate their membership. It is the only permission in the Member role.
Provider-Org	View	View the list of provider organizations at Cloud Manager > Provider Organizations
	Manage	Add, edit, and delete provider organizations and invite owners from Cloud Manager > Provider Organizations
Settings	View	View the items on the Cloud Manager > Resources menu plus Roles located at Cloud Manager > Settings > Roles
	Manage	Add, edit, and delete the items on the Cloud Manager > Resources menu plus Roles located at Cloud Manager > Settings > Roles
Topology	View	View the items on the Cloud Manager > Topology menu
	Manage	Add, edit, and delete the items on the Cloud Manager > Topology menu

The following table lists the various Cloud Manager UI roles and the permissions assigned to them.

Table 2. Cloud Manager UI roles

Role	Permissions	Actions	Default role provides access to	Notes
Owner	All permissions	All actions	All menus	Cannot be modified or deleted.
Administrator	All permissions	All actions	All menus	Can be modified and deleted.
Member	Org	View	Membership activation only	Cannot be modified or deleted. Member role is automatically assigned to all users when they activate their membership from the invitation. It allows them to activate but does not provide access to any menus.
Organization Manager	Org	View	N/A	Can be modified and deleted.
	Provider-Org	View, Manage	Provider Organizations menu	
Topology Administrator	Org	View	N/A	Can be modified and deleted.
	Topology	View, Manage	Topology Menu	
	Settings	View, Manage	Resources menu plus Settings > Roles	
Viewer	All permissions	View	All menus, view only	Cannot be modified or deleted.

## User roles and permissions in the API Manager UI

The following tables describe the API Manager UI user permissions.

A user with Roles permission can change the permission assignments, and can create custom roles; for more information, see [Creating custom roles](#) in the section, [Managing your APIs](#).

Note: In API Manager, the Owner role has full access and cannot be edited or deleted. All other roles, including custom roles, can be deleted. If you delete a role, users lose that role. If a user loses that role, their account remains in API Manager, enabling you to add a role to the user at a future date.

Table 3. Organization permissions

Permissions	Action	Permits the member to
Member	View	View organization's members



Permissions	Action	Permits the member to
	Manage	Manage organization's members
Settings	View	View organization's configuration settings, including roles, TLS profiles, and user registries. View configuration settings for a Catalog or Space, including policies and OpenAPI extensions.
	Manage	Manage organization's configuration settings, including roles, TLS profiles, and user registries. Manage configuration settings for a Catalog or Space, including policies and OpenAPI extensions.
Topology	View	View topology permissions for configuring portal and gateway services. Also view the clusters.
	Manage	Manage topology permissions for configuring portal and gateway services. Also manage the clusters.
Org	View	View organization
Product-Drafts	View	View draft APIs and Products
	Edit	View draft APIs and edit draft Products
Api-Drafts	View	View draft APIs
	Edit	Edit draft APIs and view draft Products
Product	View	View Products
	Stage	Stage Product
	Manage	Manage Product
Product-Approval	View	View Product lifecycle changes
	Stage	Approve the staging of a Product
	Publish	Approve the publishing of a Product
	Supersede	Approve the superseding of a Product
	Replace	Approve the replacement a Product
	Deprecate	Approve the deprecation of a Product
	Retire	Approve the retiring of a Product
Consumer-Org	View	View consumer organization and developers
	Manage	Manage consumer organization and developers
App	View	View both production and development applications.
	Manage	Manage both production and development applications. A member with this permission can also request the promotion of a development app to a production app. This request triggers a task that needs approval by a member with the App-approval Manage permission.
App-dev	Manage	Manage development applications. This permission does not include the ability to manage production apps.
App-Approval	View	View application approvals, for requests to promote a development app to a production app.
	Manage	Manage (Approve or Decline) requests for approval to promote a development app to a production app.
Subscription	View	View application Plan subscriptions that have been created by application developers in the Developer Portal.
	Manage	Manage the application Plan subscriptions that have been created by application developers in the Developer Portal. The Manage permission includes ability to migrate a subscription to another plan.
Subscription-Approval	View	View application Plan subscription approvals
	Manage	Manage (Approve or Decline) application Plan subscriptions
Api-Analytics	View	View analytics
	Manage	Manage analytics
Child	View	At the provider organization level, view Catalogs in the provider organization. At the Catalog level, view Spaces in the Catalog.
	Create	At the provider organization level, create Catalogs in the provider organization. At the Catalog level, create Spaces in the Catalog.
	Manage	At the provider organization level, manage Catalogs in the provider organization. At the catalog level, manage Spaces in the Catalog. Management tasks including deleting a Catalog or Space, or transferring ownership of a Catalog or Space.

A user with Settings > Manage permission can change the permission assignments, and can create custom roles; for more information, see [Creating custom roles](#) in the section, [Managing your APIs](#).

Table 4. Default API Manager UI roles and the default permissions assigned to those roles.

Role	Role description	Permissions	Actions
Organization Owner	A provider organization owner has the full set of access permissions to API Connect functions, and also commission APIs and tracks their business adoption.	All permissions	All actions.
Administrator	A provider organization administrator has, by default, the full set of access permissions to API Connect functions, and also commission APIs and tracks their business adoption.	All permissions	All actions.
API Administrator	API administrators manage the lifecycle of APIs and publish APIs for discovery and use.	All permissions	All actions except cannot manage the following permissions: Member, Settings, Topology, and Child.
Community Manager	A community manager manages the relationship between the provider organization and application developers, provides information about API usage, and provides support to application developers.	Member	View
		Settings	View
		Topology	View gateway services or portal services at the provider organization.
		Org	View

Role	Role description	Permissions	Actions
		Drafts	View, Edit
		Product	View
		Product-approval	View
		Consumer-org	View, Manage
		App	View, Manage
		App-dev	Manage
		App-approval	View, Manage
		Subscription	View, Manage
		Subscription-approval	View, Manage
		Api-analytics	View, Manage
		Child	View
Developer	API developers design and develop APIs and applications for the provider organizations to which they belong. Note: The Developer role allows the creation of Products and APIs, and the staging and publishing of Products to a Catalog or Space, when assigned to a user at the provider organization level, but <b>not</b> when assigned to a user who is a member only of a Catalog or Space within a provider organization. A Developer in a Catalog or Space can, however, manage Products that have been staged or published to the Catalog or Space.	Member	View
		Settings	View
		Topology	View gateway services or portal services at the provider organization.
		Org	View
		Drafts	View, Edit
		Product	View, Stage, Manage
		Product-approval	View, Stage, Publish, Supersede, Replace, Deprecate, Retire
		Consumer-org	View
		App	View, Manage
		App-dev	Manage
		App-approval	View, Manage
		Subscription	View, Manage
		Subscription-approval	View, Manage
		Api-analytics	View, Manage
		Child	View, Create
Member	Member of a provider organization	Org	View
Viewer	Viewer of a provider organization	Member	View
		Topology	View gateway services or portal services at the provider organization.
		Org	View
		Drafts	View
		Product-approval	View
		Consumer-org	View
		App	View
		App-approval	View
		Subscription	View
		Subscription-approval	View
		Api-analytics	View
		Child	View

Note: In API Manager, the Organization Owner role has full access and cannot be edited or deleted. All other roles, including custom roles, can be deleted. If you delete a role, users lose that role. If a user loses that role, their account remains in API Manager, enabling you to add a role to the user at a future date.

## User roles in the Developer Portal UI

The following table describes the various Developer Portal UI roles that relate to working with APIs and applications. In addition, you can create custom roles for the Developer Portal site itself.

Table 5. Developer Portal UI roles

Role	Role Description	Permission	Actions
Owner	Owns and administers the app developer organization	Organization member	View, Manage
		Organization settings	View, Manage
		Organization view	View
		Consumer product	View
		Consumer app	View or Manage production or development applications
		Consumer app-dev	Manage development applications
		Consumer subscription	View or Manage the application Plan subscriptions that have been created by application developers in the Developer Portal. The Manage permission includes ability to migrate a subscription to another plan.
		Consumer app-analytics	View application analytics
Administrator	Administers the app developer organization	Organization member	View, Manage
		Organization settings	View, Manage
		Organization	View
		Consumer product	View
		Consumer app	View, Manage production or development applications
		Consumer app-dev	Manage development applications
		Consumer subscription	View or Manage the application Plan subscriptions that have been created by application developers in the Developer Portal. The Manage permission includes ability to migrate a subscription to another plan.
		Consumer app-analytics	View application analytics
Developer	Builds and manages apps in the developer organization	Organization member	View
		Organization	View
		Consumer product	View
		Consumer app	View or Manage production or development applications.
		Consumer app-dev	Manage development applications
		Consumer app-analytics	View
Member	Member of the app developer organization	Organization	View
Viewer	Viewer of the app developer organization	Organization member	View
		Organization settings	View
		Organization	View
		Consumer product	View
		Consumer app	View applications
		Consumer production-app	View production applications
		Consumer app-analytics	View application analytics

Note: A user called admin is created automatically, that has full administrator access to the Developer Portal site. The admin user can view Products and APIs but has no access to use APIs. The admin user assumes the email address of the owner of the provider organization associated with the Developer Portal.

## Related information

- [Managing your cloud](#)
- [Managing your APIs](#)
- [Developer Portal: discover and use APIs](#)
- [Administering provider organizations](#)
- [Administering consumer organizations](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## API Connect support

If you experience a problem with IBM® API Connect that you cannot resolve, you can check the IBM Support website for the most recent technical bulletins and fixes. Otherwise, you can contact IBM Support.

For the most recent API Connect technical information, see the [IBM Support Community](#).

If you cannot find a resolution, before you contact IBM Support, you must gather information about the problem. For details of the information you must gather before contacting IBM Support, see [MustGather - Collecting data: API Connect problem determination](#).

Providing IBM Support with a good description of the problem together with the details of your API Connect configuration, helps to expedite the problem resolution.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## API Connect glossary

The IBM® API Connect and Cloud Manager glossary of terms and definitions.

### API Administrator (role)

Manages the API product lifecycle for the Provider Organizations for which they are a member.

### API Event

An event captured for use by API Connect Analytics or third-party analytics, such as response time, HTTP response code, payload of the request and response body, and so on. An API event is logged each time an API operation is invoked via the Gateway server.

### API gateway

Service that acts as a single entry point or “front door” for provider APIs and back-end services. An API Gateway accepts and processes concurrent API calls, and performs traffic management, authorization and access control, monitoring, and API version management. See also [IBM DataPower Gateway](#).

### API Manager

Graphical tool in API Connect that enables you to manage Catalogs, Spaces, and APIs, users and roles in the Provider organization, consumer organizations and communities, publish Products to the Developer Portal, and analyze API usage.

### API Designer

Graphical tool that runs locally on a laptop or desktop system that enables you to create and modify APIs and LoopBack® apps, and publish them to IBM Cloud or container runtimes.

### API operation

REST API call consisting of an HTTP verb and a URL path (endpoint). For example, `GET http://myserver.com/api/users` that returns a list of users.

### Application

Software that consumes (calls) an API. One or more Applications are registered by a consumer organization to subscribe to APIs. The application is allocated client ID and client secret credentials that it supplies when invoking API calls.

### assembly

Makes side calls to external services and then transforms and aggregates the response before a response is relayed to the calling application.

### Availability zones

Isolated locations within data centers from which public cloud services originate and operate. Businesses choose one or multiple worldwide availability zones for their services depending on business needs. Availability Zones provide high availability for services by providing redundancy across regions.

### Availability zone (primary)

For Kubernetes form factor, contains the management service. Other zones can contain other services such as the Developer Portal, but not the management service.

### Catalog

A staging target that behaves as a logical partition of the gateway and the Developer Portal. The URLs for API calls and for the Developer Portal are specific to a particular Catalog.

### client ID

A piece of information that identifies an individual application. An application can invoke an API only if it passes an application key that is recognized by the IBM API Connect system and is granted access to the API. The application key is passed by the client by using an HTTP query parameter.

### client secret

A piece of information used together with the application key to verify the identity of an application. An API can be configured to require that client applications supply their client secret with their client ID. The client secret functions effectively as a password known only to the application. The client secret is passed by the client by using an HTTP query parameter.

### Cloud Manager (tool)

Graphical tool in API Connect on-premises installation that enables you to define servers, administer and scale system resources, monitor runtime health and create Provider organizations. The Cloud Administrator is the primary user of the Cloud Manager.

### Cloud Administrator (role)

Manages the configuration of resources, regions, and availability zones for the Admin Organization.

### cluster

A collection of one or more servers that provide a specific function.

### Cold standby

A deployment configuration in which failover servers are in a stopped state until a failover is required, resulting in a longer time window to restore the normal operation of the solution.

### community

A collection of consumer organizations. It is used as a grouping construct when publishing APIs. Communities are used to restrict the visibility and accessibility of APIs. An API can be published to selected communities, which means that only application developers within those organizations can see the API.

### Community Manager (role)

Manages application developer communities for the Provider Organizations for which they are a member.

### Consumer organization

Within a catalog, representation of a business entity that wishes to consume APIs exposed by the Catalog. For example, a third-party application development company would register themselves as a consumer organization (via the Developer Portal). Each developer in their company can then be registered as a user inside that consumer organization.

**IBM DataPower® Gateway**

API Gateway service component of API Connect that helps provide security, control, integration and optimized access to APIs. Due to its security and hardening, it is well-suited for a deployment in the demilitarized zone (DMZ) for externally-facing production scenarios. Available in physical, virtual, cloud, Linux and Docker form factors. There are two gateway types, DataPower Gateway (v5 compatible) and DataPower API Gateway; for details, see [API Connect gateway types](#).

**Developer (role)**

Creates and configures APIs, Products, and policies for the Provider Organizations for which they are a member. An API Developer can be a member of one or more Provider Organizations. The API Developer focuses on the technical implementation of APIs more than they do on the business relationship with application developers.

**Development Catalog**

Catalog used for testing APIs that are under development and in which approvals are bypassed for publishing and lifecycle actions. Pending approvals are canceled when a non-Development Catalog is converted to a Development Catalog.

**Developer Portal**

Component of API Connect that provides a customizable graphical web portal for developers to discover APIs, register applications that consume APIs, subscribe to usage plans, and test and use APIs.

**Developer Toolkit**

Locally-installed package that includes API Designer and the `apic` command-line tool that enables you to create, edit, manage, and publish APIs and apps.

**Gateway**

See [API gateway](#).

**Gateway service**

API Connect service that provides API gateway functionality, such as microgateway or IBM DataPower.

**Hot standby**

A deployment configuration in which a set of servers are actively running ready to instantaneously take over in the event of a failure, but not actively serving traffic until that failover.

**Identity provider**

Provides identifiers for users looking to interact with a system, assert to such a system that such an identifier presented by a user is known to the provider, and possibly provide other information about the user that is known to the provider. API Connect supports LDAP, AuthURL, and Local User Registries.

**Integrations**

Third-party tools to build on and improve your API Connect workflow; for example, Slack, Auth0, etc.

**Integration profile**

Standard API that you can use to integrate with management services for things like identity, notifications, API analytics, and so on.

**Keystore**

A repository of security certificates, either authorization certificates or public key certificates, and corresponding private keys, used in SSL encryption. The Keystore file has a `.jks` extension.

**LoopBack model**

A JavaScript object that represents application data and includes validation rules, data access capabilities, and business logic. LoopBack models provide a REST API by default, and connect to data sources for access to back-end data

**LoopBack data source**

A JavaScript object that represents a back-end service such as a database, REST API (to be consumed), or SOAP web service. Data sources are backed by connectors that communicate directly with the database or other back-end service.

**Management server**

Stores all of the cloud configuration, and controls communication between the other servers within API Connect.

**Management service**

Consists of one or more Management servers.

**Member (role)**

The default role for all users in both the Admin and Provider Organization. All users are assigned the Member role in addition to other roles they may require.

**microservice**

Modular, independently-deployable application service that communicates with other microservices through a REST API. Microservices are typically organized around capabilities, for example, recommendation, inventory, shipping, or billing.

**Mutual authentication**

Process in which both entities on a network authenticate each other. In a network environment, the client authenticates the server and vice-versa. It is optional for TLS. Also called two-way authentication.

**Notification settings**

How you configure notifications for API Connect users (API providers and consumers).

**Notification services**

Integrations that provide notification capabilities, such as email.

**Notification templates**

The configuration of the message format and wording for a notification service.

**OAuth provider**

The OAuth provider supplies the OAuth authentication for logins. API Connect supports both Native and Third Party OAuth providers. Some common third-party OAuth providers are Google and Facebook.

**OpenAPI Components**

Part of API specification that contains a set of reusable objects for aspects of an API specification, such as schemas, responses, requestBodies, and headers. For more information, see [OpenAPI v3 specification](<https://github.com/OAI/OpenAPI-Specification/blob/OpenAPI.next/versions/3.0.md#components-object>).

**organization**

The entity that owns APIs or applications that use APIs. A provider organization owns APIs and associated Plans, and can additionally own applications. A consumer organization owns only applications. An organization has at least one owner. An organization can be a project team, department, or division.

**Organization Manager (role)**

The Organization Manager manages Provider Organizations for the Admin Organization.

**Path**

Defines the route through which users access REST APIs. A path consists of one or more HTTP operations such as GET or POST.

**Plan**

The packaging construct by which APIs are made available to developers. A Plan makes a collection of operations from one or more APIs available, and is published to communities of application developers. Application developers gain access to APIs by registering applications to access Plans. A Plan carries with it a collection of policy settings. In the simplest form, a Plan defines a single quota policy that applies to all the API operations that are accessed through the Plan. In more advanced cases, additional policies can be associated with a Plan.

**policy**

A configuration that controls a specific aspect of processing in the Gateway server during the handling of an API invocation at run time. Policies are the building blocks of assembly flows. Policies provide the means to configure capability, such as security, logging, routing of requests to target services, and transformation of data from one format to another. Policies can be configured in the context of an API or in the context of a Plan.

**Product**

Provide a method by which you can group APIs into a package that is intended for a particular use. Additionally, they contain Plans, which can be used to differentiate between different offerings. You can create Plans only within Products, and these Products are then published in a Catalog.

**Provider Organization Owner**

Owns and administers API provider organizations, manages application developer communities, authors APIs and defines products, manages the API product lifecycle. Owners are invited by the Cloud Administrator to join API Connect as an owner of a provider organization.

**proxy**

Application programming interface that forwards requests to a user-defined back-end resource and relays responses back to the calling application.

**Region**

A physical location (site or data center) hosting infrastructure isolated from other locations, with independent power and networking connectivity. A region may or may not have further sub-isolation characteristics such as semi-independent pods or availability zones.

**Resources (User registries, TLS Profiles, Notifications)**

Resources supply necessary functions for the API Connect cloud, such as user authentication, SSL security, and sending system-generated emails.

**role**

Defines permissions that can enable functionality for users. Each role has a different set of permissions.

**security definition**

Specifies all the settings for a particular aspect of API security; for example, the user registry that you use to authenticate access to the API.

**server**

A single appliance, such as an IBM WebSphere IBM DataPower appliance.

**service**

A user-configurable element implemented through one or more server processes in the API Connect runtime, such as microgateway, Analytics, IBM DataPower, and Developer Portal.

**SNMP**

Simple Network Management Protocol (SNMP) is a popular protocol for network management. It is used for collecting information from, and configuring, network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP) network. The SNMP hosts are configured in the API Connect Cloud Manager.

**Space**

A subdivision of a Catalog. Each Space is used by a different API provider development team and has its own set of management capabilities relating specifically to the APIs that the associated team publishes to that Space, enabling each team to manage their APIs independently.

**SSL (TLS) Profile**

An SSL/TLS profile is used to secure the transmission of data through web sites. SSL certificates guarantee that information you submit to web sites will not be stolen or tampered with.

**subscription**

The means by which an application developer gains access to the resources provided by an API. An application developer uses the Developer Portal to subscribe to the plan in which the API is published.

**TLS**

Transport Layer Security - a cryptographic protocol that provides secure communication over a network to prevent eavesdropping and tampering. It is a successor to SSL and runs over TCP.

**TLS Profiles**

The Cloud Manager and API Manager use TLS profiles to secure transmission of data through web sites. TLS certificates guarantee that information you submit to web sites will not be stolen or tampered with. A TLS Profile consists of Server name, Protocol used (SSL or TLS), and whether Mutual Authentication is required.

**Toolkit**

See Developer Toolkit.

**Topology administrator (role)**

Configures gateway services for the Admin Organization.

**Truststore**

Stores certificates from trusted Certificate authorities(CA) which are used to verify certificate presented by Server in SSL/TSL Connection. While a keystore stores a server's credentials, the truststore stores certificates from a third-party CA.

**User registry**

A database or other collection containing credentials for users such as provider organization members, consumer organization members, and API Connect administrators. The users are authenticated by an identity provider such as LDAP. User registries authenticate users at login time when accessing the Cloud Manager or API Manager applications. User registries are also used to protect APIs so that user credentials must be supplied when an API is called.

**vendor extension**

An extension to OpenAPI (Swagger) specification required by a particular use case.

**Visibility**

Setting visibility determines whether a Provider Organization has access to an Availability Zone, or other service.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Accessibility features for IBM API Connect

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

### Accessibility features

The following list includes the major accessibility features in IBM® App Connect Enterprise. You can use screen-reader software to hear what is displayed on the screen.

- Supports keyboard-only operation.
- Supports interfaces commonly used by screen readers.

Tip: This product documentation, and its related publications, are accessibility-enabled for the IBM Home Page Reader. You can operate all features by using the keyboard instead of the mouse.

If you are reading a PDF file with a screen reader, the default reading option typically returns the best results. In some cases, how the PDF file is generated might require you to select one of the other reading options. For example, Use reading order in raw print stream.

---

## Keyboard navigation

---

This product uses standard Linux® and Microsoft Windows navigation keys.

For more information about the commitments that IBM makes towards accessibility, see the [IBM Accessibility Center](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Legal information

---

Notices, and terms and conditions for information centers.

- [License updates](#)  
IBM API Connect v2018 is changing the way licenses are created to better serve customers in these dynamic times and reduce the number of licenses that need to be managed.
- [Tracking API volume for auditing and compliance](#)  
For client security reasons, IBM entrusts its clients with monitoring their own API volume and ensuring that it is within the limits of the contract.
- [Notices](#)  
This information was developed for products and services offered in the U.S.A.
- [Terms and conditions for information centers](#)  
Permissions for the use of these publications are granted subject to the following terms and conditions.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## License updates

---

IBM API Connect v2018 is changing the way licenses are created to better serve customers in these dynamic times and reduce the number of licenses that need to be managed.

Currently, licenses follow the product numbering standard based on IBM V.R.M.F (Version.Release.Mod.FixPack) structure and API Connect v2018 creates a new license for every fix pack.

Starting in June 2022, API Connect v2018 future release licenses will be updated to use the single V.R.M Mod-level license that covers all fixes and security roll-ups underneath it. For example, the next license covering the 2018.4.1.x stream will be listed as "2018.4.1". This will be the master mod-level license for all Fix Pack and security roll-up releases in this stream. New licensing applies to v2018.4.1.20 and later. There are no changes to license terms, only updates to open-source notices documented in the notes for each.

Note: For v2018.4.1.20, some API Connect components, such as Analytics, still use the v2018.4.1.19 license even though the platform and most components use the newer license. The use of the v2018.4.1.19 license for the affected components will not cause installation problems. All components of API Connect v2018.4.1.20 are covered by the 2018.4.1 license.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tracking API volume for auditing and compliance

---

For client security reasons, IBM entrusts its clients with monitoring their own API volume and ensuring that it is within the limits of the contract.

---

## Usage archiving

---

You should periodically record the number of your API calls from the Analytics tool to maintain a record of your yearly usage. To ensure the most accurate results, you should capture the counts of the API Calls on a daily basis. For information about creating a query that counts API calls for different response status codes and for a specified time range, see [Counting total API calls for your Analytics service](#).

---

## Overage charging

---

If you exceed your allotted usage, it is your responsibility to report this to IBM in the form of a CSV file so the correct overage charge can be applied. IBM has the right to audit customer usage data at any time. If unreported overages are found, there are severe penalties.

Overage is based on the measurement period. For example, overage for a contract might be measured by the year. If your API volume exceeds the entitlement after 6 months, you must either pay overages for the remaining period or purchase additional volume.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
1623-14,  
Shimotsuruma,  
Yamato-shi  
Kanagawa  
242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England  
SO21 2JN

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.



This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Copyright and trademark information page at <https://www.ibm.com/legal/copytrade.shtml>.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names might be trademarks of IBM or other companies.

## Privacy Policy Considerations

---

IBM Software products, including software as a service solutions, (*Software Offerings*) may use cookies or other technologies to collect product usage information, to help improve the user experience, to tailor interactions with the user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth in the following paragraphs.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's session ID for purposes of session management, or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <https://www.ibm.com/privacy> and IBM's Online Privacy Statement at <https://www.ibm.com/privacy/details> the section entitled *Cookies, Web Beacons and Other Technologies* and the *IBM Software Products and Software-as-a-Service Privacy Statement* at <https://www.ibm.com/software/info/product-privacy>.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Terms and conditions for information centers

Permissions for the use of these publications are granted subject to the following terms and conditions.

### Applicability

---

These terms and conditions are in addition to any terms of use for the IBM® website.

### Personal use

---

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### Commercial use

---

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

### Rights

---

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the aforementioned instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## IBM API Connect Considerations for GDPR Readiness

Information about features of IBM® API Connect that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness.

### For PID(s): 5725-Z22 5725-Z63

---

#### Notice:

---

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of API Connect that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

**Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.**

**The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.**

#### Table of Contents

---

1. [GDPR](#)
2. [Product Configuration for GDPR](#)
3. [Data Life Cycle](#)
4. [Data Collection](#)
5. [Data Storage](#)
6. [Data Access](#)
7. [Data Processing](#)
8. [Data Deletion](#)
9. [Data Monitoring](#)
10. [Capability for Restricting Use of Personal Data](#)

## GDPR

---

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

#### Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

#### Read more about GDPR

- [EU GDPR Information Portal](#)
- [ibm.com/GDPR website](http://ibm.com/GDPR)

## Product Configuration - considerations for GDPR Readiness

---

The following sections provide considerations for configuring API Connect to help your organization with GDPR readiness.

#### Configuration to support data handling requirements

The GDPR legislation requires that personal data is strictly controlled and that the integrity of the data is maintained. This requires the data to be secured against loss through system failure and also through unauthorized access or via theft of computer equipment or storage media.

IBM API Connect stores identity data in a local database. This encompasses both clients' employee identity data and end users' identity data. Direct access to this database is not available. This data is encrypted by default in IBM API Connect Version 5.0 - refer to [Disk encryption](#) for details. Identity information collected is protected in transit, refer to [TLS profiles](#) for details on configuring TLS profiles.

API Connect supports a variety of user registry types for authenticating users. Refer to [Authenticating by using your enterprise user registry](#) for details. When using a local user registry, passwords are stored in encrypted form in the local API Connect database. If you want alternative password management, leverage a non-user registry option to manage passwords.

Administrators, that you define, can view identity information. Administrators can take backups that include identity information. It is your responsibility to protect these backups.

A core component for an API Connect deployment are gateways. Refer to [API Gateways](#) for details about gateways. DataPower® Gateways are commonly leveraged, refer to [DataPower Gateway Version 7.7 Documentation](#) for details on DataPower Gateways. Refer to the DataPower Gateway deployment guidelines document for considerations for configuring DataPower Gateways to help your organization with GDPR readiness.

Configuration to support Data Privacy

For Developer Portal, you can customize the privacy policy statement, refer to [Customizing the privacy policy statement](#) for details.

Configuration to support Data Security

To learn about securing your solution, use the API Connect product documentation ([https://www.ibm.com/support/knowledgecenter/en/SSMNE5\\_5.0.0](https://www.ibm.com/support/knowledgecenter/en/SSMNE5_5.0.0)) and search for "security".

## Data Life Cycle

---

GDPR requires that personal data is:

- Processed lawfully, fairly and in a transparent manner in relation to individuals.
- Collected for specified, explicit and legitimate purposes.
- Adequate, relevant and limited to what is necessary.
- Accurate and, where necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate personal data are erased or rectified without delay.
- Kept in a form which permits identification of the data subject for no longer than necessary.

### What is the end-to-end process through which personal data go through when using our offering?

API Connect collects and stores identity information, including first and last name, and email address, for the purposes of user registration. Cloud Manager and API Manager accounts are for your employees (or designated actors). Developer Portal accounts are for your consumers of your APIs. Identity information can be collected directly from users or can be copied from LDAP registries. In situations where non-local user registries are used, only email address is copied from LDAP registry. Developer Portal user accounts can be deleted - refer to [Deleting your Developer account](#) for details. Cloud Manager and API Manager user accounts' identity information can be anonymized by users.

Users of the API Manager UI can publish Products and APIs to the Developer Portal for Application Developers to access and use. Refer to [Developer Portal: discover and use APIs](#) to learn about Developer Portal. Developer Portal accounts are for consumers of your APIs. You can define and customize a terms and conditions statement that your users must accept before they can register to use your Developer Portal - refer to [Customizing the terms and conditions statement](#) for details.

API Connect optionally logs information related to API invocations. This capability in API Connect is known as API Analytics. Refer to [API Analytics](#) for details about API Analytics.

The API Analytics log information can optionally include unknown / unclassified information such as query headers and request and response information related to API calls - you control defining the APIs and data associated with API invocations. To disable API Analytics, refer to [Enabling or disabling access to analytics event data in API Connect](#). Logging preferences can be configured at the API level, refer to [Activity Log](#) for details.

The retention period for Analytics data is configurable - refer to [Specifying the cloud settings](#) for details. Backup capability for this information is not available.

API Connect logs collect technical information related to service use including tracing of service execution and sequences of operation use. Other technical data related to service use includes data values that define the mechanisms used to connect to the service, for example, IP address. This data is collected for debugging and service improvement. Service diagnostics are collected during unexpected or error situations to allow the offering team to correct the situation and hopefully prevent it from occurring in the future. There is no direct access available to these logs. These logs are managed by API Connect and rollover based on size and time criteria. The logs can be downloaded from the system, refer to [Gathering postmortem information about your servers](#) for details.

API Connect can generate audit events. An audit event is logged from each management node when there are changes to the API lifecycle or to the organization. For example, publishing a product or creating an organization would trigger this event. The audit event record contains information about the changes to the API lifecycle or organization. Refer to [Audit event fields](#) for details. The retention for these events is the Analytics retention period.

## Data Collection

---

Developer Portal accounts are for consumers of your APIs. You can define and customize a terms and conditions statement that your users must accept before they can register to use your Developer Portal - refer to [Customizing the terms and conditions statement](#) for details. You can customize the privacy policy statement for Developer Portal, refer to [Customizing the privacy policy statement](#) for details.

### Types of Data Collected

API Connect collects and stores identity information, including first and last name, and email address, for the purposes of user registration. Cloud Manager and API Manager accounts are for your employees (or designated actors). Developer Portal accounts are for your consumers of your APIs. Identity information can be collected directly from users or can be copied from LDAP registries. In situations where non-local user registries are used, only email address is copied from LDAP registry. Developer Portal user accounts can be deleted - refer to [Deleting your Developer account](#) for details. Cloud Manager and API Manager user accounts' identity information can be anonymized by users.

You can customize the privacy policy statement for Developer Portal, refer to [Customizing the privacy policy statement](#) for details.

## Data Storage

---

Identity data is stored in API Connect local data store. There is no direct access available to this data store.

API Analytics leverages Elasticsearch real-time distributed search and analytics engine for storage of logged data. There is no direct access available to this data store.

Identity data is included in backups, refer to [Creating a backup of an API Connect configuration](#) for details on taking backups. It is your responsibility to protect and discard backups.

## Data Access

---

Identity information can be viewed by administrators that you define.

Analytics information can be accessed via a variety of means. Refer to [Viewing and exporting analytics and API event data](#) and [Analytics in the Developer Portal](#) for details.

Analytics information can be offloaded to third party systems. Refer to [Specifying the cloud settings](#) for details.

Technical information related to service use is collected in logs. The logs can be downloaded from the system, refer to [Gathering postmortem information about your servers](#) for details. These logs are managed by API Connect and rollover based on size and time criteria. Downloaded logs can be provided to IBM Support for use in problem determination.

API Connect can generate audit events. Refer to [Audit event fields](#) for details. Audit events can be offloaded to third party systems, refer to [Configuring the offload of analytics event data to third-party systems](#) for details. Administrators can view audit events as notifications - refer to [Viewing information about activities](#) for details. Audit events can be emitted as syslog messages, refer to [Syslog auditing and your cloud](#) for details.

API Connect logs collect technical information related to service use including tracing of service execution and sequences of operation use. Other technical data related to service use includes data values that define the mechanisms used to connect to the service, for example, IP address. This data is collected for debugging and service improvement. Service diagnostics are collected during unexpected or error situations to allow the offering team to correct the situation and hopefully prevent it from occurring in the future. There is no direct access available to these logs. The logs can be downloaded from the system, refer to [Gathering postmortem information about your servers](#) for details. These logs are managed by API Connect and rollover based on size and time criteria.

## Data Processing

---

Data collected by API Connect or to gateways via API invocations is protected by TLS in transit. Refer to [TLS profiles](#) for details.

Data is stored in API Connect local database on the API Connect appliances. There is no direct access available to this data. This data is encrypted by default in IBM API Connect Version 5.0 - refer to [Disk encryption](#) for details.

Cloud Manager and API Manager administrators (defined by you) have read access to identity data.

## Data Deletion

---

### Right to Erasure

Article 17 of the GDPR states that data subjects have the right to have their personal data removed from the systems of controllers and processors - without undue delay - under a set of circumstances.

### Data Deletion characteristics

Users can delete Developer Portal user accounts - refer to [Deleting your Developer account](#) for details. Cloud Manager and API Manager user account identity data can be anonymized by the users thus deleting users association to the account data.

Technical information related to service use collected in logs is rolled over based on size and time criteria.

To disable API Analytics, refer to [Enabling or disabling access to analytics event data in API Connect](#). API Analytics data retention period is configurable - refer to [Specifying the cloud settings](#) for details. IBM Support personnel can delete API Analytics data, this capability is only available through screen sharing with your authorized personnel. Analytics information can be offloaded to other systems - refer to [Specifying the cloud settings](#) and [Syslog auditing and your cloud](#) for details. You are responsible for protection and discarding of offloaded data.

Identity information for accounts is included in system backups. You manage the deletion of system backups.

## Data Monitoring

---

Customers should regularly test, assess, and evaluate the effectiveness of their technical and organizational measures to comply with GDPR. These measures should include ongoing privacy assessments, threat modeling, centralized security logging and monitoring among others.

API Connect can generate audit events. An audit event is logged from each management node when there are changes to the API lifecycle or to the organization. For example, publishing a product or creating an organization would trigger this event. The audit event record contains information about the changes to the API lifecycle or organization. Refer to [Audit event fields](#) for details. Audit events can be offloaded to a third party system, refer to [Configuring destination targets for API Connect analytics data](#) for more information. Audit events can be emitted as syslog messages - refer to [Syslog auditing and your cloud](#) for details.

## Capability for Restricting Use of Personal Data

---

Users of the API Manager UI can publish Products and APIs to the Developer Portal for Application Developers to access and use. Refer to [Developer Portal: discover and use APIs](#) to learn about Developer Portal. Developer Portal accounts are for consumers of your APIs. You can define and customize a terms and conditions statement that your users must accept before they can register to use your Developer Portal - refer to [Customizing the terms and conditions statement](#) for details. You can customize the privacy policy statement for Developer Portal, refer to [Customizing the privacy policy statement](#) for details.

Developer Portal users can modify their own account information, and delete their account.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Essential reading

These articles by IBM® API Connect product specialists provide a wealth of supporting information on APIs and the API economy.

### [Why Become a Digital Business?](#)

Executing a Digital Transformation to become a Digital Business is among the hottest initiatives now – crossing both business and IT. But, is this just the latest buzzword or is there something to this that is different? Do you know **why** you should become a “Digital Business”?

### [Creating A Digital Ecosystem – Past, Present, and Future](#)

The ability to create a digital ecosystem is critical to digital transformation success. Your success is your network reach.

### [Agile integration](#)

Your business needs a modern, agile approach to integration. It should empower extended teams to create integrations, leverages a complete set of integration styles and capabilities, and increases overall productivity.

### [What is an API? and What is the API Economy?](#)

Businesses start to consider APIs and the API Economy at various times. Many are long down their API journeys, while others are still considering whether to start. This article looks at some of the basics that companies considering APIs might want to know.

### [What is API Management?](#)

APIs are not new. Software and hardware have had APIs (Application Programming Interfaces) for decades. However, *having* an API and *managing* an API are not the same thing.

### [Providing APIs or Managing APIs – There is a Big Difference](#)

A discussion of the potential for confusion between APIs that are provided and APIs that are managed.

### [Recommendations for an API Economy Center of Excellence](#)

What roles are required to drive a successful API initiative, how should this fit in the current organization, and how do these roles relate to existing roles in the company?

### [Focus on the API Developer](#)

A productive developer is a happy developer. One of the most frequently discussed topics in the API economy is focusing on the needs of the Application developer – the consumer target for your APIs.

### [Agile API development](#)

Agile API Development Customer expectations and behavior are continuously changing. To deliver exceptional customer experience, a business must be nimble to adapt to these changing needs.

### [API Products – Who, What, Where, When, Why and How”](#)

An API Product is an API offering made available for consumer use that is offered to a target market to satisfy a customer’s needs.

### [Changing Culture – How Committed Are You?](#)

How do we change the culture in our organization to create an API culture?

### [Plan Ahead! Don't Build an API Superhighway into a Cul-de-sac](#)

Without proper planning, a business can start their API initiatives, build incredible excitement quickly, but find that the path they have taken leads them into a cul-de-sac (or dead end) that cannot handle the demand they have created.

### [IBM API Connect 2018.4.1.x Deployment WhitePaper](#)

A technical deep dive on the deployment options for IBM API Connect.

### [Principles for API Security - White Paper](#)

API security is of paramount importance in gaining the promised benefits without exposure to negative consequences.

### [Can you trust your APIs?](#)

As enterprises are continuously expanding their digital footprint, they must ensure the API behavior is intact, as it has a far-reaching effect on an application's execution and end-user experience.

### [Istio Service Mesh and APIConnect/DataPower Gateway integration](#)

What is Istio, and how can DataPower API Gateway integrate in an Istio Service Mesh.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorials

These tutorials guide you through the steps typically required to set up, configure and run an API Connect cloud. These tutorials cover topics such as creating the cloud topology, creating Provider organizations, developing and publishing API Products, and promoting those products to external developers.

## Introduction

---

These tutorials are arranged according to the desired goal, such as creating and publishing an API, or managing catalogs of products, or consuming published API products. These goals can be grouped into categories as follows.

Table 1. Tutorial Categories

Category	Description
Cloud Administration	Setting up and configuring the API Connect cloud, including the creation of Provider Organizations.
API Provider Administration	Setting up and managing catalogs, inviting organization members (such as developers), creating Developer Portals, managing resources.
API Product Development and Management	Developing and publishing APIs, products and plans, managing product life cycles.
API Product Promotion and Usage	Managing the Developer Portal presentation and user administration, as well as consuming published API products as an external developer.

---

## Time required

Each tutorial should take approximately 20 - 30 minutes to finish. If you explore other concepts that are related to this tutorial, it might take longer.

Use the links in the following sections to access the tutorials that fit your goals.

## Cloud Administration

---

Table 2. Cloud Administration Tasks. Setting up and configuring the API Connect cloud, including the creation of Provider Organizations.

Category	Tutorials
Initial Configuration	<ul style="list-style-type: none"><li><a href="#">Initial Cloud Configuration</a></li></ul>
Provider Organization	<ul style="list-style-type: none"><li><a href="#">Creating a Provider Organization</a></li></ul>

## API Provider Administration

---

Table 3. API Provider Tasks. Setting up and managing catalogs, inviting organization members (such as developers), creating Developer Portals, managing resources.

Category	Tutorials
Developer Portal Set Up	<ul style="list-style-type: none"><li><a href="#">Creating the Portal</a></li><li><a href="#">Register a Consumer Application</a></li></ul>

## API Product Development and Management

---

Table 4. API Development. Developing and publishing APIs, products and plans, managing product life cycles.

Note: All tutorials listed here use the API Manager interface. The user experience using the API Designer may differ.

Category	Tutorials
Getting Started	<ul style="list-style-type: none"><li><a href="#">Invoke a REST API</a></li><li><a href="#">Create a SOAP API from WSDL</a></li><li><a href="#">Import an API</a></li></ul>
Building APIs	<ul style="list-style-type: none"><li><a href="#">Expose SOAP as REST API</a></li><li><a href="#">Mapping JSON Content</a></li></ul>
Security	<ul style="list-style-type: none"><li><a href="#">Using OAuth Password Grant Tokens</a></li><li><a href="#">Using OpenID Connect security</a></li><li><a href="#">Generating a JSON Web Token (JWT)</a></li><li><a href="#">Validating a JSON Web Token (JWT)</a></li></ul>
Managing APIs	<ul style="list-style-type: none"><li><a href="#">Supercede an Existing API</a></li></ul>

## API Product Promotion and Usage through the Developer Portal

---

There are many tutorials available for the Developer Portal from getting started to advanced customization. For more information, see [Developer Portal tutorials](#).

## Prerequisites

---

- You must have a web browser available, whether you are working online or offline.
- When publishing Products in any of the tutorials you must have the permissions of an Organization Manager or Administrator. However, you can complete several of the tutorials with fewer permissions. For more information about user roles, see [Administering user access](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing and maintaining your IBM API Connect cloud

To ensure that your IBM® API Connect cloud functions, your cloud must have the necessary system requirements to support the installation. During the installation process, the components of IBM API Connect can be configured to satisfy your requirements.

Note: For a comprehensive technical guide to best practices, considerations, and deployment options for API Connect, see the [API Connect 2018.4.1.x Whitepaper](#).

- [Kubernetes](#)  
Use the instructions in this section to install, upgrade, and maintain API Connect on Kubernetes.
- [VMware](#)  
Use the instructions in this section to install, upgrade, and maintain API Connect on VMware.
- [IBM Cloud Private](#)  
Use the instructions in this section to install, upgrade, and maintain API Connect on IBM Cloud Private.
- [OpenShift](#)  
You can install API Connect on OpenShift.
- [IBM Cloud Pak for Integration](#)  
IBM API Connect provides the API management capability in IBM Cloud Pak for Integration.
- [Migrating a Version 5 deployment](#)  
The current migration path from v5 is to API Connect v10. You can get more information about migrating to v10 here: [https://www.ibm.com/support/knowledgecenter/SSMNEJ\\_v10/com.ibm.apic.install.doc/migrating.html](https://www.ibm.com/support/knowledgecenter/SSMNEJ_v10/com.ibm.apic.install.doc/migrating.html).
- [Using the API Connect operations command line interface](#)  
The IBM API Connect v2018 `apicops` command line interface is targeted at Operations teams. It contains commands to check the health of the system and some commands to fix specific problems that might be encountered.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Kubernetes

Use the instructions in this section to install, upgrade, and maintain API Connect on Kubernetes.

- [Installing and upgrading on Kubernetes](#)  
Use these instructions to install or upgrade a deployment of API Connect on Kubernetes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing and upgrading on Kubernetes

Use these instructions to install or upgrade a deployment of API Connect on Kubernetes.

- [IBM API Connect Version 2018 software product compatibility requirements](#)  
Ensure that you install the minimum API Connect operating system requirements. Use the IBM® Software Product Compatibility Reports site to generate a requirements report appropriate for your API Connect version and environment.
- [Estimating internal storage space for Analytics](#)  
If you plan to store data locally in your IBM API Connect Analytics deployment, estimate disk space requirements.
- [Working with certificates](#)  
Use the `certs` command included in the APICUP installer to set and manage certificates for each subsystem. Default certificates are automatically applied, but defaults can be overridden by user-supplied custom certificates.
- [Tips and tricks for using APICUP](#)  
The APICUP installer in Install Assist contains built-in time saving functions.
- [Deploying to a Kubernetes environment](#)  
Install Assist provides script-based installation using the APICUP tool into a Kubernetes runtime environment.
- [Upgrading API Connect in a Kubernetes environment](#)  
Upgrades are performed from the same project directory used for the initial installation.
- [Deleting the API Connect deployment in a Kubernetes runtime environment](#)  
To delete the Kubernetes deployment of API Connect, you delete the Helm charts, Custom Resource Definitions, the Persistent Volumes, and the namespace.
- [Maintaining a Kubernetes deployment](#)  
You can use utilities to complete maintenance tasks such as backup, restore, and certificate management on Kubernetes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## IBM API Connect Version 2018 software product compatibility requirements

Ensure that you install the minimum API Connect operating system requirements. Use the IBM® Software Product Compatibility Reports site to generate a requirements report appropriate for your API Connect version and environment.

To generate an API Connect requirements report, complete the following steps:

1. Open the [Detailed system requirements for a specific product](#) page on the IBM Software Product Compatibility Reports site.
2. Search for the IBM API Connect product.

3. In the Search results list, select IBM API Connect.
4. From the Version list, select the required version.
5. Use the Filters to refine the contents of the requirements report.
6. Click Submit to generate your requirements report.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Estimating internal storage space for Analytics

If you plan to store data locally in your IBM® API Connect Analytics deployment, estimate disk space requirements.

This information applies only to Analytics deployments where the ingestion-only option is set to false (the default setting). The formula and guidelines are based on known information about how the Analytics service stores data, and cannot be directly applied to any other storage system.

Use the following guidelines to calculate a rough estimate of the amount disk space you need for storing stateful data in the API Connect Analytics microservices.

### Data storage: calculate how much data you want to store

---

This section applies if you have chosen a topology that uses the analytics-storage-data or analytics-storage-basic microservices.

The amount of disk space needed for storing analytics data is determined by the following factors:

#### Number of copies of the analytics data

Each copy of the analytics data must live on a separate node. The number of nodes in your deployment determines the number of copies of analytics data that is stored in your deployment. With the **Production** deployment profile, analytics data is replicated to three nodes (two replicas and one primary copy of the data) for storage. In the **Nonproduction** profile, you only need to store one copy of the data.

Number of copies of data:

```
Copies of data = [1 | 3]
```

#### Number of days that data is retained

By default, analytics data is retained for 90 days, but you can [modify the retention setting](#) as needed. Make sure you know how long you want to store the data before attempting to calculate required disk space.

Number of days that data is retained:

```
Data retention = [90 days | preferred length]
```

#### Amount of each type of data stored

The required storage space for each type of logging is highly dependent on your APIs and usage. For each API, you can [configure logging](#) for the API activity, header, and payload. You can also [customize the data](#) to add, redact, or remove fields, which also impacts the amount of data that you store.

To estimate storage needs, calculate the average size of each type of log. When calculating your estimates, remember that the header logging size is the sum of activity logging size and the average size of your headers. The payload logging size is the sum of the header logging size and the average size of your payloads.

Typically the average size of an activity logged event is 600-1000 bytes depending on the uniqueness and complexity of your analytics data. This number is highly dependent on your APIs and your implementations. For a rough estimate, you can use an average of 800 bytes per activity logged event.

If you choose to add fields, calculate the average size of the new fields as well, and add that number to all types of log policies. If you choose to remove fields, you should not subtract this size from the log policies unless you are also removing headers and/or payloads.

Amount of each type of data that is stored:

```
Activity log bytes per call = [600-1000 bytes]
Header log bytes per call = Activity log bytes per call + Average size of headers
Payload log bytes per call = Header log bytes per call + Average size of payloads
```

#### Percentage of each type of data

Estimate the percentage of each type of log (activity, header, payload) for all API calls.

If you follow best practices of using only activity logging for production environments and using only payload logging for test environments, this number is easy to determine. If you use different log policies per API, and they depend on "success" or "error" factors, the percentage is more difficult to determine. Typically, if you do not use an all-or-nothing method for logging, error rates range from 3% to 25% with subsequent payload logging in test and production environments. However, this is entirely dependent on your use case and your APIs.

Percentage of each type of data that is stored:

```
% of Activity log = [0, 100 or other estimate]
% of Header log = [0, 100 or other estimate]
% of Payload log = [0, 100 or other estimate]
```

#### Estimated number of API calls per month

When planning your API Connect deployment, this number is helpful. When estimating analytics storage, this number is vital because it is directly correlated to how much storage you need for your deployment.

If you do not know the number of calls per month, but you do know the number of calls per second, use the following formula to convert it to calls per month:

```
Calls per month = Calls per second * 86400 seconds per day * 30 days per month
```

Number of API calls per month:

```
Calls per month = [any estimate]
```



## Calculating your disk space requirement

Estimate the disk space requirement for each storage node by completing following calculations.

### Formula

```
Bytes per call = (% of Activity Log * Activity Log bytes per call) + (% of Header Log * Header Log bytes per call) + (% of Payload Log * Payload Log bytes per call)
Calls per retention period = Calls per month * (Number of days retained / 30 days per month)
Storage for API calls = Calls per retention period * Bytes per call * Copies of data
Total storage = Storage for API calls + Overhead
Storage per node = Total storage / Nodes
```

### Details

#### 1. Bytes per call:

Estimate the number of bytes that are logged for a single copy of each API call. This can be calculated from the prerequisites of the percentage of each data type and the amount of each data type stored.

```
Bytes per call = (% of Activity Log * Activity Log bytes per call) + (% of Header Log * Header Log bytes per call) + (% of Payload Log * Payload Log bytes per call)
```

#### 2. Calls per retention period:

Calculate the anticipated number of API calls per retention period. This can be calculated from the prerequisites of the estimated calls per month and your desired data retention.

```
Calls per retention period = Calls per month * (Number of days retained / 30 days per month)
```

#### 3. Storage for API Calls:

Calculate the storage needed for all copies of your api calls for your retention period. This can be calculated from steps 1 and 2, as well as the prerequisite of your total copies of your data.

```
Storage for API calls = Calls per retention period * Bytes per call * Copies of data
```

#### 4. Total Storage:

Calculate the total storage you need by adding buffer space to the value from step 3. The Analytics service requires the some overhead space to complete its operations; for example, to contain system data and temporary debug header or payload logging. In addition, allowing extra space provides a buffer in case you underestimated the number of calls, or experience an unexpected increase.

There is no specific value for the buffer because it's based on your own situation. One approach is to use a value that brings the Storage for API calls result from step 3 up to the next round number. Make sure the rounding leaves you with a comfortable amount of additional space. For example, if the result from step 3 is 380 GB, then adding 20 GB to reach 400 GB is probably not sufficient and you should consider rounding to the a larger value such as 500 GB.

```
Total storage = Storage for API calls + Buffer
```

#### 5. Storage per node:

Calculate the total storage amount required per storage node. The number of storage nodes is dependent on your deployment profile. For the **development** profile, use 1. For the **production** profile, it defaults to 3. If you manually scaled the analytics-storage-data or analytics-storage-basic microservices to be greater than 3, use your actual values.

```
Storage per node = Total storage / Nodes
```

Remember: This result is only an estimate. You should monitor the use of space over time and adjust storage as needed.

### Example

Deployment information:

```
Copies of data = 3
Data retention = 90 days
Activity log bytes per call = 850 bytes
Header log bytes per call = 15k bytes
Payload log bytes per call = 30.5k bytes
% of Activity log = 100%
% of Header log = 0%
% of Payload log = 0%
Calls per month = 64 million
```

Formula:

```
Bytes per call = (% of Activity Log * Activity Log bytes per call) + (% of Header Log * Header Log bytes per call) + (% of Payload Log * Payload Log bytes per call)
Calls per retention period = Calls per month * (Number of days retained / 30 days per month)
Storage for API calls = Calls per retention period * Bytes per call * Copies of data
Total storage = Storage for API calls + Overhead
Storage per node = Total storage / Nodes
```

Details:

#### 1. Bytes per call = 850 bytes

```
(100% of Activity Log * 850 bytes) + (0% of Header Log * 15k bytes) + (0% of Payload Log * 30.5 bytes)
```

#### 2. Calls per retention period = 192 million calls

```
64 million calls per month * (90 days retained / 30 days per month)
```

#### 3. Storage for API calls = 489.6 GB

```
192 million calls per period * 850 bytes per call * 3 copies of data
```

#### 4. Total storage = 600 GB

#### 489.6GB storage + Buffer

Since rounding to 500 GB only provides 10.4 GB of extra space, it's good practice to round to 600 GB instead.

5. Storage per node = 200 GB

#### 600GB Total storage / 3 nodes

In this example, the estimated disk needed on each node for analytics-storage-data and analytics-storage-basic microservices is 200GB.

## Master storage: estimate disk space needs

This section applies if you are planning a topology with the analytics-storage-master microservice enabled.

For the analytics-storage-master microservice, estimating the amount of disk space you need is much easier. For a production environment, set this value to 10GB. For a development environment, you can optionally reduce the value to 5GB.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with certificates

Use the `certs` command included in the APICUP installer to set and manage certificates for each subsystem. Default certificates are automatically applied, but defaults can be overridden by user-supplied custom certificates.

### About this task

Note: Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

- **Certificate management: Read This First**  
Requirements and best practices for managing API Connect certificates.
- **Setting and managing certificates**  
Default certificates are automatically applied, but defaults can be overridden by custom certificates.
- **Reference for certificates, commands, and validations**  
This section contains the reference information for certificates, commands for working with certificates, and validations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Certificate management: Read This First

Requirements and best practices for managing API Connect certificates.

Important: Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged. For management purposes, API Connect certificates can be grouped by type (default, custom, and common) and by usage (public, public and user-facing, and internal). Understand each type and usage before you change any certificates.

- A default certificate is a private certificate that is uniquely generated by the installer for the current project directory, and will pass validation. Default certificates are automatically generated for each subsystem by the `apicup subsys install` command, unless the certificates were explicitly set by using the `apicup certs set` command. Note that the default certificates are self-signed, so they might not provide a level of trust suitable for external communication.
- For optimal trust levels, we recommend that you explicitly set all public and user-facing certificates by creating custom certificates.
- Some certificates are common across subsystems. Subsystems require the common certificates to allow them to register with the management subsystem. When installing any one subsystem, the common certificates are set for that subsystem and for all the other subsystems. If you use custom certificates for the common certificates, the custom certificates must be set prior to setting any other custom certificates.
- We do not recommend the explicit setting of internal certificates. The changing of internal certificates creates a risk of incompatibility with other internal certificates.

To review the usage (public, public and user-facing, or internal) of each certificate, see the following [Table 1](#) table.

Table 1. Certificate management best practice

Certificate usage	Certificate name	Best practice management
Public	<ul style="list-style-type: none"><li>• apic-gw-service-ingress</li></ul>	For optimal trust levels, set these explicitly. If certificates are not explicitly set by using the <code>apicup certs set</code> command, then default self-signed certificates are automatically generated by <code>apicup subsys install</code> . Typically you usually want to customize them, to ensure a level of trust suitable for external communication.

Certificate usage	Certificate name	Best practice management
Public and User-facing	<ul style="list-style-type: none"> <li>platform-api, api-manager-ui, cloud-admin-ui</li> <li>consumer-api</li> <li>portal-www-ingress</li> </ul>	For optimal trust levels, set these explicitly. Recommended to be explicitly set as custom certificates because they are presented to an end user through a browser or Command Line Interface (CLI).
Internal	<ul style="list-style-type: none"> <li>root-ca, ingress-ca</li> <li>mgmt-db-ca, mgmt-ca, service-server, service-client, db-server, db-client, service-plugin</li> <li>portal-admin-ingress, portal-client, portal-ca, portal-db-ca, service-server, service-client, apim-client</li> <li>analytics-client-ingress, analytics-ingestion-ingress, analytics-ingestion-client, analytics-client-client</li> <li>analytics-ca, service-server, service-client</li> <li>gw-ca, gateway-peering</li> </ul> <p>For VMware appliance deployments only: k8s-ca, appliance-client</p>	<p>Do not change. Accept the default certificates. It is possible to change these certificates (except ingress-ca) but is strongly discouraged because you risk creating incompatibilities that can block internal communications.</p> <p>Each intermediate cert (mgmt-db-ca, mgmt-ca, portal-ca, portal-db-ca, analytics-ca, gw-ca), is used to generate other internal certs. If, for example, you change an intermediate cert, the certs generated from it might not work with internal certs generated from other intermediate certs.</p> <p>Note that ingress-ca is auto-generated and cannot be set using the <code>apicup certs set</code> command.</p>

See also:

- [Setting and managing certificates](#)
- [Reference for certificates, commands, and validations.](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting and managing certificates

Default certificates are automatically applied, but defaults can be overridden by custom certificates.

- [Setting default certificates](#)  
Default certificates are automatically generated by APICUP when the subsystem is installed.
- [Setting custom certificates](#)  
Use the APICUP installer `certs` commands to set custom certificates.
- [Replacing custom certificates](#)  
Use the APICUP installer `certs` commands to replace existing certificates.
- [Setting common certificates](#)  
Common certificates are set for one subsystem, but are applied to all subsystems. Use the APICUP installer `certs` commands to set the common certificates.
- [Setting the encryption-secret for the management database](#)  
Use the APICUP installer `certs` commands to set the encryption-secret for the management database.
- [Clearing certificates](#)  
Certificates can be cleared in order to set new certificates.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting default certificates

Default certificates are automatically generated by APICUP when the subsystem is installed.

### About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

Default certificates are generated for each subsystem by the `apicup subsys install` command. If certificates are not explicitly set by using the `apicup certs`

set command, then default certificates are automatically generated by APICUP. The default certificates are self-signed, so they might not provide a level of trust suitable for external communication.

## Procedure

1. Enter the settings for the subsystem by using `apicup subsys set <SUBSYS>` and validate the subsystem settings by using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Install the subsystem by using the `apicup subsys install` command.
3. The default certificates are created for the subsystem. A default certificate is a private certificate that is uniquely generated by the installer for this project directory, they are self-signed and always pass validation.
4. List all certificates that are set for a subsystem by using the `apicup certs list` command.

```
apicup certs list -help
```

```
List all configured certificates
```

```
Usage:
```

```
apicup certs list SUBSYS [flags]
```

```
Flags:
```

```
-h, --help help for list
```

```
Global Flags:
```

```
--accept-license Accept the license for API Connect
--debug Enable debug logging
```

Following is example output from the `apicup certs list` command:

```
Common certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
analytics-client-client	CN: analytics-client-client SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
analytics-ingestion-client	CN: analytics-ingestion-client SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
ingress-ca	CN: ingress-ca SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
mgmt-db-ca	CN: mgmt-db-ca SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
portal-client	CN: portal-client SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
root-ca	CN: root-ca SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7 AuthorityKeyId:	

```
Subsystem mgmt_subsys certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
consumer-api	CN: consumer-api SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
db-client	CN: db-client SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88 AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
db-server	CN: db-server SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
encryption-secret	A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30	
mgmt-ca	CN: mgmt-ca SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
migration-client	CN: migration-client SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
platform-api	CN: platform-api SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:95:7D:3E:C5:43:F9:48 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
service-client	CN: service-client SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-plugin	CN: service-plugin SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-server	CN: service-server SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting custom certificates

Use the APICUP installer `certs` commands to set custom certificates.

---

### About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

The APICUP installer can be used to set certificates for each subsystem during installation. If certificates are not explicitly set by using the `apicup certs set` command, then default certificates are generated by APICUP. The default certificates are self-signed, so they might not provide a level of trust suitable for external communication.

Requirements for custom certificates:

- Extended Key Usage (EKU), either `serverAuth` or `clientAuth` depending upon the type of certificate. Certificates of type *Server* must have an Extended Key Usage with `serverAuth` purpose. Certificates of type *Client* must have an Extended Key Usage with `clientAuth` purpose.
- Subject Alternative Name (SAN) for the required hosts
- Any custom common certificates that are being used must be set prior to setting any custom certificates for a subsystem.

See [Certificate Reference](#) to view the list of common certificates and to determine whether an EKU is needed for a certificate and which type of EKU (`serverAuth` or `clientAuth`).

Precedence order for TLS certificates for Management endpoints:

The Management subsystem has four public endpoints: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, and `consumer-api`. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, `consumer-api`.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the `api-manager-ui` TLS certificate will be effective for all endpoints, as it is first in the precedence order.
- If the `api-manager-ui`, `cloud-admin-ui`, and `platform-api` endpoints are the same, and `consumer-api` is a different endpoint, then the `api-manager-ui` TLS certificate will be effective for the `api-manager-ui/cloud-admin-ui/platform-api` endpoints, while the `consumer-api` TLS certificate will be effective for the `consumer-api` endpoint

Note: Once API Connect has been installed (meaning that the `apicup subsys install` `SUBSYS` command has been executed) with a given set of certificates, only the certificates for the public ingress endpoints (`portal-www`, `api-manager-ui`, `cloud-admin-ui`, `platform-api`, `consumer-api`) can be modified. The TLS certificates involved in mutual authentication (`portal-admin-ingress`, `portal-client`, `analytics-ingestion-ingress`, `analytics-ingestion-client`, `analytics-client-ingress`, and `analytics-client-client`) cannot be modified after the `install` command has been executed.

---

## Procedure

1. Set up and validate the subsystem. Enter the settings for the subsystem using `apicup subsys set <SUBSYS>` and validate the subsystem settings using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Generate the custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (`platform-api-example`) with an EKU `serverAuth` and SAN using `openssl`:

```
openssl x509 -req -days 360 -in platform-api-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:fqdn.myserver.com\nextendedKeyUsage=serverAuth") -extensions SAN
```

where

- `DNS:fqdn.myserver.com` is the fully qualified domain name of the endpoint the certificate applies to. This matches the endpoints entered in the APICUP installer. See [Installing the Management subsystem into a Kubernetes environment](#)
- `platform-api-example.csr` is the file name for the certificate signing request

Following is an example for how to generate a certificate (`portal-client`) with an EKU `clientAuth` and SAN using `openssl`:

```
openssl x509 -req -days 360 -in portal-client-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical, CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

3. Once the certificate has been created, set the certificate by entering the following command:  
`apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE]`

You can find definitions for commands at the following location: [Command reference](#)

If the certificate is signed by an intermediate CA, the `CA_File` argument must point to a file that concatenates the intermediate CA, followed by the root CA, in that order.

If the certificate is signed by an internal or custom CA, include the full chain in the end certificate. If you omit the full chain, then a user who uses openssl to access the endpoint will see the following error: `error:num=20:unable to get local issuer certificate`

The following example shows the full chain for an end certificate (signed by a custom CA):

```
-----BEGIN CERTIFICATE -----
Cert contents for end-cert
-----END CERTIFICATE -----
-----BEGIN CERTIFICATE -----
Cert contents for Intermediate-CA
-----END CERTIFICATE -----
-----BEGIN CERTIFICATE -----
Cert contents for Root CA
-----END CERTIFICATE -----
```

If the certificate was generated with an EKU serverAuth, it must be assigned to a server certificate. If the certificate was generated with an EKU clientAuth, it must be assigned to a client certificate.

- Repeat for additional custom certificates.
- After setting the custom certificates, you can optionally generate the remaining default certificates prior to installation by entering the `apicup certs generate` command. The `generate` command generates any certificates that have not been set, so it will create default certificates for all remaining certificates. It will not overwrite any custom certificates you have set. You can review the certificates prior to installation.
- List all certificates with `apicup certs list SUBSYS`. The results will include the generated default certificates and the custom certificates that you set.

Following is example output from the `apicup certs`

`list` command:

**Common certificates**

=====

Name	Summary	Validation errors
analytics-client-client	CN: analytics-client-client SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
analytics-ingestion-client	CN: analytics-ingestion-client SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
ingress-ca	CN: ingress-ca SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
mgmt-db-ca	CN: mgmt-db-ca SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
portal-client	CN: portal-client SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
root-ca	CN: root-ca SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7 AuthorityKeyId:	

**Subsystem mgmt\_subsys certificates**

=====

Name	Summary	Validation errors
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
consumer-api	CN: consumer-api SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
db-client	CN: db-client SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88 AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
db-server	CN: db-server SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
encryption-secret	A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30	
mgmt-ca	CN: mgmt-ca SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
migration-client	CN: migration-client SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
platform-api	CN: platform-api SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:95:7D:3E:C5:43:F9:48 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
service-client	CN: service-client SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-plugin	CN: service-plugin SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-server	CN: service-server	

```
SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7
AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
```

7. Install the subsystem with the certificates using `apicup subsys install`  
**SUBSYS.** Any missing certificates will be generated. The installation will not proceed if there are any validation issues with the certificates. See [Validation reference](#).
8. Repeat for other subsystems.
9. If necessary, you can replace custom certificates after installation is complete. See [Replacing custom certificates](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Replacing custom certificates

Use the APICUP installer `certs` commands to replace existing certificates.

---

### About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

The APICUP installer can be used to update certificates for each subsystem after installation.

Requirements for custom certificates:

- Extended Key Usage (EKU), either `serverAuth` or `clientAuth` depending upon the type of certificate. Certificates of type *Server* must have an Extended Key Usage with `serverAuth` purpose. Certificates of type *Client* must have an Extended Key Usage with `clientAuth` purpose.
- Subject Alternative Name (SAN) for the required hosts
- Any custom common certificates that are being used must be set prior to setting any custom certificates for a subsystem.

See [Certificate Reference](#) to view the list of common certificates and to determine whether an EKU is needed for a certificate and which type of EKU (`serverAuth` or `clientAuth`).

Precedence order for TLS certificates for Management endpoints:

The Management subsystem has four public endpoints: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, and *consumer-api*. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, *consumer-api*.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the *api-manager-ui* TLS certificate will be effective for all endpoints, as it is first in the precedence order.
- If the *api-manager-ui*, *cloud-admin-ui*, and *platform-api* endpoints are the same, and *consumer-api* is a different endpoint, then the *api-manager-ui* TLS certificate will be effective for the *api-manager-ui/cloud-admin-ui/platform-api* endpoints, while the *consumer-api* TLS certificate will be effective for the *consumer-api* endpoint

---

## Procedure

1. Generate the custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (*platform-api-example*) with an EKU `serverAuth` and SAN using `openssl`:

```
openssl x509 -req -days 360 -in platform-api-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:fqdn.myserver.com\nextendedKeyUsage=serverAuth")) -extensions SAN
```

where

- `DNS:fqdn.myserver.com` is the fully qualified domain name of the endpoint the certificate applies to. This matches the endpoints entered in the APICUP installer. See [Installing the Management subsystem into a Kubernetes environment](#)
- `platform-api-example.csr` is the file name for the certificate signing request

Following is an example for how to generate a certificate (*portal-client*) with an EKU `clientAuth` and SAN using `openssl`:

```
openssl x509 -req -days 360 -in portal-client-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical,CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

2. Once the certificate has been created, set the certificate by entering the following command:  
`apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE]`

You can find definitions for commands at the following location: [Command reference](#)



If the certificate is signed by an intermediate CA, the `CA_File` argument must point to a file that concatenates the intermediate CA, followed by the root CA, in that order.

If the certificate was generated with an EKU serverAuth, it must be assigned to a server certificate. If the certificate was generated with an EKU clientAuth, it must be assigned to a client certificate.

3. Install the subsystem with the new certificate using `apicup subsys install SUBSYS`. Any missing certificates will be generated. The installation will not proceed if there are any validation issues with the certificates. See [Validation reference](#).
4. Repeat for other subsystems requiring new certificates.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting common certificates

Common certificates are set for one subsystem, but are applied to all subsystems. Use the APICUP installer `certs` commands to set the common certificates.

---

### About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

The common certificates are identical across subsystems. Subsystems require the common certificates to allow them to register with the management subsystem. When installing any one subsystem, the common certificates will be set for that subsystem and for all the other subsystems. If you are using custom certificates for the common certificates, they must be set prior to setting any custom certificates. See [Certificates reference](#) for a description of the common certificates.

Common certificates cannot be changed between subsystem installs. For example, you cannot set a common certificate for the management subsystem, install the management subsystem, then change a common certificate for the analytics subsystem, then install the analytics subsystem. This scenario will result in a failed installation because the common certificates are not identical.

---

### Procedure

Follow these steps to set custom common certificates. If using default certificates, the common certificates will be set for you. See [Setting custom certificates](#)

1. Set up and validate all subsystems. Enter the settings for the subsystem using `apicup subsys set <SUBSYS>` and validate the subsystem settings using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Generate a custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (platform-api) with an EKU serverAuth and SAN using openssl:

```
openssl x509 -req -days 360 -in platform-api.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:ac-msntest.myserver.com,DNS:ac2-msntest.myserver.com\nextendedKeyUsage=serverAuth")) -extensions SAN
```

Following is an example for how to generate a certificate (portal-client) with an EKU clientAuth and SAN using openssl:

```
openssl x509 -req -days 360 -in portal-client.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical,CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

3. Once the certificate has been created and you have a `.pem` file, set the custom common certificates in one of your subsystems. After setting the custom certificates for one subsystem, they will take effect for all subsystems. The command is targeted at a specific subsystem, but the common certificates are copied to all subsystems regardless of which subsystem they are originally set in. Following is an example for setting the `portal-client` certificate:  
`apicup certs set mgmt portal-client myCertFile.pem myKeyFile.key myCAFile.crt`
4. Set the remaining certificates for each subsystem, default or custom. See [Setting default certificates](#) and [Setting custom certificates](#).
5. List and validate the certificates for each subsystem. See [Validation reference](#)
6. Install each subsystem using `apicup subsys install SUBSYS`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Setting the encryption-secret for the management database

Use the APICUP installer `certs` commands to set the encryption-secret for the management database.

### About this task

---

Note: Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

The encryption-secret is a secure random bytes password used for field level encryption in the management database. You can generate 128 random bytes using the following command in openssl:

```
openssl rand -out /path/to/secret/encryption-secret.bin 128
```

Important: The encryption-secret can only be set once and only during initial installation. See [Installing the Management subsystem into a Kubernetes environment](#).

### Procedure

---

1. Enter the `apicup certs set SUBSYS CERT_NAME [KEY_FILE]` command and complete the following values:
  - SUBSYS - The subsystem for the `encryption-secret` is the name of your management subsystem, because it is used for field-level encryption for the management database.
  - CERT\_NAME - The certificate name is `encryption-secret`.
  - KEY\_FILE - Enter the file name for a secure random bytes string that is 128 bytes in length, for example `encryption-secret.bin`.
2. Set the remaining certificates if using custom certificates and install the management subsystem.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Clearing certificates

Certificates can be cleared in order to set new certificates.

### About this task

---

Note: Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Upgrading API Connect in a Kubernetes environment](#)

Existing certificates must be cleared in order to set new certificates. When using default certificates, new certificates are created only for those certificates that are not set. Some of the use cases for clearing certificates are: expired certificates and configuration changes. For example, of endpoints are changed, the existing certificates must be cleared and new certificates created.

### Procedure

---

1. Enter the `apicup certs set SUBSYS CERT_NAME --clear` command and complete the following values:
  - SUBSYS - The name of the subsystem
  - CERT\_NAME - The name of the certificate that you want to clear.
2. The certificate will be cleared and can be reset, either as a default or custom certificate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Reference for certificates, commands, and validations

This section contains the reference information for certificates, commands for working with certificates, and validations.

- [Certificate reference](#)  
The Certificate reference provides a description of all the certificates required in API Connect.
- [Command reference](#)  
The APICUP installer includes the `certs` commands to set and manage certificates.
- [Validation reference](#)  
Certificates are validated using several parameters.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Certificate reference

The Certificate reference provides a description of all the certificates required in API Connect.

### Before you begin

APICUP sets default certificates for each subsystem during installation. The default certificates are self-signed so may not provide a level of trust suitable for external communication. Custom certificates can be set and managed following the steps described in [Setting custom certificates](#).

The Certificates reference topic lists all certificates that are set by the `apicup certs` commands. The default certificates can be used for the majority of these certificates. The certificates that are marked as *public and user-facing* are recommended to be explicitly set as custom certificates because they are presented to an end user through a browser or Command Line Interface (CLI).

The certificates that are described as *TLS certificate used by ingress* are also considered *public* in the sense that they interact with a client that sits outside of an API Connect cluster.

The remaining certificates are considered *internal* because they interact with internal components.

Certificates that are listed as *auto-generated* cannot be set using the `apicup certs set` command. For example, the common certificate `ingress-ca` is auto-generated and used as an intermediate CA for all default ingress certificates. If you set an ingress certificate as a custom certificate, you will need to configure an intermediate CA if desired.

Important:

When setting custom certificates, additional steps must be taken to provide an Extended Key Usage (EKU) serverAuth and ClientAuth and a Subject Alternative Name (SAN) for the required hosts. These certificates are generated automatically by the `apicup certs` command when using the default certificates.

- For custom certificates of type *server*, an additional extended key usage client authentication (EKU serverAuth) certificate is required.
- For custom certificates of type *client*, an additional extended key usage server authentication (EKU clientAuth) certificate is required.

### Procedure

1. **Common certificates** - The following certificates are common to all subsystems in a deployment. Subsystems require the common certificates to allow them to register with the management subsystem. The common certificates are identical across subsystems. When installing any one subsystem, the common certificates will be set for that subsystem and for all the other subsystems. Custom common certificates must be set prior to setting any custom subsystem certificates.

Common certificates cannot be changed between subsystem installs. For example, you cannot set a common certificate for the management subsystem, install the management subsystem, then change a common certificate for the analytics subsystem, then install the analytics subsystem. This scenario will result in a failed installation because the common certificates are not identical.

Table 1. Common certificates

Certificate name (value used in <code>apicup certs</code> )	Type	Usage	Requirements	Description
root-ca	CA	internal		CA certificate which forms the root of the certificate chain
ingress-ca	CA	internal	signed by: root-ca	Auto-generated intermediate certificate used to generate certificates for subsystem ingress endpoints if not provided by user. The ingress-ca intermediate certificate cannot be set explicitly, it is always only generated. TLS certificates for the ingresses are not required to use ingress-ca as an intermediate certificate, but if a given ingress TLS certificate is left to be auto-generated then it will be signed by this ingress-ca.
mgmt-db-ca	CA	internal	signed by: root-ca	Intermediate CA certificate used to sign certificates used by Cassandra.
portal-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <code>portal-admin-ingress</code> .	Client certificate used by management subsystem to authenticate with the Portal admin endpoint. Requires EKU clientAuth.
analytics-client-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <code>analytics-client-ingress</code> .	Client certificate used by management subsystem to authenticate with analytics client endpoint. Requires EKU clientAuth.
analytics-ingestion-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <code>analytics-ingestion-ingress</code> .	Client certificate used by gateway subsystem to authenticate with analytics ingestion endpoint. Requires EKU clientAuth.

2. **Management certificates**

These certificates apply to a single Management subsystem.

The Management subsystem has four endpoints: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, and `consumer-api`. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, `consumer-api`.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the `api-manager-ui` TLS certificate will be effective for all endpoints, as it is first in the precedence order.

- If the api-manager-ui, cloud-admin-ui, and platform-api endpoints are the same, and consumer-api is a different endpoint, then the api-manager-ui TLS certificate will be effective for the api-manager-ui/cloud-admin-ui/platform-api endpoints, while the consumer-api TLS certificate will be effective for the consumer-api endpoint

The encryption-secret certificate is unique in that it is a secure random number. It is used to provide field level encryption in management (Cassandra) database. To set the encryption secret for the management database, use the following command: `apicup certs set SUBSYS CERT_NAME [KEY_FILE]` For example: `apicup certs set mgmt1 encryption-secret /path/to/keyfile`. See [Setting the encryption-secret for the management database](#)

Table 2. Management certificates

Certificate name	Type	Usage	Requirements	Description
encryption-secret	secure random bytes		length: 128 bytes	Encryption secret used to do field level encryption in management (Cassandra) database
platform-api	Server	public and user-facing	The host names for which the certificate is valid must include the platform-api endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <code>store.acme.com</code> , the certificate can be one that is valid for host names matching <code>*.acme.com</code> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
consumer-api	Server	public and user-facing	The host names for which the certificate is valid must include the consumer-api endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <code>store.acme.com</code> , the certificate can be one that is valid for host names matching <code>*.acme.com</code> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
api-manager-ui	Server	public and user-facing	The host names for which the certificate is valid must include the api-manager-ui endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <code>store.acme.com</code> , the certificate can be one that is valid for host names matching <code>*.acme.com</code> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
cloud-admin-ui	Server	public and user-facing	The host names for which the certificate is valid must include the cloud-admin-ui endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <code>store.acme.com</code> , the certificate can be one that is valid for host names matching <code>*.acme.com</code> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
mgmt-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign management subsystem certificates
service-server	Server	internal	signed by: mgmt-ca Required hosts: <ul style="list-style-type: none"> <li>• <code>*.&lt;namespace&gt;</code></li> <li>• <code>*.&lt;namespace&gt;.svc</code></li> <li>• <code>*.&lt;namespace&gt;.svc.cluster.local</code></li> </ul>	Server certificates used by management services. Requires EKU serverAuth.
service-client	Client	internal	signed by: mgmt-ca	Client certificate used by management services. Requires EKU clientAuth.
db-server	Server	internal	signed by: mgmt-db-ca Required hosts: <ul style="list-style-type: none"> <li>• <code>*.&lt;namespace&gt;</code></li> <li>• <code>*.&lt;namespace&gt;.svc</code></li> <li>• <code>*.&lt;namespace&gt;.svc.cluster.local</code></li> </ul>	Server certificate used by management (Cassandra) database to communicate with other nodes. Requires EKU serverAuth.
db-client	Client	internal	signed by: mgmt-db-ca	Client certificate used by management services. Requires EKU clientAuth.
service-plugin	Client	internal	signed by: mgmt-ca	Client certificate used by plugin services to talk to management subsystem. Requires EKU clientAuth.
migration-client	Client		signed by: mgmt-ca	This certificate is deprecated and no longer used, and can be ignored.

### 3. Portal certificates

These certificates apply to a single portal subsystem.

Table 3. Portal certificates

Certificate name	Type	Usage	Requirements	Description
portal-admin-ingress	Server	internal	host must match admin endpoint	TLS certificate used by ingress. The <code>portal-client</code> common certificate must be set prior to setting the <code>portal-admin-ingress</code> certificate. Requires EKU serverAuth. The <code>portal-admin-ingress</code> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <code>ingress-ca</code> .
portal-www-ingress	Server	public and user-facing	host must match www endpoint	TLS certificate used by ingress. Requires EKU serverAuth.
portal-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign portal subsystem certificates
portal-db-ca	CA	internal	signed by: portal-ca	Intermediate CA certificate used to sign certificates used by portal DB

Certificate name	Type	Usage	Requirements	Description
service-server	Server	internal	signed by: portal-ca  Required hosts: <ul style="list-style-type: none"> <li>*. &lt;namespace&gt;</li> <li>*. &lt;namespace&gt;.svc</li> <li>*. &lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificates used by portal services. Requires ECU serverAuth.
service-client	Client	internal	signed by: portal-ca	Client certificate used by portal services. Requires ECU clientAuth.
apim-client	Client	internal	signed by: portal-ca	Client certificate used by portal services to talk to management subsystem. Requires ECU clientAuth.

#### 4. Analytics certificates

These certificates apply to a single analytics subsystem.

Table 4. Analytics certificates

Certificate name	Type	Usage	Requirements	Description
analytics-client-ingress	Server	internal	Required hosts: <ul style="list-style-type: none"> <li>client ingress endpoint</li> <li>*. &lt;namespace&gt;</li> <li>*. &lt;namespace&gt;.svc</li> <li>*. &lt;namespace&gt;.svc.cluster.local</li> </ul>	TLS certificate used by ingress. The <b>analytics-client-client</b> common certificate must be set prior to setting the <b>analytics-client-ingress</b> certificate. Requires ECU serverAuth.  The <b>analytics-client-ingress</b> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <b>ingress-ca</b> .
analytics-ingestion-ingress	Server	internal	Required hosts: <ul style="list-style-type: none"> <li>ingestion ingress endpoint</li> <li>*. &lt;namespace&gt;</li> <li>*. &lt;namespace&gt;.svc</li> <li>*. &lt;namespace&gt;.svc.cluster.local</li> </ul>	TLS certificate used by ingress. The <b>analytics-ingestion-client</b> common certificate must be set prior to setting the <b>analytics-ingestion-ingress</b> certificate. Requires ECU serverAuth.  The <b>analytics-ingestion-ingress</b> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <b>ingress-ca</b> .
analytics-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign analytics subsystem certificates
service-server	Server	internal	signed by: analytics-ca  Required hosts: <ul style="list-style-type: none"> <li>*. &lt;namespace&gt;</li> <li>*. &lt;namespace&gt;.svc</li> <li>*. &lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificates used by analytics services. Requires ECU serverAuth.
service-client	Client	internal	signed by: analytics-ca	Client certificate used by analytics services; requires ECU clientAuth.

#### 5. Gateway certificates

These certificates apply to a single gateway subsystem.

Table 5. Gateway certificates

Certificate name	Type	Usage	Requirements	Description
api-gateway-ingress	Server	public and user-facing	host must match api-gateway endpoint	TLS certificate used by ingress. Requires EKU serverAuth. (deprecated, see <b>Note</b> .)
apic-gw-service-ingress	Server	public	host must match apic-gw-service endpoint	TLS certificate used by ingress. Requires EKU serverAuth.
gw-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign gateway subsystem certificates
gateway-peering	Server	internal	signed by: gw-ca	Server certificates used by gateway services. Requires EKU serverAuth.

Note: The **api-gateway-ingress** certificate is a legacy certificate which has been deprecated. It is no longer used to terminate TLS at the api-gateway endpoint. You configure a TLS profile for the termination of the api-gateway endpoint using the Cloud Manager API Invocation Endpoint and SNI settings when registering the gateway. The profiles used during configuration can be updated as needed. See [Registering a gateway service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Command reference

The APICUP installer includes the `certs` commands to set and manage certificates.

### About this task

The APICUP installer can be used to set certificates for each subsystem during installation. If certificates are not explicitly set using the `apicup certs set` command, then default certificates are generated by APICUP. We recommend that certificates be set at installation time only (or carried over from an upgrade). The default certificates are self-signed, so they may not be optimal for external communication.

For a description of the certificates that can be set, see [Certificate reference](#). We recommend that all public and user-facing certificates be explicitly set, including `portal-www-ingress` and `api-gateway-ingress`, and the four management endpoints (`platform-api`, `consumer-api`, `api-manager-ui`, and `cloud-admin-ui`). Following is the help reference for the `apicup certs set` command:

```
apicup certs set --help
Set or clear certificates and keys

Usage:
  apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE] [KEY_FILE] [flags]

Flags:
  --clear    Clear out a certificate or key entry
  -h, --help help for set

Global Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging
```

## Procedure

To set and clear certificates, complete the following steps:

1. Enter the `apicup certs set` command and complete the following values:

Table 1. apicup certs set

Command	Values	Result
<code>apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE] [KEY_FILE] [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem to which the certificate applies</li> <li>CERT_NAME - name of the certificate; see <a href="#">Certificate reference</a> for a list of certificates that can be set for each subsystem.</li> <li>CERT_FILE - Path to the certificate file in PEM format.</li> <li>KEY_FILE - Path to the private key file in PEM format.</li> <li>CA_FILE - Path to the Certificate Authority (CA) file. The contents of the file may be the concatenation of an intermediate CA and the root CA (in that order). <b>Note:</b> When setting the <code>root-ca</code> certificate, omit the <code>CA_FILE</code> parameter.</li> </ul>	Applies the certificate when the subsystem is installed.
<code>apicup certs set SUBSYS CERT_NAME [KEY_FILE] [flags]</code>	<code>KEY_FILE</code> - The file containing the encryption-secret for field level encryption in the management database. Applies only to the management subsystem. The certificate name is <code>encryption-secret</code> . The type is secure random bytes with a length of 128 bytes. For example, <code>apicup certs set mgmt1 encryption-secret /path/to/encryption-secret.bin</code> . <b>Note:</b> Do not specify any of the <code>[CERT_FILE KEY_FILE CA_FILE]</code> parameters when setting the encryption-secret.	Applies the <code>encryption-secret</code> when the management subsystem is installed.
<code>flags</code> <ul style="list-style-type: none"> <li><code>--clear</code></li> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--clear</code> - Clears the specified certificate. For example, <code>apicup certs set mgmt1 encryption-secret --clear</code></li> <li><code>--help</code> - Displays help for the command.</li> </ul>	The specified certificate will be cleared. When making configuration changes such as changing endpoints, the corresponding certificate must be cleared so that a new certificate can be set.

2. The `apicup certs get` command retrieves a specific certificate for the specified subsystem.

```
apicup certs get --help
Get a certificate
```

```
Usage:
apicup certs get SUBSYS CERT_NAME [flags]
```

```
Flags:
-h, --help          help for get
-o, --output string output to file or - (stdout) (default "-")
-t, --type string   type of object to return: cert, key, ca (default "cert")
```

```
Global Flags:
--accept-license  Accept the license for API Connect
--debug          Enable debug logging
```

Table 2. apicup certs get

Command	Values	Result
<code>apicup certs get SUBSYS CERT_NAME [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem to which the certificate applies</li> <li>CERT_NAME - name of the certificate to retrieve; see <a href="#">Certificate reference</a> for a list of certificates</li> </ul>	Returns the specified certificate for the specified subsystem.
<b>flags</b> <ul style="list-style-type: none"> <li><code>--output string</code></li> <li><code>--type string</code></li> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--output string</code> - Specify a file for the retrieved values, or specify "-" to send to stdout. Default is "-" to send to stdout. For example, <code>apicup certs get mgmt1 --output myCertsFile</code></li> <li><code>--type string</code> - Returns only the specified type. If not specified, the type is cert. For example, <code>apicup certs get mgmt1 --type ca</code></li> <li><code>--help</code> - Displays help for the command.</li> </ul>	<ul style="list-style-type: none"> <li>For <code>--output</code>: The specified certificate will be retrieved and sent to stdout or saved to the specified file</li> <li>For <code>--type</code>: Certificates will be retrieved that match the type specified.</li> </ul>

3. List all certificates that have been set for a subsystem using the `apicup certs list` command. You can list the certificates at any time to summarize the certificates that have been set.

```
apicup certs list -help
```

```
List all configured certificates
```

```
Usage:
apicup certs list SUBSYS [flags]
```

```
Flags:
-h, --help help for list
```

```
Global Flags:
--accept-license  Accept the license for API Connect
--debug          Enable debug logging
```

Table 3. apicup certs list

Command	Values	Result
<code>apicup certs list SUBSYS [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem for which you want to list certificates</li> </ul>	Returns a list of certificates that are configured for the subsystem.
<b>flags</b> <ul style="list-style-type: none"> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--help</code> - Displays help for the command.</li> </ul>	Help text is displayed.

Following is example output from the `apicup certs list` command:

```
Common certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
analytics-client-client	CN: analytics-client-client SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
analytics-ingestion-client	CN: analytics-ingestion-client SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
ingress-ca	CN: ingress-ca SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
mgmt-db-ca	CN: mgmt-db-ca SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
portal-client	CN: portal-client SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
root-ca	CN: root-ca SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7 AuthorityKeyId:	

```
Subsystem mgmt_subsys certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C	

```

consumer-api      AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
                  CN: consumer-api
                  SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46
db-client         AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
                  CN: db-client
                  SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88
db-server        AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66
                  CN: db-server
                  SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC
                  AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66
encryption-secret mgmt-ca  AuthorityKeyId: A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30
                  CN: mgmt-ca
                  SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
                  AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7
migration-client CN: migration-client
                  SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66
                  AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
platform-api     CN: platform-api
                  SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:95:7D:3E:C5:43:F9:48
                  AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
service-client   CN: service-client
                  SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76
                  AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
service-plugin   CN: service-plugin
                  SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73
                  AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
service-server   CN: service-server
                  SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7
                  AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75

```

4. The `apicup certs generate` command generates and sets default certificates. The `generate` command only generates and sets a certificate if it is not already set; it only sets the missing default certificates that have not been explicitly set using the `set` command. Execute the `generate` command before running the `apicup subsys install <SUBSYS>` command to confirm the certificates are correct before installing. It allows you to validate all certificates before performing the installation. The `generate` command is used as a tool to assist you when entering a combination of default and custom certificates. If you need to set specific certificates you can set them upfront (using `set`) and then generate the missing ones with default certificates. Or you can generate all certificates upfront and then override specific certificates to set custom certificates. Using `generate` helps to avoid validation errors during the installation procedure. Note that you must configure the subsystems and pass the `--validate` option before generating the default certificates.

```

apicup certs generate -help
Generate all unset certificates

```

```

Usage:
  apicup certs generate SUBSYS [flags]

```

```

Flags:
  -h, --help      help for generate
Global Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging

```

Table 4. apicup certs generate

Command	Values	Result
<code>apicup certs generate SUBSYS [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem for which you want to generate certificates</li> </ul>	Generates certificates that have not been set for the subsystem. Generates self-signed certificates.
<code>flags</code> <ul style="list-style-type: none"> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--help</code> - Displays help for the command.</li> </ul>	Help text is displayed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Validation reference

Certificates are validated using several parameters.

The validations described in the following table are applied for all certificates, but are most helpful for custom certificates. For default certificates, the certificate validations will always pass, as the required elements are generated by APICUP. However, with custom certificates, some of the required elements may be missing or incorrect.

Validation	Messages	Error	See also/Action
Verify the certificate is set properly.	<code>certificate &lt;cert&gt; not set</code>	The certificate is not set.	
	<code>unable to load cert &lt;cert&gt;</code>	The certificate is set but cannot be read.	
Verify certificate key usage (Extended Key Usage).	<code>unable to verify cert &lt;cert&gt;: missing key usage &lt;n&gt;</code>	The certificate is missing the required key usage.	See <a href="#">Certificates Reference</a> to see more information, including the type, for all certificates. See <a href="#">Setting custom certificates</a> for tips on how to generate the EKUs for custom certificates.

Validation	Messages	Error	See also/Action
Verify the certificate signing CA. If available, the CA file is loaded. Then the certificate is verified against the provided CA file, including enforcement of Extended Key Usage.	unable to parse CA to verify cert <cert>	The CA file could not be parsed and loaded.	
	unable to verify cert <cert>	The certificate failed verification against the provided CA file.	One possible reason for receiving this error is that the correct EKU is missing. For a custom certificate, see <a href="#">Setting custom certificates</a> for information on generating EKUs.
Verify certificate hosts. The certificate must be valid for the hosts listed for the certificate in the Requirements column in the Certificates Reference.	unable to verify cert <cert>: missing <host>	The certificate is not valid for the required host.	See <a href="#">Certificate reference</a> for the required hosts.
Verify that a certificate that is being used as a CA is actually a CA.	unable to verify cert <cert>: certificate is not a CA	The certificate is not a valid CA.	
Verify client certificate match. The portal-client, analytics-client-client, and analytics-ingestion-client certificates are verified against the CA of, respectively, portal-admin-ingress, analytics-client-ingress, and analytics-ingestion-ingress.	a CA certificate must be provided for this certificate	The CA certificate is missing for one of the portal-admin-ingress, analytics-client-ingress, and analytics-ingestion-ingress.	The common certificates portal-client, analytics-client-client, and analytics-ingestion-client must be set prior to setting any custom certificates.
	client cert cannot be verified against provided CA certificate	The verification failed.	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tips and tricks for using APICUP

The APICUP installer in Install Assist contains built-in time saving functions.

### Introduction

This section describes tips and techniques for working with the APICUP installation commands. The APICUP installer creates charts and secrets that are then managed by Helm.

- [Running Tiller with APICUP](#)
- [Entering multiple settings per command](#)
- [Entering multiple values](#)
- [Viewing the settings for a subsystem](#)
- [Validating the settings for a subsystem](#)
- [Output an installation plan](#)
- [Getting help for commands](#)
- [Getting help on a specific command](#)

### Running Tiller with APICUP

To run Tiller in a namespace:

```
export TILLER_NAMESPACE=apic
```

### Entering multiple settings per command

To configure multiple settings per command, enter a space between each setting and enter an equals sign (=) to configure the setting.

In the following examples, be sure to replace [spc] with an actual space.

You can set multiple database parameters on one line:

```
apicup subsys set mgmt cassandra-max-memory-gb=9 [spc] cassandra-cluster-size=3 [spc] cassandra-volume-size-gb=16 [spc] cassandra-backup-protocol=sftp
```

You can set all endpoints on one line:

```
apicup subsys set mgmt platform-api=my.platform.com [spc] api-manager-ui=my.apim.com [spc] cloud-admin-ui=my.cloud.com [spc] consumer-api=my.consumer.com
```

### Entering multiple values

Separate multiple values for a key/value pair with commas.



## Viewing the settings for a subsystem

To view the current values that are set for a subsystem, enter the following command: `apicup subsys get <subsys_name>`. For example: `apicup subsys get mgmt` outputs the current values for the subsystem named `mgmt` and provides a description of each value. The output from the `get` command is organized into Kubernetes settings and Subsystem settings. Following is an example:

Figure 1. Output for the `get` command

```
Kubernetes settings
=====

Name          Value          Description
----          -
extra-values-file (Optional) Path to additional configuration yml file
ingress-type   ingress        Ingress type (use `route` for OpenShift)
mode          standard      mode
namespace     default       K8S namespace to install into
registry      Docker image registry to use
registry-secret Docker registry credentials secret
storage-class  -            K8S storage class for persistent storage

Subsystem settings
=====

Name          Value          Description
----          -
cassandra-backup-auth-pass (Optional) Server password for DB backups
cassandra-backup-auth-user (Optional) Server username for DB backups
cassandra-backup-host (Optional) FQDN for DB backups server
cassandra-backup-path (Optional) path for DB backups server
cassandra-backup-port (Optional) Server port for DB backups
cassandra-backup-protocol (Optional) Protocol for DB backups (sftp/ftp)
cassandra-backup-schedule (Optional) Cron schedule for DB backups
cassandra-cluster-size Size of DB cluster (min 3 for HA)
cassandra-max-memory-gb Memory limit for DB
cassandra-volume-size-gb Size of DB storage volume (not resizable)
create-crd     true          Create DB cluster CRDS (Required cluster admin privilege)
external-cassandra-host (Optional) Hostname of externally hosted DB

Endpoints
=====

Name          Value          Description
----          -
api-manager-ui FQDN of API manager UI endpoint
cloud-admin-ui FQDN of Cloud admin endpoint
consumer-api  FQDN of consumer API endpoint
platform-api  FQDN of platform API endpoint

Error: Subsystem validation failure. Run with --validate to see details
```

## Validating the settings for a subsystem

The values set for a subsystem can be validated for syntax by entering the `--validate` option for the `get` command: `apicup subsys get <subsys_name> --validate`. For example: `apicup subsys get mgmt --validate` validates the current values for the subsystem named `mgmt`. In the following example of the output for the `--validate` option, the check mark indicates a valid setting. The `x` indicates an invalid setting with an error message provided.

Figure 2. Output for the `--validate` option

```
Subsystem settings
=====

Name          Value          Description
----          -
cassandra-backup-auth-pass ✓
cassandra-backup-auth-user ✓
cassandra-backup-host ✓
cassandra-backup-path /backups ✓
cassandra-backup-port 22 ✓
cassandra-backup-protocol sftp ✓
cassandra-backup-schedule 0 0 * * * ✓
cassandra-cluster-size 1 ✓
cassandra-max-memory-gb 9 ✓
cassandra-volume-size-gb 50 ✓
create-crd true ✓
external-cassandra-host ✓

Endpoints
=====

Name          Value          Description
----          -
api-manager-ui x api-manager-ui is not a valid hostname
cloud-admin-ui x cloud-admin-ui is not a valid hostname
```

```
consumer-api          ✗ consumer-api is not a valid hostname
platform-api         ✗ platform-api is not a valid hostname
```

## Output an installation plan

Using APICUP, you can generate an installation plan and confirm it is correct prior to running the installation. You can then install the subsystem from the plan.

To output an installation plan, enter the following command:

```
apicup subsys install SUBSYS --out=install-plan
```

where <install-plan> is the name of the directory where the installation plan will be stored.

In the example, a directory named install-plan is created in the project directory. The myProject/install-plan directory contains the configuration parameters for the subsystem.

To install the subsystem from the install-plan, enter the following command from the project directory:

```
apicup subsys install SUBSYS --plan-dir=<full-path-to-plan-directory>
```

where <full-path-to-plan-directory> is the full qualified path to the plan directory.

Following are general rules for installing from the installation plan:

- Never edit the files in the output directory directly. Instead, if changes are needed, update the parameters using APICUP and generate a new plan.
- Always run APICUP commands from the original project directory created during the initial product installation. The project directory contains the apiconnect-up.yml file.
- You can generate the plan in any location, but the install command must be run from the project directory. Enter the full path to the plan as the argument to `--plan-dir` to perform an installation.

The plan must be current with the project and the certificates. If the plan is older than the last modification date of the project or certificates, you will receive an error message such as:

```
the project was modified since the plan was generated, regenerate plan or skip this check
with a --no-verify flag in the command
```

or

```
the certs were changed since the plan was generated, regenerate plan or skip this check with
a --no-verify flag in the command
```

## Getting help for commands

You can get help for all commands by entering: `apicup --help`. Following is an example of the output:

Figure 3. Help for Install Assist apicup commands

```
APIConnect Install Assist
Usage:
  apicup [command]

Available Commands:
  certs          Subsystem certificates
  completion    Generates bash or zsh completion scripts
  help          Help about any command
  hosts         Commands to configure subsystem hosts
  iface         Commands to configure hosts interfaces
  init          Create a new APIConnect UP project
  registry-upload Retag and upload images to custom registry
  server        Starts the APIConnect cluster operator
  subsys        Subsystem commands
  version       Get the APIConnect UP version

Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging
  -h, --help       help for apicup

Use "apicup [command] --help" for more information about a command.
```

## Getting help on a specific command

For help on a specific command, enter `--help` after the command. For example, `apicup subsys get mgmt --help` prints out the usage and flags for the `get` command. For example:

Figure 4. Help for get command

```
Get subsystem properties

Usage:
  apicup subsys get SUBSYS [flags]

Flags:
  --endpoints  List endpoints (default true)
  -h, --help  help for get
  --platform  List platform settings (default true)
  --subsystem  List subsystem settings (default true)
```

```
--validate    Validate settings

Global Flags:
--accept-license  Accept the license for API Connect
--debug          Enable debug logging
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deploying to a Kubernetes environment

Install Assist provides script-based installation using the APICUP tool into a Kubernetes runtime environment.

### About this task

---

The Install Assist tool contains the APICUP installation program to provide an automated installation process for API Connect in a Kubernetes environment. This section contains the software requirements, download and installation procedure, and the recommended minimum allocations for system resources in the Kubernetes cluster.

- [Requirements for a Kubernetes deployment](#)  
Ensure that your environment meets these requirements for deploying API Connect on Kubernetes.
- [Load balancer configuration in a Kubernetes deployment](#)  
When deploying API Connect for High Availability, it is recommended that you configure a cluster with at least three nodes and a load balancer. A sample configuration is provided for placing a load balancer in front of your API Connect Kubernetes deployment.
- [Configuring logging for a Kubernetes deployment](#)  
Logs must be collected for both running and terminated containers and processes. Logging collection is required for IBM Support to assist with troubleshooting. For container-based deployments (Kubernetes), specific log commands and mechanisms vary depending on a customer's environment.
- [Installing API Connect into a Kubernetes environment](#)  
You can use Kubernetes and Docker containers to deploy and run API Connect. The Install Assist utility program automates the installation process into a Kubernetes runtime environment using an installation program called APICUP.
- [Installing the Gateway subsystem into a Kubernetes environment](#)  
Use the Install Assist utility program to install the Gateway subsystem into your Kubernetes runtime environment.
- [Deploying pods to specific worker nodes in a multi-node cluster](#)  
Use node labels to deploy Management, Analytics, and Portal pods to specific worker nodes in a multi-node cluster.
- [Manually creating a CustomResourceDefinition in a Kubernetes environment](#)  
Describes the steps for manually creating the CustomResourceDefinitions which are required for the management subsystem. If the `create-crd` command is set to false during installation, the CRDs must be manually created by the Kubernetes administrator.
- [Creating an extra values file in a Kubernetes environment](#)  
Describes the steps for creating an extra values file for configuration parameters that are not set by Install Assist, such as ingress annotations and resource limits. An example `.yaml` file is provided with entries for each subsystem.
- [Configuring user-defined policies on the API Gateway in a Kubernetes deployment](#)  
For a Kubernetes deployment that uses the DataPower API Gateway, user-defined Policies are externally configured. To distribute the user-defined policies on the DataPower API Gateway, you create a Kubernetes ConfigMap that is installed using the extra values file. The ConfigMap ensures the policies are available to the Management server.
- [Installing the toolkit](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Requirements for a Kubernetes deployment

Ensure that your environment meets these requirements for deploying API Connect on Kubernetes.

- [Requirements for deploying IBM API Connect into a Kubernetes environment](#)  
Describes the system requirements for deploying into a Kubernetes runtime environment.
- [Deployment overview for endpoints and certificates](#)  
Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.
- [Firewall requirements on Kubernetes](#)  
Diagram for port configuration, and list of active ports, for an IBM® API Connect deployment on Kubernetes.
- [Kubernetes ingress controller prerequisites](#)  
Describes the prerequisite settings for the ingress controller for a Kubernetes runtime environment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Requirements for deploying IBM API Connect into a Kubernetes environment

Describes the system requirements for deploying into a Kubernetes runtime environment.

## Before you begin

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated. These instructions assume you have a working Kubernetes environment and understand how to manage Kubernetes.

Kubernetes is a platform for automated deployment, scaling, and operation of application containers across clusters of hosts, providing container-centric infrastructure. For more information, see <https://kubernetes.io>.

### Pre-installation Requirements

#### Important:

Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

The original project directory created with APICUP during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. Note that the endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored or upgraded system. A good practice is to back up the original project directory to a location from where it can always be retrieved.

- Refer to the IBM® Software Product Compatibility Reports site for detailed requirements for operating systems, supporting software, and other prerequisites. Ensure that your Helm installation is up-to-date and compatible with your Kubernetes version. See [Detailed system requirements for a specific product](#).
- The Management server and Gateway firmware versions must match, for example, API Connect 2018.4.1.3 and DataPower 2018.4.1.3.
- For API Connect v2018.4.1.15 and later, Helm v3 is required. See <https://helm.sh/>.
- Namespaces must be configured in Kubernetes before starting the installation process. We recommend that you do not use the default namespace.
- The timezone for API Connect pods is set to UTC. Do not change the timezone for the pods.
- Block storage is required. Select a block storage format of your choosing in order for API Connect components to be supported. GlusterFS is not recommended as a storage option due to severe performance degradation. For AWS, the gp2 and io1 types of Elastic Block Storage (EBS) are supported. The Developer Portal and the Analytics component additionally support the use of local volume storage and do not require block storage.
- Manually create the required CustomResourceDefinitions if you are not going to use the APICUP commands in Install Assist to create them. For more information, see [Manually creating a CustomResourceDefinition in a Kubernetes environment](#) and [Installing the Management subsystem into a Kubernetes environment](#).
- When deploying with a DataPower appliance, configure a Gateway Service Management Endpoint and API invocation Endpoint in DataPower. See [Configuring API Connect Gateway Service](#).
- Configure a remote server for logging. Follow these instructions: [Configuring remote logging for a Kubernetes deployment](#). Pods are terminated during an upgrade, and the logs will be lost if not stored remotely.
- Configure the log rotation by setting the **max-size** and **max-files** values to a setting appropriate for your deployment. In a Kubernetes deployment, insufficient values for max-size and max-files can cause logs to grow too large and eventually force pods to restart. For information, visit the Docker documentation and search for "JSON File logging driver".
- DNS must be configured with domain names that correspond to the endpoints configured at installation time for each subsystem.

Note: Host names and DNS entries may not be changed on a cluster after the initial installation.

The endpoints are also subsequently entered in the Cloud Manager UI to configure the services for your cloud. We recommend that you keep a record of the DNS entries and endpoint names, as you will need to map the DNS entries and use the same endpoints when restoring a backup of the management database. Note that endpoints are FQDNs, entered in all lowercase. See [Defining your topology](#). The following endpoints require DNS entries:

Important: For API Connect on native Kubernetes and OpenShift, do not use coreDNS 1.8.1 through 1.8.3. These releases have a defect that can prevent pods for Developer Portal from starting. See <https://coredns.io/>.

- Four domain names are recommended for the endpoints for the manager subsystem (although these may be mapped to one domain name, if desired), as follows:
  - Cloud Manager URL: `<cm>.<hostname>.<domainname>`
  - API Manager URL: `<apim>.<hostname>.<domainname>`
  - Platform REST API Endpoint for admin and provider APIs: `<api>.<hostname>.<domainname>`
  - Platform REST API Endpoint for consumer APIs: `<consumer>.<hostname>.<domainname>`

#### Note:

Kubernetes ingress limits the character set for DNS names to not support the underscore character "\_". This means you cannot specify underscores in domain names that are used as endpoints. For example, `foo_abc.bar.com` and `foo.bar_abc.com` are not supported for `<xxx>.<hostname>.<domainname>`, and will cause an error. For example:

```
Invalid value: "foo_abc.bar.com": a DNS-1123 subdomain must consist of lowercase alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character (e.g. 'example.com', regex used for validation is '[a-z0-9]([-a-z0-9]*[a-z0-9])?(\.[a-z0-9]([-a-z0-9]*[a-z0-9])?)*')
```

- The Gateway service requires two domain names to correspond to the following endpoints (both of these endpoints are entered in the Cloud Manager when defining a Gateway service). See [Registering a gateway service](#).
  - The API invocation endpoint that is defined by `api-gateway`, for example, `<gw>.<hostname>.<domainname>`. This is the API Endpoint Base defined in Cloud Manager.
  - The management endpoint that is defined by `apic-gw-service`, for example, `<gwd>.<hostname>.<domainname>`. This is the Management Endpoint defined in Cloud Manager.
- The Analytics service requires two domain names to correspond to the following endpoints:
  - The `analytics-ingestion` endpoint, for example, `<ai>.<hostname>.<domainname>`.
  - The management endpoint that is defined by `analytics-client`, for example, `<ac>.<hostname>.<domainname>`. This is the endpoint that is entered when defining an analytics service in Cloud Manager (see [Registering an analytics service](#)).
- The Portal service requires two domain names to correspond to the following endpoints (both of these endpoints are entered in the Cloud Manager when defining a Portal service). See [Registering a portal service](#):
  - The `portal-admin` endpoint, for example, `<padmin>.<hostname>.<domainname>`. This is the Management Endpoint defined in Cloud Manager.
  - The `portal-www` endpoint, for example, `<portal>.<hostname>.<domainname>`. This is the Portal Website URL defined in Cloud Manager.

- IBM API Connect requires certain repositories, and APICUP automatically creates the repositories in most cases. In some instances, for example, if installing into AWS ECR, you will need to manually create the following repositories for each subsystem:

**MANAGEMENT REPOSITORIES**

```
apim
client-downloads-server
ui
juhu
lur
analytics-proxy
ldap
busybox
cassandra
cassandra-operator
cassandra-health-check
migration
k8s-init
```

**PORTAL REPOSITORIES**

```
portal-db
portal-dbproxy
portal-admin
portal-web
openresty
portal-exec-job
```

**ANALYTICS REPOSITORIES**

```
analytics-cronjobs
analytics-client
analytics-ingestion
analytics-mq-kafka
analytics-mq-zookeeper
analytics-storage
analytics-operator
k8s-init
```

**GATEWAY REPOSITORIES**

```
datapower-api-gateway
k8s-datapower-monitor
```

- Decide the *mode* to use for the installation.

Table 1. Deployment modes

Mode	Description
<b>dev</b>	Development mode ( <b>dev</b> ) deploys a subsystem with the scale of one. Development mode is recommended for development, testing, and demonstration purposes.  <code>apicup subsys set mgmt mode=dev</code>  Do not use <b>dev</b> mode for production environments. Development mode does not provide high availability.
<b>standard</b>	Standard mode ( <b>standard</b> ) deploys in high availability mode for a production environment.  <code>apicup subsys set mgmt mode=standard</code>

Usage:

- Mode is set for each subsystem type: Management, Analytics, Developer Portal, and Gateway.
- If you do not specify the mode, a default mode is applied, as follows:
  - **Version 2018.4.1.4 and later:** **dev** mode
  - **Version 2018.4.1.3 and earlier:** **standard** mode
- Use of **standard** mode is supported only on installations with three or more nodes. Installations with less than three nodes must use **dev** mode. If you install in **standard** mode with only a single node, some pods can remain in a pending state. To avoid having pods remain pending, either install in **dev** mode or add additional nodes.
- To ensure the API Connect services have time to start, we recommend increasing the *proxy-read-timeout* and *proxy-send-timeout* values in the *config.map* used to configure the *kubernetes/ingress-nginx* ingress controller. The following settings should be increased:
  - `proxy-read-timeout: "240"`
  - `proxy-send-timeout: "240"`

240 seconds (4 minutes) is a recommended minimum; actual value will vary depending upon your environment. If there is a load balancer in front of the worker node(s), then the load balancer configuration may also need to have extended timeouts.

- Ensure that your Kubernetes deployment addresses known security vulnerabilities with SSH to prevent the use of weak SSH Message Authentication Code (MAC) and Key exchange algorithms (KexAlgorithms) with TCP over port 22. Use of weak algorithms can allow an attacker to recover the plain text message from the encrypted text. If you have not yet addressed the security vulnerabilities with weak MACs and KexAlgorithms, update your configuration manually by adding the following lines to */etc/ssh/sshd\_config*:

```
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com
```

System and Software Requirements

The installation instructions have been tested with the requirements described in the Software Product Compatibility Reports. See [Detailed system requirements for a specific product](#)

Note: API Connect v2018 cannot be deployed on NFS.

Important: There are known GlusterFS issues that affect the following functionality areas:

- Elasticsearch, which is used in the API Connect analytics service.
- The Developer Portal service.

If you use GlusterFS, you might encounter severe performance degradation and, in some cases, loss of data.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

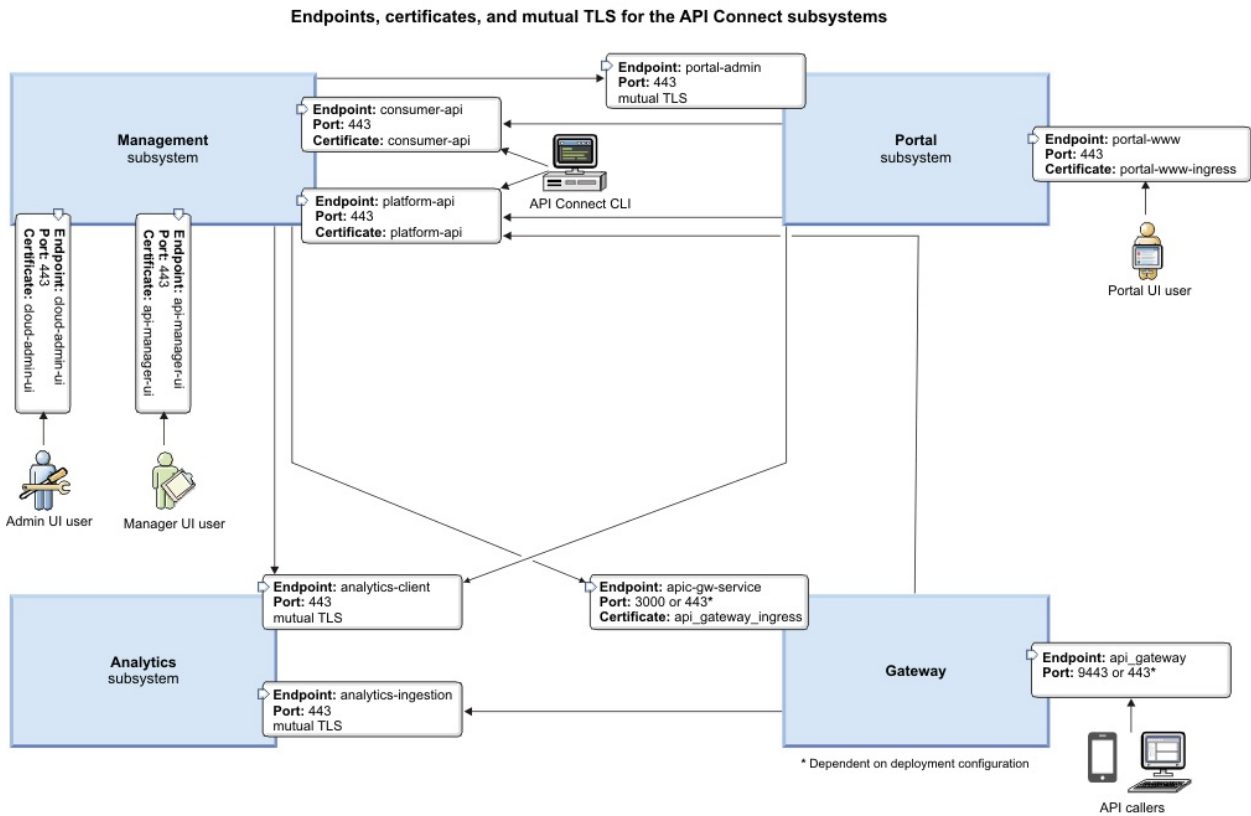
## Deployment overview for endpoints and certificates

Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.

### Introduction

When deploying API Connect, you will create one or more endpoints for the subsystems and then configure certificates or mutual TLS for most endpoints. [Figure 1](#) shows the endpoints for each subsystem by name, the name of the certificate that secures the endpoint, and whether mutual TLS is required. It also shows the ports consumed by the endpoints, which are standard for HTTP and HTTPS.

Figure 1. Deployment Overview diagram



### Configuring endpoints

The endpoints are configured by the Install Assist program using the APICUP installer. They are set for each subsystem. Endpoints are also entered when configuring the Topology for the Gateway, Portal, and Analytics subsystems in Cloud Manager.

For instructions on configuring endpoints and installing into a Kubernetes environment, see [Installing API Connect into a Kubernetes environment](#).

Instructions for installing into an IBM Cloud Private environment are here: [Deploying to an IBM Cloud Private environment](#).

Subsystem	Endpoints	Description	Certificates
Management	cloud-admin-ui	Configured using APICUP installer. Endpoint on the management server for communication with the Cloud Manager user interface.	cloud-admin-ui
	api-manager-ui	Configured using APICUP installer. API Manager URL endpoint on the management server for communication with the API Manager user interface.	api-manager-ui
	consumer-api	Configured using APICUP installer. Platform REST API endpoint for running consumer APIs on the management server.	consumer-api
	platform-api	Configured using APICUP installer. Platform REST API endpoint for running admin and provider APIs on the management server.	platform-api
Portal	portal-admin	Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS.	mutual TLS
	portal-www	Configured using APICUP installer. Portal Web site URL entered in Cloud Manager. Used publicly to access Portal.	portal-www-ingress

Subsystem	Endpoints	Description	Certificates
Analytics	analytics-client	Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS.	mutual TLS
	analytics-ingestion	Configured using APICUP installer. The analytics-ingestion endpoint is used by the Gateway service to push data to the Analytics service. Requires a TLS profile configured with mutual TLS.	mutual TLS
Gateway	apic-gw-service	Configured using APICUP installer. This is the endpoint the gateway uses for network communication. Enter this endpoint as the Management Endpoint entered in Cloud Manager.	apic-gw-service-ingress
	api-gateway	Configured using APICUP installer. This is the endpoint the gateway uses for API traffic. Enter this endpoint as the API Invocation Endpoint in Cloud Manager.	api-gateway-ingress

## Configuring certificates

The certificates are configured by the Install Assist program using the APICUP installer. The certificates for the endpoints are usually configured as custom certificates as described in [Setting custom certificates](#).

## Configuring mutual TLS

Mutual TLS is configured for TLS profiles in Cloud Manager. See [Creating a TLS Server Profile](#).

## Configuring a proxy

If a Developer Portal is deployed externally to the management server zone, it does not have access to the consumer and product APIs. You need to configure a proxy to enable communication. For more information, see [Configuring a proxy](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Firewall requirements on Kubernetes

Diagram for port configuration, and list of active ports, for an IBM® API Connect deployment on Kubernetes.

## Required Ports between zones

The following network diagram example helps to explain which ports must be configured in an API Connect network. Specific ports must be configured to enable the communication between the various zones, both public and private, in a network.

The ports specified in the diagram are default ports. Check your deployment to understand which communication, if any, is configured to use non-default ports.



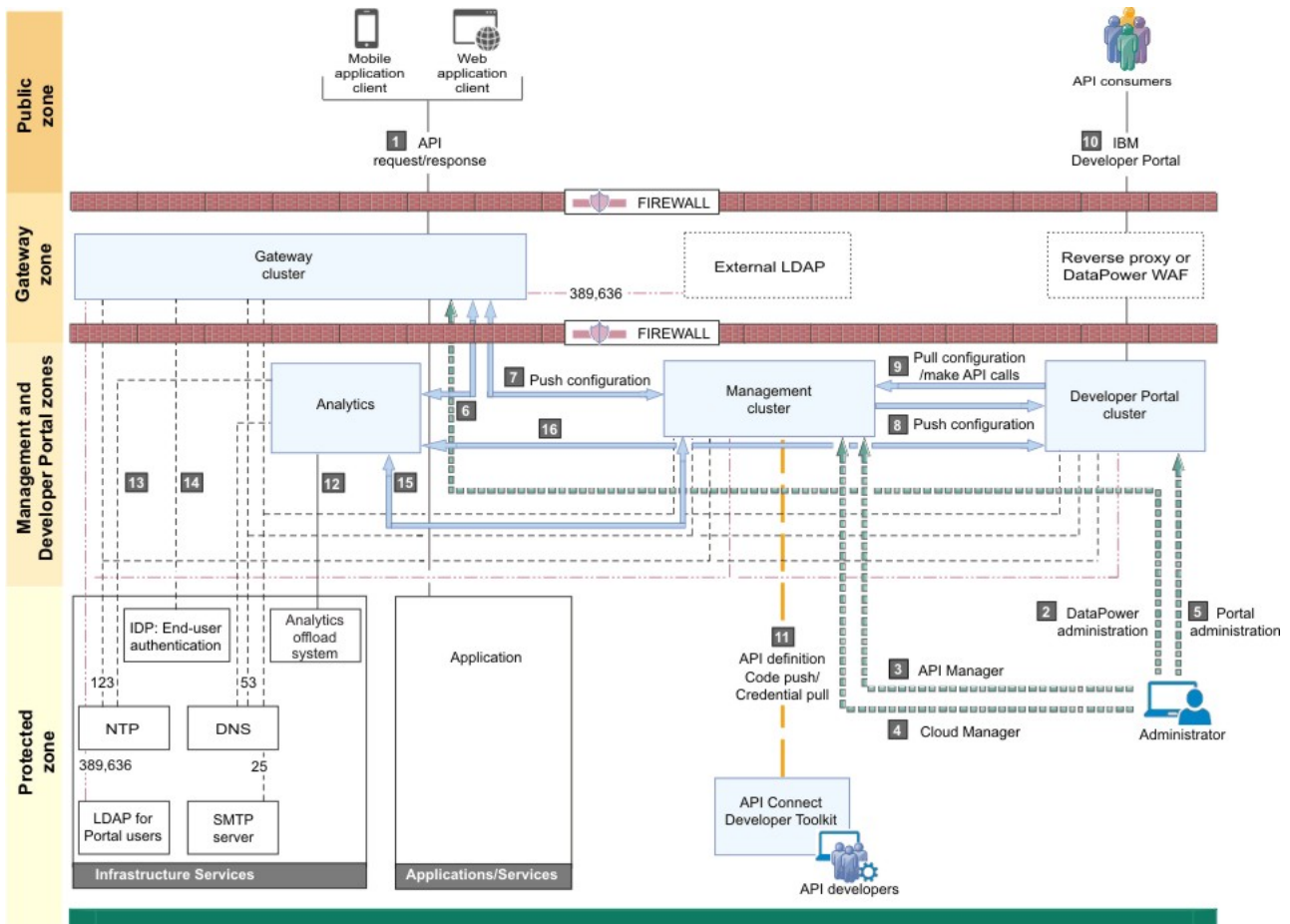


Table 1. Key for the network diagram example.  
The following table lists the port numbers with a usage description.

	Usage description	Default port number
1	API request/response – Users invoking the provided APIs.	443 HTTPS from Public zone to Gateway zone
2	DataPower® administration – Internal operators who are managing the Gateway servers.	22 SSH, 9090 HTTPS from Protected zone to Gateway zone
3	API Manager – Internal business users who are defining and monitoring APIs.	443 HTTPS from Protected zone to Management zone
4	Cloud Manager – Internal operators who are administering the Cloud.	22 SSH, 443 HTTPS from Protected zone to Management zone
5	Developer Portal administration – Internal operators who are managing the Portal servers.	22 SSH, 443 HTTPS from Protected zone to Management zone
6	Gateway servers post traffic to Analytics service.	443 HTTPS from Gateway servers to Analytics service
7	Push configuration – Management servers communicate bi-directionally with Gateway servers.	3000 or 443 (dependent on configuration) HTTPS Management servers to and from Gateway servers for webhook delivery
8	Push configuration/webhooks – Management servers push configuration and webhooks to the Developer Portal.	443 HTTPS Management servers to Developer Portal servers for webhook delivery
9	Pull configuration/make API calls – Developer Portal servers pull configuration and call REST APIs.	443 HTTPS from Developer Portal servers to Management servers within Management zone



	Usage description	Default port number
10	Developer Portal – External developers who are accessing the Developer Portal.	443 HTTPS from Public zone to Developer Portal management zone. The reverse proxy/DataPower WAF for incoming web traffic to the Developer Portal cluster must be a transparent proxy - no modification of the portal URL, port, host name or path is allowed. For more information, see <a href="#">Configuring a proxy</a> .
11	Push API definition to Management server. Pick up credential for microservice code push.	443 HTTPS from Protected zone to Management zone
12	Analytics offload	Port will depend on type of plugin and protocol used for the offload. Some possible protocols are: HTTP, HTTPS, TCP, UDP, KAFKA
13	Analytics accesses NTP	Standard NTP
14	Analytics access DNS	Standard DNS
15	Management service queries Analytics service	443 HTTPS within Management zone
16	The Portal service invokes an API (GET) on the Analytics service to retrieve data.	443 HTTPS within Management zone

## Firewall port requirements on Kubernetes

The following tables lists ports that must be open in both cluster and non-clustered deployments.

Table 2. Firewall port requirements common to all subsystems

Subsystem	Ports and description
Ports that must be open on all API Connect subsystems	<p>The following ports must be open on the Management Server, Analytics, and Developer Portal subsystems, whether in a cluster or not.</p> <ul style="list-style-type: none"> <li>• 22 (inbound and outbound) SSH</li> <li>• 53 (outbound) DNS</li> <li>• 123 (outbound) NTP</li> <li>• 443 (inbound and outbound) Used by all subsystems for communication with other subsystems</li> <li>• 44135 (inbound and outbound)</li> </ul>

Each subsystem uses ports in addition to the ports in [Table 2](#). See the following table.

Table 3. Additional firewall port requirements for each subsystem

Subsystem	Ports and description
Management Service	<p>Management Service uses the ports in <a href="#">Table 2</a> plus:</p> <ul style="list-style-type: none"> <li>• 22 remote backup server port (configurable) If backups are configured to use another port, ensure that the port is open. See <a href="#">Backing up the management database in a Kubernetes environment</a>.</li> <li>• 161 (inbound) SNMP</li> <li>• LDAP server port (if using LDAP user registry), typically 389 (outbound)</li> <li>• 3007 (outbound) LDAP</li> </ul>
Developer Portal	<p>The Developer Portal uses the ports listed in <a href="#">Table 2</a>, plus:</p> <ul style="list-style-type: none"> <li>• 22 - A remote backup server port (configurable) If backups are configured to use another port, ensure that the port is open. See <a href="#">Backing up and restoring the Developer Portal in a Kubernetes environment</a>.</li> </ul>
Analytics	<p>The Analytics subsystem uses the ports listed in <a href="#">Table 2</a>.</p> <p>Analytics port usage considerations:</p> <ul style="list-style-type: none"> <li>• 161 for SNMP is required for both non-clustered and clustered deployments</li> <li>• In a non-clustered deployment, no additional ports are required.</li> <li>• Analytics supports an optional configuration for offload of data. This configuration might require additional outbound ports to be open.</li> </ul>
Gateway Server	<p>The Gateway Server uses these ports in both non-clustered and clustered deployments:</p> <ul style="list-style-type: none"> <li>• 161 (inbound and outbound) SNMP</li> <li>• 162 (outbound) SNMP traps</li> <li>• 3000 (inbound) Gateway Service local port (configurable)</li> <li>• 5550 (inbound) XML management port (configurable)</li> <li>• 5554 (inbound) REST management port (if enabled; configurable)</li> <li>• 9022 (inbound) Gateway SSH (if enabled; configurable)</li> <li>• 9090 (inbound) Web GUI console (if enabled; configurable)</li> <li>• 9443 (inbound) Gateway local port (configurable)</li> </ul>

## Communications inside the Gateway cluster

There are a number of important points to note regarding the communications within the Gateway cluster.

- We advise that you use the same port for all Gateway servers within a cluster.
- Gateway servers communicate with each other to synchronize invocation counts.
- All Gateway servers in a Gateway cluster must be able to reach all of the other Gateway servers in the same Gateway cluster.

- Gateway servers in a Gateway cluster do not directly communicate with Gateway servers in a different Gateway cluster.
- All Gateway servers must be able to reach the management subsystem platform API endpoint, which was configured during the installation of your API Connect environment.

## Ethernet interface usage

To separate network traffic, you can use two or more Ethernet interfaces on the DataPower appliance on which a Gateway server is installed. For example, you can use one interface for internal IBM API Connect communications, and another for processing incoming API calls.

## Related concepts

- [Installing and maintaining your IBM API Connect cloud](#)

## Related reference

- [IBM API Connect Version 2018 software product compatibility requirements](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Kubernetes ingress controller prerequisites

Describes the prerequisite settings for the ingress controller for a Kubernetes runtime environment.

## Before you begin

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated.

These instructions assume you have a working Kubernetes environment and understand how to manage Kubernetes. Kubernetes is a platform for automated deployment, scaling, and operation of application containers across clusters of hosts, providing container-centric infrastructure. For more information, see <https://kubernetes.io>.

Kubernetes/ingress-nginx ingress controller `ingress-config.yml` settings

A Kubernetes deployment for IBM® API Connect requires the `kubernetes/ingress-nginx` ingress controller implementation (see <https://github.com/kubernetes/ingress-nginx>) with SSL passthrough enabled.

Important: When deploying to an OpenShift environment, an ingress controller is not used. Setting the `ingress-type` parameter to `route` using `apicup subsys set SUBSYS ingress-type route` command completes the configuration for the ingress on OpenShift. See [Settings for OpenShift](#). API Connect v2018 currently only supports Helm2, but for the purpose of installing the ingress controller we recommend using Helm3 just for that part of the installation. Follow these steps:

1. Create a file `ingress-config.yaml` where the following values are required:

```
controller:
  admissionWebhooks:
    enabled: false
  config:
    ssl-protocols: TLSv1.2
  extraArgs:
    annotations-prefix: ingress.kubernetes.io
    enable-ssl-passthrough: true
```

You may use the following sample `ingress-config.yml` file to configure the ingress controller:

```
controller:
  admissionWebhooks:
    enabled: false
  config:
    hsts-max-age: "31536000"
    keepalive: "32"
    log-format: '{ "@timestamp": "$time_iso8601", "@version": "1", "clientip": "$remote_addr",
      "tag": "ingress", "remote user": "$remote_user", "bytes": $bytes_sent, "duration":
      $request_time, "status": $status, "request": "$request_uri", "urlpath": "$uri",
      "urlquery": "$args", "method": "$request_method", "referer": "$http_referer",
      "useragent": "$http_user_agent", "software": "nginx", "version": "$nginx_version",
      "host": "$host", "upstream": "$upstream_addr", "upstream-status": "$upstream_status"
    }'
    main-snippets: load_module "modules/nginx_stream_module.so"
    proxy-body-size: "0"
    proxy-buffering: "off"
    server-name-hash-bucket-size: "128"
    server-name-hash-max-size: "1024"
    server-tokens: "False"
    ssl-ciphers: HIGH:!aNULL!MD5
    ssl-prefer-server-ciphers: "True"
    ssl-protocols: TLSv1.2
    use-http2: "true"
    worker-connections: "10240"
    worker-cpu-affinity: auto
    worker-processes: "1"
    worker-rlimit-nofile: "65536"
    worker-shutdown-timeout: 5m
```

```
daemonset:
  useHostPort: false
extraArgs:
  annotations-prefix: ingress.kubernetes.io
  enable-ssl-passthrough: true
hostNetwork: true
kind: DaemonSet
name: controller
rbac:
  create: "true"
```

2. Run the commands:

```
helm3 repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm3 repo update
helm3 install ingress-controller ingress-nginx/ingress-nginx --namespace kube-system --values ingress-config.yaml
```

Kubernetes/ingress-nginx ingress controller `config.map` settings

To ensure that the IBM API Connect services have time to start, increase the `proxy-read-timeout` and `proxy-send-timeout` values, which are in seconds, in the `kubernetes/ingress-nginx` ingress controller `config.map` to at least the following:

- `proxy-read-timeout: "240"`
- `proxy-send-timeout: "240"`

Depending on your environment, you might need to increase these further if the IBM API Connect services do not start. If there is a load balancer in front of the worker nodes, then the load balancer configuration might also need to have extended timeouts.

Attention: In OpenShift, you must individually annotate all routes are for the Management subsystem (updating the corresponding configuration for an ingress controller affects all ingresses). Refer to the OpenShift docs on how to annotate the routes: <https://docs.openshift.com/container-platform/4.2/networking/routes/route-configuration.html>.

System and Software Requirements

The system and software requirements are described in the Software Product Compatibility Reports. See [Detailed system requirements for a specific product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Load balancer configuration in a Kubernetes deployment

When deploying API Connect for High Availability, it is recommended that you configure a cluster with at least three nodes and a load balancer. A sample configuration is provided for placing a load balancer in front of your API Connect Kubernetes deployment.

---

### About this task

API Connect can be deployed on a single node cluster. In this case the ingress endpoints are host names for which the DNS resolution points to the single IP address of the corresponding node hosting a particular subsystem, and no load balancer is required. For high availability, it is recommended to have at least a three node cluster. With three nodes, the ingress endpoints cannot resolve to a single IP address. A load balancer should be placed in front of an API Connect subsystem to route traffic.

Because it is difficult to add nodes once endpoints are configured, a good practice is to configure a load balancer even for single node deployments. With the load balancer in place, you can easily add nodes when needed. Add the node to the list of servers pointed to by the load balancer and the ingress endpoints defined during installation of API Connect can remain unchanged.

To support Mutual TLS communication between the API Connect subsystems, configure the load balancer with **SSL Passthrough** and **Layer 4** load balancing. In order for Mutual TLS to be performed directly by the API Connect subsystems, the load balancer should leave the packets unmodified, as is accomplished by Layer 4. Following is a description of the communication between the endpoints that are configured with Mutual TLS:

- API Manager (with the client certificate `portal-client`) communicates with the Portal Admin endpoint `portal-admin` (with the server certificate `portal-admin-ingress`)
- API Manager (with the client certificate `analytics-client-client`) communicates with the Analytics Client endpoint `analytics-client` (with the server certificate `analytics-client-ingress`)
- API Manager (with the client certificate `analytics-ingestion-client`) communicates with the Analytics Ingestion endpoint `analytics-ingestion` (with the server certificate `analytics-ingestion-ingress`)

Set endpoints to resolve to the load balancer

When configuring a load balancer in front of the API Connect subsystems, the ingress endpoints are set to host names that resolve to a load balancer, rather than to the host name of any specific node. For an overview of endpoints, see [Deployment overview for endpoints and certificates](#).

Use these example topologies as a guideline to determine the best way to configure the load balancer for your deployment.

---

### Procedure

- **Kubernetes cluster with wildcard DNS**

In this example, API Connect is deployed to a Kubernetes cluster with three master and six worker nodes. The master nodes have the Kubernetes API server listening on port 6443. The Kubernetes API Server is used for communicating with the `kubectl` command line interface. Note that it is generally not necessary for the Kubernetes API server (on port 6443 in this example) to be exposed outside the cluster through the load balancer, this is just one possible setup shown here.

The `kubernetes/ingress-nginx` ingress controller is deployed as a daemonset, so that every worker node in the cluster has an ingress controller pod listening on port 443. The API Connect subsystems (API Manager, Developer Portal, Analytics and Gateway) are all deployed on this same cluster.

Note:

When you configure a load balancer in front of a Management subsystem, specify timeouts of at least 240 seconds. Note that large deployments might need larger values.

The default timeout is typically 50 or 60 seconds, which is not long enough to avoid **409**

**Conflict** or **504 Gateway Timeout** errors. The **409 Conflict** error can occur when the time needed to complete an operation is sufficiently long that a second request gets issued.

For example, to specify 240 seconds when using HAProxy as a load balancer, set `timeout client` and `timeout server` to **240000**.

Best practice is to ensure that the same values are specified for the timeout settings for the load balancer and for the Kubernetes ingress controller. The ingress controller timeout settings are set in the ingress controller config.map. For more information, see the `proxy-read-timeout` and `proxy-send-timeout` settings in [Kubernetes ingress controller prerequisites](#).

A host running HAProxy acts as the load balancer, with a configuration for proxies in the HAProxy configuration file such as:

```
defaults
    log          global
    mode        http
    option      httplog
    option      dontlognull
    timeout connect 5000
    timeout client 240000
    timeout server 240000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend apiservers
    bind *:6443
    mode tcp
    option tcplog
    option forwardfor
    default_backend k8s_apiservers

frontend ingress
    bind *:443
    mode tcp
    option tcplog
    option forwardfor
    default_backend k8s_ingress

backend k8s_apiservers
    mode tcp
    option tcplog
    option ssl-hello-chk
    option log-health-checks
    default-server inter 10s fall 2
    server cluster1-master-1 10.169.241.226:6443 check
    server cluster1-master-2 10.169.241.163:6443 check
    server cluster1-master-3 10.169.241.153:6443 check

backend k8s_ingress
    mode tcp
    option tcplog
    option ssl-hello-chk
    option log-health-checks
    default-server inter 10s fall 2
    server cluster1-worker-1 10.169.241.142:443 check
    server cluster1-worker-2 10.169.241.150:443 check
    server cluster1-worker-3 10.169.241.188:443 check
    server cluster1-worker-4 10.169.241.196:443 check
    server cluster1-worker-5 10.169.241.168:443 check
    server cluster1-worker-6 10.169.241.156:443 check
```

A wildcard DNS record allows the resolution of `*.example.com` to the IP address of the HAProxy host.

The ingress endpoints are defined as:

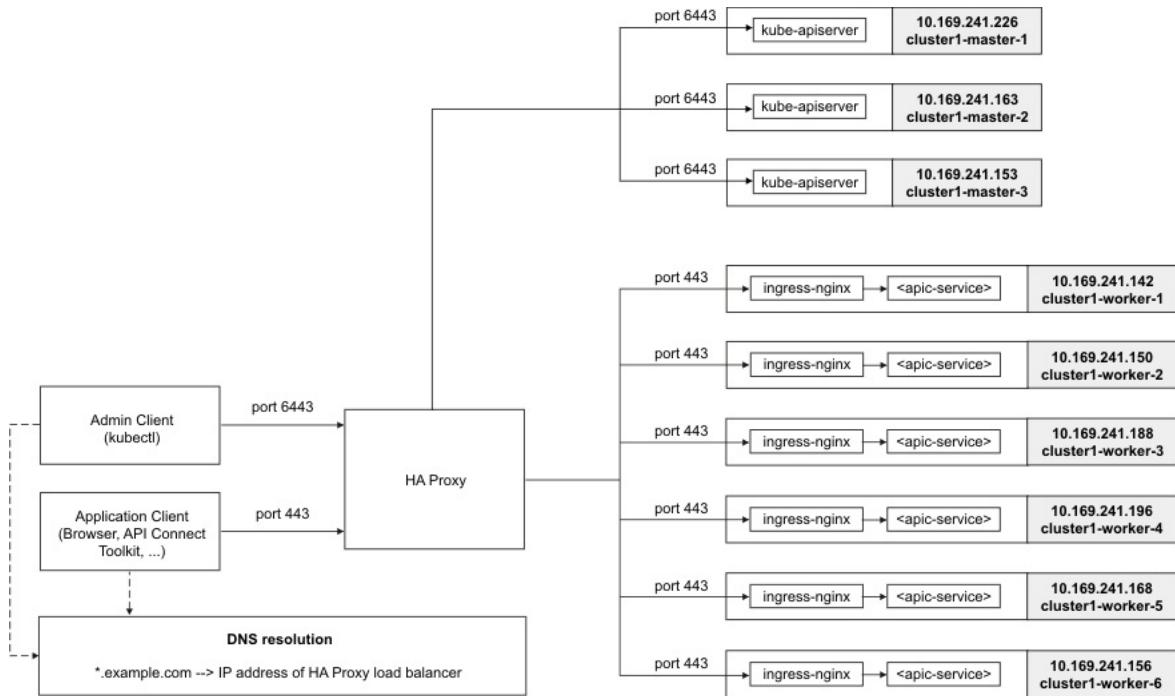
```
# API Manager
apicup subsys set mgmt api-manager-ui=apim.example.com
apicup subsys set mgmt cloud-admin-ui=apim.example.com
apicup subsys set mgmt consumer-api=apim.example.com
apicup subsys set mgmt platform-api=apim.example.com

# Developer Portal
apicup subsys set ptl portal-www=portal.example.com
apicup subsys set ptl portal-admin=portal-admin.example.com

# Analytics
apicup subsys set a7s analytics-client=a7s-client.example.com
apicup subsys set a7s analytics-ingestion=a7s-ingestion.example.com

# Gateway
apicup subsys set gw api-gateway=gw.example.com
apicup subsys set gw apic-gw-service=gwd.example.com
```

The following diagram illustrates the wildcard DNS example:



- **Kubernetes cluster without wildcard DNS**

This example uses the same topology as the previous one, except that there is no wildcard DNS resolution. Instead, individual DNS records are added, all pointing to the IP address of the load balancer. For example: `portal.example.com` and `admin.portal.example.com` resolve to the IP address of the load balancer.

The Developer Portal ingress endpoints can be defined as:

```
# Developer Portal
apicup subsys set ptl portal-www=portal.example.com
apicup subsys set ptl portal-admin=admin.portal.example.com
```

- **Kubernetes cluster with `xip.io` or `nip.io`**

This example uses the same topology as the first one, except that there is no specific DNS resolution configured and instead a service such as `xip.io` or `nip.io` provides wildcard DNS resolution. It is not recommended to use this approach for a production setup, as the ingress endpoints configured during installation of API Connect would be tied to the IP address of the load balancer. This approach can be useful when configuring a test deployment in situations where access to DNS records is not practical or would introduce delay in the deployment. If the load balancer IP is `192.168.100.100`, then the ingress endpoints are configured as:

```
# Developer Portal
apicup subsys set ptl portal-www=portal.192.168.100.100.xip.io
apicup subsys set ptl portal-admin=portal-admin.192.168.100.100.xip.io
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring logging for a Kubernetes deployment

Logs must be collected for both running and terminated containers and processes. Logging collection is required for IBM Support to assist with troubleshooting. For container-based deployments (Kubernetes), specific log commands and mechanisms vary depending on a customer's environment.

### About this task

For container-based deployments that use Kubernetes, collect logs by using a method suitable for your environment. You can run the following command:

```
kubect1 logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER]
```

for each pod in the Kubernetes cluster. For more information about Kubect1 documentation, see <https://kubernetes.io/docs/reference/generated/kubect1/kubect1-commands#logs>

However, the logs that are generated by the `kubect1 logs` command do not include terminated pods and processes. Use ELK, or another logging infrastructure, to gather logs that include terminated pods and processes. For more information, see <https://www.elastic.co/elk-stack>.

Specific log commands and mechanisms vary depending on a customer's environment. Log collection must be enabled for all running and terminated containers.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Installing API Connect into a Kubernetes environment

You can use Kubernetes and Docker containers to deploy and run API Connect. The Install Assist utility program automates the installation process into a Kubernetes runtime environment using an installation program called APICUP.

## Before you begin

- **First steps for installing API Connect: Upload files to registry**  
Before installing API Connect into a Kubernetes environment, you must first download the tar files and then upload them to your registry.
- **Installing the Management subsystem into a Kubernetes environment**  
Use the Install Assist utility program to install the Management subsystem into your Kubernetes runtime environment.
- **Installing the Analytics subsystem into a Kubernetes environment**  
Use the Install Assist utility program with the APICUP installer to install the Analytics subsystem into your Kubernetes runtime environment.
- **Installing the Developer Portal subsystem into a Kubernetes environment**  
Use the Install Assist utility program to install the Developer Portal subsystem into your Kubernetes runtime environment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## First steps for installing API Connect: Upload files to registry

Before installing API Connect into a Kubernetes environment, you must first download the tar files and then upload them to your registry.

## Before you begin

Before starting the installation into a Kubernetes runtime environment, complete the following tasks:

- Ensure you have supported software as described in the IBM® Software Product Compatibility Report for your version of API Connect. See [Detailed system requirements for a specific product](#).
- Complete the installation requirements described in [Requirements for deploying IBM API Connect into a Kubernetes environment](#).
- Install and run Docker on the local machine being used for API Connect installation.
- Login to your image registry (for example, Artifactory).

Note:

- Keep a record of the DNS entries mapped to the endpoints for each API Connect subsystem. You will need to map the DNS entries to new IP addresses when restoring the management database to a new cluster. See [Restoring the management database in a Kubernetes environment](#).
- The instructions in this section apply to an initial installation of API Connect into a Kubernetes environment. If you already have API Connect installed in a Kubernetes environment, you perform an upgrade. The upgrade instructions are available here: [Upgrading API Connect in a Kubernetes runtime environment](#)
- The Management server and Gateway firmware versions must match, for example, API Connect 2018.4.1.3 and DataPower 2018.4.1.3.
- If you plan to migrate a Version 5 deployment to Version 2018, see [Migrating a Version 5 deployment](#) before you start the installation process.

In these First Steps, you will use the APICUP installer included in Install Assist to create a project directory that contains the configuration file `apiconnect-up.yml`. Only one project directory is allowed. The configuration for all subsystems is stored in a single `apiconnect-up.yml` file contained in that project directory. All subsystems use the same project directory in order to update `apiconnect-up.yml`.

The first steps for a new Kubernetes installation are:

- Download the tar files containing the latest version of the API Connect subsystems and the Install Assist installer from Fix Central
- Upload the files to your registry
- Initialize **one** project directory for all subsystems

## Procedure

Download the appropriate files and then upload them to your image registry.

1. To obtain the most recent release, download the latest Fix Pack from IBM® Fix Central. For new installations, select ALL to list all versions, and then choose the latest version. For existing installations, choose your current installed version number to obtain the upgrade files. See [IBM Fix Central](#).
  - a. You will need a tar file for each subsystem and the Install Assist file for your operating system. Install Assist is an easy-to-use tool that automatically manages certificate creation, deployment configuration, and other technical aspects of configuring an IBM API Connect installation. The Install Assist download file contains the APICUP installation program, which automates the installation process. Following are the available files for a Kubernetes deployment:

Description	File name
IBM API Connect Management V2018.4.x.x Containers	management-images-kubernetes_lts_v2018.4.x.x.tgz
IBM API Connect Developer Portal V2018.4.x.x Containers	portal-images-kubernetes_lts_v2018.4.x.x.tgz
IBM API Connect Analytics V2018.4.x.x Containers	analytics-images-kubernetes_lts_v2018.4.x.x.tgz
IBM API Connect Install Assist V2018.4.x.x for Linux	apicup-linux_lts_v2018.4.x.x
IBM API Connect Install Assist V2018.4.x.x for Mac	apicup-mac_lts_v2018.4.x.x
IBM API Connect Install Assist V2018.4.x.x for Windows	apicup-windows_lts_v2018.4.x.x
Production DataPower Gateway	idg_dk20184xx.lts.prod.tar.gz

Description	File name
Nonproduction DataPower Gateway	idg_dk20184xx.lts.nonprod.tar.gz
Kubernetes DataPower Monitor	dpm2018417.lts.tar.gz

The production version of the Gateway is for use in a production environment. The nonproduction version is for use in a test or development environment.

- b. The toolkit files (with or without API Designer) can be downloaded from Fix Central directly, or from the Cloud Manager and API Manager UIs after installation is completed. See [Installing the toolkit](#).

2. Upload API Connect files to the registry. Ensure that the registry has sufficient disk space for the files.

For the Management, Portal, and Analytics subsystems, upload the tar files directly to the image registry using the `registry-upload` command. The `registry-upload` command creates the tags required to push the files to the registry, as follows:

```
apicup registry-upload <subsystem_type> <image_tgz> <registry_host>
```

where:

- `subsystem_type` is the name of subsystem for the file you are uploading, one of `management`, `portal`, or `analytics`.
- `image_tgz` is the name of the tar file you are uploading
- `registry_host` is the hostname of the target registry where you are uploading tar files. For IBM Container Service Registry, a namespace is required, for example: `us.icr.io/<namespace>` or `<region>.icr.io/<namespace>`. See [https://cloud.ibm.com/docs/services/Registry?topic=registry-registry-overview#registry\\_regions\\_local](https://cloud.ibm.com/docs/services/Registry?topic=registry-registry-overview#registry_regions_local)

3. Upload the DataPower Gateway images and any necessary supporting files.

Note: For API Connect Version 2018.4.1.7 or earlier, production deployments that use v5-compatible Gateway Servers also require an Application Optimization module. For these deployments, do not use Step 3.a. Instead, use the alternative instructions in the IBM technical note:

<https://www.ibm.com/support/pages/node/1283110>.

Note that for Version 2018.4.1.8 or later, all deployments use Step 3.a on this page.

- a. Upload the DataPower Gateway image (either Production or Nonproduction) that you downloaded in Step 1.

- i. Load the Gateway image to your local docker registry:

```
docker load -i idg_dk2018417.lts.prod.tar.gz
```

For these examples, the version number is 2018.4.1.7 and the production (`-prod`) version is downloaded. The image is automatically tagged in the docker registry to `ibmcom/datapower:2018.4.1.7.312001-prod`.

- ii. Re-tag the image for the new registry with the registry host, image name, and tag in the following format:

```
docker tag <existing REGISTRY_HOST>/<IMAGE_NAME>:<TAG> <new REGISTRY_HOST>/<IMAGE_NAME>:<TAG>
```

Table 1. Docker tag parameters

Parameter	Description
<REGISTRY_HOST>/<IMAGE_NAME>	Enter your registry host name and an image name, for example, <i>My Registry/MyImage</i> . The <REGISTRY_HOST>/<IMAGE_NAME> matches the value entered for installation in the <code>apicup subsys set gwy image-repository</code> command.
<TAG>	Enter a tag for the image, for example, <code>2018.4.1.7-312001-release-prod</code> . <ul style="list-style-type: none"> <li>• The &lt;TAG&gt; value must match the value entered for the gateway image tag during installation using the <code>apicup subsys set gwy image-tag</code> command.</li> <li>• You can look up required tag using the <code>./apicup version --images</code> command to list all images required by APICUP.</li> <li>• The tag must match the tag for the <code>datapower-api-gateway</code> image, for example: <code>apiconnect/datapower-api-gateway:2018.4.1.7-312001-release-prod</code>.</li> </ul>

For example:

```
docker tag ibmcom/datapower:2018.4.1.7.312001-prod MyRegistry/datapower-api-gateway:2018.4.1.7-312001-release-prod
```

- iii. Push the Gateway image to the new registry: `docker push`

<REGISTRY\_HOST>/<IMAGE\_NAME>:<TAG>. For example:

```
docker push MyRegistry/datapower-api-gateway:2018.4.1.7-312001-release-prod
```

- iv. The image is now added to your registry. You can now upload the DataPower Monitor image. Continue with Step 3.b.

- b. For the Gateway Kubernetes DataPower Monitor, load the images into your local Docker registry, re-tag them, and push them to the image repository, as follows:

- i. Load the Kubernetes DataPower Monitor image to your local docker registry:

```
docker load -i dpm2018417.lts.tar.gz
```

The image is automatically tagged in the docker registry:

```
ibmcom/k8s-datapower-monitor:2018.4.1-9-00bcbcf
```

- ii. Re-tag the image for the new registry:

```
docker tag <existing REGISTRY_HOST>/<IMAGE_NAME>:<TAG> <new REGISTRY_HOST>/<IMAGE_NAME>:<TAG>
```

Table 2. Docker tag parameters

Parameter	Description
<REGISTRY_HOST>/<IMAGE_NAME>	Enter your registry host name and an image name, for example, <i>My Registry/k8s-datapower-monitor</i> . The <REGISTRY_HOST>/<IMAGE_NAME> matches the value entered for installation in the <code>apicup subsys set gwy monitor-image-repository</code> command.

Parameter	Description
<TAG>	Enter a tag for the image, for example, 2018.4.1-9-00bcbcf. <ul style="list-style-type: none"> <li>The &lt;TAG&gt; value must match the value entered for the gateway image tag during installation using the <code>apicup subsys set gwy monitor-image-tag</code> command.</li> <li>You can look up required tag using the <code>./apicup version --images</code> command to list all images required by APICUP.</li> <li>The tag must match the tag for the Kubernetes DataPower Monitor. For example: 2018.4.1-9-00bcbcf.</li> </ul>

For example:

```
docker tag ibmcom/k8s-datapower-monitor:2018.4.1-9-00bcbcf MyRegistry/k8s-datapower-monitor:2018.4.1-9-00bcbcf
```

- iii. Push the monitor pod image to the new registry: `docker push <REGISTRY_HOST>/<IMAGE_NAME>:<TAG>`. For example:

```
docker push MyRegistry/k8s-datapower-monitor:2018.4.1-9-00bcbcf
```

- c. For DataPower Gateway, BusyBox 1.29 is required.

Note: An internet connection is required to obtain the BusyBox image from Docker Hub. If you want to download the image and copy it to the system containing the DataPower image, follow these steps:

- i. Download the Docker image for BusyBox from a system with an internet connection:

```
docker pull busybox:1.29-glibc
```

- ii. Save the image to a tar file:

```
docker save busybox:1.29-glibc > ./busybox-1.29-glibc.tar
```

- iii. Copy the image to the same location as the DataPower image and load it:

```
docker load < ./busybox-1.29-glibc.tar
```

- iv. Tag the image and push it to your local registry:

```
docker tag busybox:1.29-glibc <MyRegistry>/busybox:1.29-glibc
docker push MyRegistry/busybox:1.29-glibc
```

- i. Login to Docker Hub <https://hub.docker.com/> [/busybox?tab=tags](https://hub.docker.com/_/busybox?tab=tags) to obtain BusyBox 1.29.

- ii. Tag the image and push it to your local registry:

```
docker pull busybox:1.29-glibc
docker tag busybox:1.29-glibc <MyRegistry>/busybox:1.29-glibc
docker push MyRegistry/busybox:1.29-glibc
```

The images for all subsystems and BusyBox are now stored in the appropriate registries. The registry locations will be set by APICUP commands during the installation process.

4. Create one project directory, for example, `myProject` and change directories to the `myProject` directory. Run the `apicup init` command to initialize the project directory. An `apiconnect-up.yml` configuration file will be created. The `apiconnect-up.yml` file contains all values entered during the installation process as you move forward.

For example:

```
mkdir ./myProject
cd ./myProject
apicup init
```

Important:

Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

The original project directory created with APICUP during the initial product installation (for example, `myProject`) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. Note that the endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored or upgraded system. A good practice is to back up the original project directory to a location from where it can always be retrieved.

## What to do next

Install the subsystems (Management, Gateway, Analytics, and Portal) for your API Connect cloud. You can install the subsystems in any order, but the installation of all subsystems must be completed before configuring your API Connect cloud using Cloud Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Installing the Management subsystem into a Kubernetes environment

Use the Install Assist utility program to install the Management subsystem into your Kubernetes runtime environment.

### Before you begin

Before installing the Management subsystem, complete the steps described in [First steps for installing API Connect: Upload files to registry](#). For instructions on using the APICUP installer, see [Tips and tricks for using APICUP](#).



## Procedure

- The commands for installing the Management subsystem are described in the following table. Sample code is also provided. These are the basic commands for installing an API Connect Management service in a Kubernetes cluster. Namespaces must be configured in Kubernetes before starting the installation process. We do not recommend that you use the default namespace. DNS domain names must be configured for the endpoints. Resource allocation sizes are recommended minimums and will depend upon your environment. It is highly recommended that you include the backup schedule for the management database.

Note:

- If the `create-crd` command is set to false, then the Kubernetes administrator is responsible for creating CRD's before installing the management subsystem. See [Manually creating a CustomResourceDefinition in a Kubernetes environment](#).
- Host names and DNS entries may not be changed on a cluster after the initial installation.
- Domain names that are used for endpoints cannot contain the underscore "\_" character. See [Pre-installation requirements](#).

A few lines in the following commands set multiple values with a single set command. For instructions on configuring multiple settings per command and other tips for using `apicup`, see [Tips and tricks for using APICUP](#)

Command	Values/Definition
<code>apicup init</code>	Initializes the <code>apicup</code> installation utility and creates the <code>apiconnect-up.yaml</code> configuration file
<code>apicup subsys create mgmt management -k8s</code>	Describes a management service in the Kubernetes cluster. The identifier <code>mgmt</code> is the name you have assigned to your management service. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character. The command <code>management</code> indicates that you are creating a management subsystem. The subsystem will be deployed in <code>dev</code> mode by default, unless <code>standard</code> mode is explicitly specified.
<code>apicup subsys set mgmt extra-values-file</code>	<path/file name> - Path to the extra values file. One extra values file is permitted. The extra values file requires a <code>.yaml</code> format. It contains the ingress annotations and other settings for the subsystem. For an example of an extra values file for each subsystem, see <a href="#">Creating an extra values file in a Kubernetes environment</a> .
<code>apicup subsys set mgmt ingress-type</code>	Must be explicitly set to <code>route</code> for an OpenShift environment. For a standard K8s environment, <code>ingress-type</code> defaults to <code>ingress</code> and there is no need to explicitly enter this parameter.
<code>apicup subsys set mgmt mode</code>	Determines whether the installation is for development, testing, and demo purposes or for a production environment. If not explicitly set, the mode will default to <code>dev</code> (development and testing) for versions 2018.4.1.4 and later. For versions 2018.4.1.3 and earlier, the default for mode is <code>standard</code> . <ul style="list-style-type: none"> <li><code>dev</code> - (Default) Use <code>dev</code> mode for development, testing, and demo purposes. It allows one instance of each subsystem to be created.</li> <li><code>standard</code> - Use <code>standard</code> mode for production environments. It allows three containers each containing one instance of the subsystem to be created, for a total of three instances.</li> </ul>
<code>apicup subsys set mgmt namespace</code>	<namespaceName> - Defines the namespace where the management service resides; every subsystem can be in a separate namespace. Valid namespaces are configured in Kubernetes prior to starting the API Connect installation.
<code>apicup subsys set mgmt registry</code>	<registry-url> - The location where you are running the image registry (for example, Artifactory)
<code>apicup subsys set mgmt registry-secret</code>	For example, <registrySecret> - Kubernetes secret with Docker credentials to authenticate with the image registry. The value for <code>registry-secret</code> must match the name of the secret created using the <code>kubectl create secret</code> command. It contains the Docker credentials for accessing the registry.
<code>apicup subsys set mgmt storage-class</code>	<storageClass> - Refers to a valid StorageClass object in your Kubernetes cluster that supports dynamic PersistentVolume provisioning. Static or manual PersistentVolume provisioning can be used, but it requires additional steps not covered in this documentation. Block storage is required. Select a block storage format of your choosing in order for API Connect components to be supported. GlusterFS is not recommended as a storage option due to severe performance degradation. For AWS, the gp2 and io1 types of Elastic Block Storage (EBS) are supported. Note that NFS is not supported.
<code>apicup subsys set mgmt platform-api</code>	<code>api.&lt;hostname&gt;.com</code> - The endpoint that is exposed by Kubernetes to access the admin and provider Platform REST APIs. The endpoints have to be resolvable by DNS. Requires a FQDN, in lowercase. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Refer to the deployment overview diagram for information about endpoints and certificates: <a href="#">Deployment overview for endpoints and certificates</a> .
<code>apicup subsys set mgmt api-manager-ui</code>	<code>apim.&lt;hostname&gt;.com</code> - The endpoint that is exposed by Kubernetes to access the API Manager user interface. Requires a FQDN, in lowercase. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> .
<code>apicup subsys set mgmt cloud-admin-ui</code>	<code>cm.&lt;hostname&gt;.com</code> - The endpoint that is exposed by Kubernetes to access the Cloud Manager user interface. Requires a FQDN, in lowercase. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> .
<code>apicup subsys set mgmt consumer-api</code>	<code>consumer.&lt;hostname&gt;.com</code> - The endpoint that is exposed by Kubernetes to access the consumer REST APIs. Requires a FQDN, in lowercase. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> .
<code>cassandra-backup-auth-user</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The username for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Key ID.
<code>cassandra-backup-auth-pass</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The password for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Access Key parameter. The password will be stored in Base64 encoded format. For example: <pre>apicup subsys set mgmt cassandra-backup-auth-pass '&lt;password&gt;'</pre>
<code>cassandra-backup-host</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The fully qualified domain name of the backup server, in lowercase only. Ensure that the Kubernetes nodes can access this host. If using object store, enter <code>Endpoint/Region</code> . (The "/" character between the endpoint and region are required for this setting.)

Command	Values/Definition												
<code>cassandra-backup-port</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The port for the protocol to connect to the <code>cassandra-backup-host</code> . The backup port is not required for object storage.												
<code>cassandra-backup-protocol</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The backup protocol. Specify one of the following values: <ul style="list-style-type: none"> <li><code>sftp</code> - for secure file transfer protocol</li> <li><code>objstore</code> - for S3 compatible object storage</li> </ul>												
<code>cassandra-backup-path</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) The full path to the directory where the backup files will be stored. For object storage ( <code>objstore</code> ), the path can be set to the <code>bucket</code> value or the <code>bucket/subfolder</code> value.												
<code>cassandra-backup-schedule</code>	(Optional, not required to complete the installation, but highly recommended to configure backups.) Cron like schedule for performing automatic backups. The format for the schedule is: <ul style="list-style-type: none"> <li>* * * * *</li> <li>-----</li> <li>     </li> <li>      +----- day of week (0 - 6) (Sunday=0)</li> <li>   +----- month (1 - 12)</li> <li>    +----- day of month (1 - 31)</li> <li>  +----- hour (0 - 23)</li> <li>+----- min (0 - 59)</li> </ul> For example: <code>30 22 * * 1</code> will perform backups at 10:30 pm on Mondays. The timezone for backups is that of the node on which the Cassandra pod is scheduled. The backup schedule defaults to <code>0 0 * * *</code> . This means a backup is run every day at midnight and minute zero UTC.  When you configure a host, if you do not specify a value for <code>cassandra-backup-schedule</code> , the default backup schedule is automatically set. Note that the default backup schedule is not set, and scheduled backups not enabled, until host configuration is completed.												
<code>apicup subsys set mgmt cassandra-max-memory-gb</code>	This command sets the total memory allocated to the Cassandra server. This setting is used to calculate the <code>MAX_HEAP_SIZE</code> value of the server which is always 47% of the total memory allocated to Cassandra server. The default and minimum allowed value for <code>cassandra-max-memory-gb</code> is 9GB. The following table compares values for memory allocation for the management database: <table border="1" data-bbox="370 800 1511 940"> <thead> <tr> <th>Memory allocated to Kubernetes node where Cassandra is deployed</th> <th>Recommended cassandra-max-memory-gb</th> <th>MAX_HEAP_SIZE (47% of cassandra-max-memory-gb value)</th> </tr> </thead> <tbody> <tr> <td>16GB</td> <td>9GB (default)</td> <td>4331m</td> </tr> <tr> <td>32GB</td> <td>16GB</td> <td>7700m</td> </tr> <tr> <td>64GB</td> <td>24GB</td> <td>11550m</td> </tr> </tbody> </table>	Memory allocated to Kubernetes node where Cassandra is deployed	Recommended cassandra-max-memory-gb	MAX_HEAP_SIZE (47% of cassandra-max-memory-gb value)	16GB	9GB (default)	4331m	32GB	16GB	7700m	64GB	24GB	11550m
Memory allocated to Kubernetes node where Cassandra is deployed	Recommended cassandra-max-memory-gb	MAX_HEAP_SIZE (47% of cassandra-max-memory-gb value)											
16GB	9GB (default)	4331m											
32GB	16GB	7700m											
64GB	24GB	11550m											
<code>apicup subsys set mgmt cassandra-cluster-size</code>	3 is a recommended minimum; actual value will depend upon your environment.												
<code>apicup subsys set mgmt cassandra-volume-size-gb</code>	The default value is 50GB, which is the minimum recommended setting for a non-production environment; the minimum recommended setting for a production environment is 250GB. The actual value will depend upon your environment. The <code>cassandra-volume-size-gb</code> parameter is a PER NODE value, so the total amount of storage space for the management database will be the <code>cassandra-volume-size-gb</code> multiplied by the number of nodes. <b>IMPORTANT:</b> The storage size for the management database on a node cannot be increased once the pod is deployed to the node.												
<code>apicup subsys set mgmt create-crd</code>	Options are <code>true</code> or <code>false</code> . The default value is <code>true</code> . When set to <code>true</code> , the custom resources required by the management subsystem are automatically created at the cluster level. The management database requires the following two CustomResourceDefinitions (CRDs): <ul style="list-style-type: none"> <li><code>cassandraclusters.apic.ibm.com</code> - stores the custom resource that contains the specs for the management database cluster</li> <li><code>cassandraclusterbackups.apic.ibm.com</code> - stores the custom resource that contains information for each completed backup</li> </ul> When set to <code>false</code> , the Kubernetes administrator must manually create the CRDs before installing the management subsystem. Note that CustomResourceDefinitions require ClusterRole privileges. For instructions on how to manually create the CRDs, see <a href="#">Manually creating a CustomResourceDefinition in a Kubernetes environment</a> .												
<code>apicup subsys set mgmt external-cassandra-host</code>	<host name> - Hostname where a Cassandra database cluster can be reached, if not using the Cassandra Operator bundled with API Connect.												
<code>apicup certs set mgmt CERT_NAME</code>	<ul style="list-style-type: none"> <li><code>[CERT_FILE KEY_FILE CA_FILE] [flags]</code> - Set the required certificates for the subsystem. Custom certificates may be set, otherwise, default certificates are automatically configured. For information on what certificates are available and how to set them see: <a href="#">Working with certificates</a>.</li> <li><code>[KEY_FILE] [flags]</code> - Set the encryption secret for the management database. Important: The encryption secret can only be set once and only during initial installation. See <a href="#">Setting the encryption-secret for the management database</a>.</li> </ul>												
<code>apicup subsys set mgmt license-version &lt;license_type&gt;</code>	<license_type> - Specify the type of license, either <code>Production</code> or <code>Nonproduction</code> . If not specified, the default value is <code>Nonproduction</code> . Note that the license type does not impact which Management server image to install. The API Connect Kubernetes Pod annotations <code>productID</code> and <code>productName</code> reflect the selected license.												
<code>apicup subsys install</code>	<code>mgmt</code> - Starts the installation of the management service.												

Note: API Connect v2018 cannot be deployed on NFS.

```

apicup subsys create mgmt management --k8s
apicup subsys set mgmt extra-values-file=<path/file name>
apicup subsys set mgmt ingress-type=<route - for OpenShift environments only> <ingress - default>
apicup subsys set mgmt mode=<dev>
apicup subsys set mgmt namespace=<namespaceName>
apicup subsys set mgmt registry=<registry-url>

```

```

apicup subsys set mgmt registry-secret=<registrySecret>
apicup subsys set mgmt storage-class=<storageClass>

apicup subsys set mgmt platform-api=my.platform.com api-manager-ui=my.apim.com cloud-admin-ui=my.cloud.com consumer-
api=my.consumer.com

apicup subsys set mgmt cassandra-backup-auth-user=MyUsername cassandra-backup-auth-pass 'MyPassword'
apicup subsys set mgmt cassandra-backup-host=<hostname>
apicup subsys set mgmt cassandra-backup-path=</backups>
apicup subsys set mgmt cassandra-backup-port=<22>
apicup subsys set mgmt cassandra-backup-protocol=<sftp, objstore>
apicup subsys set mgmt cassandra-backup-schedule=<"0 0 * * *">
apicup subsys set mgmt cassandra-max-memory-gb=9 cassandra-cluster-size=3
apicup subsys set mgmt cassandra-volume-size-gb=<50>
apicup subsys set mgmt create-crd=<true>
apicup subsys set mgmt external-cassandra-host=<hostname>

//Set the certificates
apicup certs set mgmt CERT_NAME=[CERT_FILE KEY_FILE CA_FILE] [flags] <sets certificates>
apicup certs set mgmt CERT_NAME=[KEY_FILE] [flags] <sets the encryption secret>

// OPTIONAL: Write the plan to an output directory to inspect myProject/install-plan prior to installation
apicup subsys install mgmt --out=mgmt-install-plan
// Start the installation from the output directory. Enter the full path to the output directory.
apicup subsys install mgmt --plan-dir=./myProject/mgmt-install-plan
//If install-plan is not used, enter the following command to start the installation
apicup subsys install mgmt

```

Important: The management database requires mmap counts higher than the operating system defaults. To change the mmap counts on the live system, run `sudo sysctl -w vm.max_map_count=1048575` on each Kubernetes node. To persist this change when node restarts occur, add the following to the `/etc/sysctl.conf` file: `vm.max_map_count = 1048575`.

- You can use the `get` command to print out the settings for the subsystem prior to installing it. For example, `apicup subsys get mgmt`, where `mgmt` is the name of the subsystem. The `get` command displays any default settings for the subsystem and the settings from the `apiconnect-up.yml` file. The `Errors` column indicates if a parameter is invalid. Following is an example of output received from `apicup subsys get mgmt`:

#### Kubernetes settings

Name	Value	Errors
extra-values-file		
ingress-type	ingress	
mode	standard	
namespace	default	
registry		
registry-secret		
storage-class		

#### Subsystem settings

Name	Value	Errors
cassandra-backup-auth-pass		
cassandra-backup-auth-user		
cassandra-backup-host		
cassandra-backup-path	/backups	
cassandra-backup-port	22	
cassandra-backup-protocol	sftp	
cassandra-backup-schedule	0 0 * * *	
cassandra-cluster_size	1	
cassandra-max-memory-gb	9	
cassandra-volume-size-gb	100	
create-crd	true	
external-cassandra-host		

#### Endpoints

Name	Value	Errors
api-manager-ui		api-manager-ui is not a valid FQDN
cloud-admin-ui		cloud-admin-ui is not a valid FQDN
consumer-api		consumer-api is not a valid FQDN
platform-api		platform-api is not a valid FQDN

- Correct any errors indicated by the output from the `get` command and then run `apicup subsys install <SUBSYS>` to install the subsystem.
- After running the `apicup subsys install <SUBSYS>` command, run the `apicup subsys get <SUBSYS> --validate` to validate the installation. You must correct any errors indicate by `--validate` prior to continuing.

#### Kubernetes settings

Name	Value	
extra-values-file		✓
ingress-type	ingress	✓
mode	standard	✓
namespace	prod-test	✓
registry	docker-local.artifactory.swg-dev.com/apicup-imgs/2018.4.1-550	✓
registry-secret	apiconnect-image-pull-secret	✓
storage-class	rook-block	✓

Subsystem settings	
Name	Value
az-name	default-az
cassandra-backup-auth-pass	
cassandra-backup-auth-user	
cassandra-backup-host	
cassandra-backup-path	/backups
cassandra-backup-port	22
cassandra-backup-protocol	sftp
cassandra-backup-schedule	0 0 * * *
cassandra-cluster-size	3
cassandra-max-memory-gb	9
cassandra-volume-size-gb	50
create-crd	true
external-cassandra-host	

Endpoints	
Name	Value
api-manager-ui	apim.test.company.com
cloud-admin-ui	cm.test.company.com
consumer-api	consumer.test.company.com
platform-api	api.test.company.com

## What to do next

Continue your API Connect installation in a Kubernetes runtime environment by installing the other subsystems.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Installing the Analytics subsystem into a Kubernetes environment

Use the Install Assist utility program with the APICUP installer to install the Analytics subsystem into your Kubernetes runtime environment.

### Before you begin

Before installing the Analytics subsystem, complete the steps described in [First steps for installing API Connect: Upload files to registry](#). For instructions on using the APICUP installer, see [Tips and tricks for using APICUP](#).

### About this task

By default, the Analytics subsystem is configured to store data so that users can review it in the Analytics user interface. After deploying the subsystem, you can optionally configure it to offload some data to a third-party service for review and storage, as explained in [Configuring analytics offload for API Connect](#). Any data that is not offloaded remains accessible from the Analytics user interface.

If you want to offload all Analytics data, you can optionally install the subsystem with the ingestion-only configuration. In this scenario, unused Analytics components (such as analytics-storage and analytics-client) are omitted from the topology. Only the components that are required for offloading data are deployed. The reduced topology requires less CPU, memory, and storage.

If you do not configure ingestion-only during installation, you can enable it later as explained in [Enabling Analytics ingestion-only on Kubernetes](#).

Settings that are required for the ingestion-only configuration are noted in the steps that follow.

## Procedure

- The commands for installing the Analytics subsystem are described in the following table. Sample code is also provided.

Important: There are known GlusterFS issues that affect the following functionality areas:

- Elasticsearch, which is used in the API Connect analytics service.
- The Developer Portal service.

If you use GlusterFS, you might encounter severe performance degradation and, in some cases, loss of data.

Note: Host names and DNS entries may not be changed on a cluster after the initial installation.

Command	Values/Definition
<code>apicup init</code>	Initializes the apicup installation utility and creates the <code>apiconnect-up.yml</code> configuration file.

Command	Values/Definition
<code>apicup subsys create analyt analytics -- k8s</code>	Describes an analytics subsystem in the Kubernetes cluster. The identifier <code>analyt</code> is the name you have assigned to your analytics service. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character. The command <code>analytics</code> indicates that you are creating an analytics subsystem.
<code>apicup subsys set analyt extra-values-file</code>	<p>&lt;path/file name&gt; - Path to the extra values file. One extra values file is permitted. The extra values file requires a <code>.yaml</code> format. It contains the ingress annotations and other settings for the subsystem. For an example of an extra values file for each subsystem, see <a href="#">Creating an extra values file in a Kubernetes environment</a>.</p> <p>For the ingestion-only configuration, specify the URL of the offload endpoint where all analytics data will be routed. You can choose the name for the file, but you must use <code>.yaml</code> as the file-name extension. Format the file as shown in the following example, making sure to include required settings for the third-party system.</p> <pre>apic-analytics-ingestion:   outputOffload:  -     elasticsearch {       hosts =&gt; "http://offload_endpoint_URL:port"       index =&gt; "api-call"     }</pre>
<code>apicup subsys set analyt analytics-ingestion</code>	<ai>.<hostname>.<domainname> - The <code>analytics-ingestion</code> endpoint is used by the Gateway service to push data to the Analytics service. All endpoints have to be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Refer to the deployment overview diagram for information about endpoints and certificates: <a href="#">Deployment overview for endpoints and certificates</a> .
<code>apicup subsys set analyt ingestion-only=true</code>	This setting is required for the ingestion-only configuration, and the value must be set to <code>true</code> .
<code>apicup subsys set analyt analytics-client</code>	<ac>.<hostname>.<domain name> - The management endpoint that is defined by <code>analytics-client</code> , for example, <ac>.<hostname>.<domain name>. This is the analytics service endpoint used by Cloud Manager, API Manager, and the Portal to communicate with the analytics service and is entered when configuring an analytics service in Cloud Manager. See <a href="#">Registering an analytics service</a> . All endpoints have to be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt namespace</code>	<namespaceName> - Defines the namespace where the analytics service resides; every subsystem can be in a separate namespace. Valid namespaces are configured in Kubernetes prior to starting the API Connect installation
<code>apicup subsys set analyt registry</code>	<registry-uri> - The location where you are running the image registry (for example, Artifactory).
<code>apicup subsys set analyt registry-secret</code>	For example, <registrySecret> - Kubernetes secret with Docker credentials to authenticate with the image registry. The value for <code>registry-secret</code> must match the name of the secret created using the <code>kubect1 create secret</code> command. It contains the Docker credentials for accessing the registry.
<code>apicup subsys set analyt coordinating-max-memory-gb</code>	The default is 6GB. The recommended minimum is 12GB. The actual value will depend upon your environment. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt data-max-memory-gb</code>	The default is 6GB. The recommended minimum is 12GB. The actual value will depend upon your environment. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt data-storage-size-gb</code>	The default is 200GB. The recommended minimum is 200GB. The actual value will depend upon your environment. The <code>data-storage-size-gb</code> value is the amount of storage used per pod. A default installation will contain three data pods. In this example, 600GB of total space is required for the data nodes. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt master-max-memory-gb</code>	The default is 6GB. The recommended minimum is 12GB. The actual value will depend upon your environment. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt master-storage-size-gb</code>	The default is 5GB. The recommended minimum is 5GB. The actual value will depend upon your environment. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt enable-message-queue</code>	Options are <code>true</code> or <code>false</code> . The default is <code>false</code> . When set to true, the message queue will be activated and the analytics pipeline will be configured to use it. See <a href="#">Configuring the analytics message queue</a> . The message queue requires at least one 5GB PVC for each pod.
<code>apicup subsys set analyt storage-class</code>	<storageClass> - Refers to a valid StorageClass object in your Kubernetes cluster that supports dynamic PersistentVolume provisioning. Static or manual PersistentVolume provisioning can be used, but it requires additional steps not covered in this documentation. You can use local storage or block storage. For block storage, select a block storage format of your choosing in order for API Connect components to be supported. GlusterFS is not recommended as a storage option due to severe performance degradation. For AWS, the <code>gp2</code> and <code>io1</code> types of Elastic Block Storage (EBS) are supported. Note that NFS is not supported.

Command	Values/Definition
<code>apicup subsys set analyt es-storage-class</code>	<es-storageClass> - Optional Elasticsearch storage class. If the <code>es-storage-class</code> is set, analytics storage ignores other storage classes and uses the <code>es-storage-class</code> . If not set, then analytics storage uses the storage class object set by the <code>apicup subsys set analyt storage-class</code> command. This setting is not needed for the ingestion-only configuration. For more information, see the note that follows this table.
<code>apicup subsys set analyt mq-storage-class</code>	<mq-storageClass> - Optional message queue storage class. If the <code>mq-storage-class</code> is set, the message queue ignores other storage classes and uses the <code>mq-storage-class</code> . If not set, then the message queue uses the storage class object set by <code>apicup subsys set analyt storage-class</code> .
<code>apicup subsys set analyt mode</code>	Determines whether the installation is for development, testing, and demo purposes or for production. If not explicitly set, the mode will default to <code>dev</code> (development and testing) for versions 2018.4.1.4 and later. For versions 2018.4.1.3 and earlier, the default for mode is <code>standard</code> . <ul style="list-style-type: none"> <li><code>dev</code> - (Default) Use <code>dev</code> mode for development, testing, and demo purposes. It allows one instance of each subsystem to be created.</li> <li><code>standard</code> - Use <code>standard</code> mode for production environments. It allows multiple instances of each subsystem to be created.</li> </ul>
<code>apicup subsys set analyt ingress-type</code>	Must be explicitly set to <code>route</code> for an OpenShift environment. For a standard K8s environment, <code>ingress-type</code> defaults to <code>ingress</code> and there is no need to explicitly enter this parameter.
<code>apicup certs set analyt CERT_NAME</code>	[CERT_FILE KEY_FILE CA_FILE] [flags] - Set the required certificates for the subsystem. Custom certificates may be set, otherwise, default certificates are automatically configured. For information on what certificates are available and how to set them see: <a href="#">Working with certificates</a> .
<code>apicup subsys set analyt license-version &lt;license_type&gt;</code>	<license_type> - Specify the type of license, either <code>Production</code> or <code>Nonproduction</code> . If not specified, the default value is <code>Nonproduction</code> . Note that the license type does not impact which Analytics image to install. The API Connect Kubernetes Pod annotations <code>productID</code> and <code>productName</code> reflect the selected license.
<code>apicup subsys install</code>	<code>analyt</code> - Starts the installation of the analytic service

Note: If a setting is not required for the ingestion-only configuration, you can omit it. A default value is supplied automatically by `apicup` but the value is ignored during installation. When you run the validation step, the value is not actually validated but is marked as valid in the results.

Example 1: Installation commands for deployment that does not use the ingestion-only configuration.

```
apicup subsys create analyt analytics --k8s
apicup subsys set analyt extra-values-file=<path/file name>
apicup subsys set analyt analytics-ingestion=<ai>.<hostname>.<domainname>
apicup subsys set analyt analytics-client=<ac>.<hostname>.<domainname>
apicup subsys set analyt registry=<registry-uri>
apicup subsys set analyt namespace=<namespaceName>
apicup subsys set analyt registry-secret=<registrySecret>
apicup subsys set analyt coordinating-max-memory-gb=6
apicup subsys set analyt data-max-memory-gb=6
apicup subsys set analyt data-storage-size-gb=200
apicup subsys set analyt master-max-memory-gb=6
apicup subsys set analyt master-storage-size-gb=5
apicup subsys set analyt enable-message-queue=<false>
apicup subsys set analyt storage-class=<storageClass>
apicup subsys set analyt es-storage-class=<es-storageClass>
apicup subsys set analyt mq-storage-class=<mq-storageClass>
apicup subsys set analyt mode=dev
apicup subsys set analyt ingress-type=route <for OpenShift environments only>

//Set the certificates
apicup certs set analyt CERT_NAME=[CERT_FILE KEY_FILE CA_FILE] [flags]

// OPTIONAL: Write the plan to an output directory to inspect myProject/install-plan prior to installation
apicup subsys install analyt --out=analyt-install-plan
// Start the installation from the output directory. Enter the full path to the output directory.
apicup subsys install analyt --plan-dir=./myProject/analyt-install-plan

//If output file is not used, enter the following command to start the installation
apicup subsys install analyt
```

Example 2: Installation commands for the ingestion-only configuration.

```
apicup subsys create analyt analytics --k8s
apicup subsys set analyt extra-values-file=extra-values-analyt.yaml
apicup subsys set analyt ingestion-only=true
apicup subsys set analyt analytics-ingestion=<ai>.<hostname>.<domainname>
apicup subsys set analyt registry=<registry-uri>
apicup subsys set analyt namespace=<namespaceName>
apicup subsys set analyt registry-secret=<registrySecret>
apicup subsys set analyt enable-message-queue=<false>
apicup subsys set analyt storage-class=<storageClass>
apicup subsys set analyt mq-storage-class=<mq-storageClass>
apicup subsys set analyt mode=dev
apicup subsys set analyt ingress-type=route <for OpenShift environments only>
apicup subsys set analyt storage-class=<storageClass>
apicup subsys set analyt mq-storage-class=<mq-storageClass>

//Set the certificates
apicup certs set analyt CERT_NAME=[CERT_FILE KEY_FILE CA_FILE] [flags]

// OPTIONAL: Write the plan to an output directory to inspect myProject/install-plan prior to installation
apicup subsys install analyt --out=analyt-install-plan

// Start the installation from the output directory. Enter the full path to the output directory.
apicup subsys install analyt --plan-dir=./myProject/analyt-install-plan
```

//If output file is not used, enter the following command to start the installation  
 apicup subsys install analyt

An analytics installation starts the following Kubernetes pods:

- Full installation in **standard** mode:

Table 1. Kubernetes pods in a full Analytics installation

Number	Pod
2	client
2	ingestion
2	mtls
1	operator
3	storage-master
3	storage-coordinating
3	storage-data

Note that in **dev** mode, there is only one of each pod.

- Message queue deployment in **standard** mode: All of the pods shown in Table [Table 1](#), plus:

Table 2. Additional pods used for Message Queue deployments

Number	Pod
3	mq-kafka
3	mq-zookeeper

Note that in **dev** mode there is only one of each pod, with the same capability to enable or disable message queue.

- Ingestion-only installation in **standard** mode:

Table 3. Kubernetes pods in an ingestion-only Analytics installation

Number	Pods
2	ingestion
2	mtls

Note that in **dev** mode there is only one of each pod.

Important: When the Analytics service is configured to store data, (that is, it is not configured for ingestion-only), the service depends on Elasticsearch which requires map counts higher than the operating system defaults. To change the map counts on the live system, run `sudo sysctl -w vm.max_map_count=262144` on each Kubernetes node. To persist this change when node restarts occur, add the following to the `/etc/sysctl.conf` file: `vm.max_map_count = 262144`. This is the minimum recommended value. For more information see <https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html>

Memory settings listed are maximum sizes. The Kubernetes memory resource *request amount*, or amount of memory reserved for each analytics pod, is 3/4 of the maximum memory. Actual memory used will usually be less than the reserved amount, but depends on current analytics load.

Modifying allocation space after installation might not be possible depending on the type of persistent volume that is used.

2. After running the `apicup subsys install` command, run the `apicup subsys`

`get <analytics_subsys_name> --validate` to validate the installation. You must correct any errors indicate by `--validate` prior to continuing. Following is an example for the `--validate` output for a configuration that does not use the ingestion-only feature:

```

Kubernetes settings
=====
Name                               Value
----                               -
extra-values-file                   ✓
ingress-type                         ingress ✓
mode                                 standard ✓
namespace                           prod ✓
registry                            docker-local.artifactory.swg-dev.com/apicup-imgs/2018.4.1-550 ✓
registry-secret                     apiconnect-image-pull-secret ✓
storage-class                       rook-block ✓

Subsystem settings
=====
Name                               Value
----                               -
coordinating-max-memory-gb         6 ✓
data-max-memory-gb                 6 ✓
data-storage-size-gb               200 ✓
enable-message-queue               false ✓
es-storage-class                   - ✓
master-max-memory-gb               6 ✓
master-storage-size-gb             5 ✓
mq-storage-class                   - ✓

Endpoints
=====
Name                               Value

```



----	-----	
analytics-client	ac.1.23.123.45.sample.io	✓
analytics-ingestion	ai.1.23.123.45.sample.io	✓

## What to do next

Continue your API Connect installation in a Kubernetes runtime environment by installing the other subsystems.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Installing the Developer Portal subsystem into a Kubernetes environment

Use the Install Assist utility program to install the Developer Portal subsystem into your Kubernetes runtime environment.

### Before you begin

Before installing the Portal subsystem, complete the steps that are described in [First steps for installing API Connect: Upload files to registry](#). For instructions on using the APICUP installer, see [Tips and tricks for using APICUP](#).

Tip: To enable effective high availability for your Portal service, you need a latency that is less than 50ms between all portal-db pods to avoid the risk of performance degradation. Servers with uniform specifications are required, as any write actions occur at the speed of the slowest portal-db pod, as the write actions are synchronous across the cluster of portal-db pods. It is recommended that there are three servers in each cluster of portal-db pods for the high availability configuration. The three servers can be situated in the same availability zone, or across three availability zones to ensure the best availability. However, you can configure high availability with two availability zones.

Note:

Ensure that your kernel or Kubernetes node has the value of its `inotify` watches set high enough so that the Developer Portal can monitor and maintain the files for each Developer Portal site. If set too low, the Developer Portal containers might fail to start or go into a `non-ready` state when this limit is reached. If you have many Developer Portal sites, or if your sites contain a lot of content, for example, many custom modules and themes, then a larger number of `inotify` watches are required. You can start with a value of 65,000, but for large deployments, this value might need to go up as high as 1,000,000. The Developer Portal containers take `inotify` watches only when they need them. The full number is not reserved or held, so it is acceptable to set this value high.

### Procedure

1. The commands for installing the Portal subsystem are described in the following table. Sample code is also provided.

Note:

- Although the `site-backup` parameters listed in the following table are optional, it's highly recommended that you include them in your installation, so that remote server backups of the portal database are also taken. If you don't want to configure these parameters at installation, you can configure them later, see [Backing up and restoring the Developer Portal in a Kubernetes environment](#).
- Host names and DNS entries may not be changed on a cluster after the initial installation.
- The backup secret is a Kubernetes secret that contains your username and password for your backup database (sftp/s3). Only password-based authentication is supported for sftp and s3, not authentication based on public certificates and private keys. Password-based authentication for s3 requires that you generate an access key and secret. For example:
  - IBM (Cloud Object Storage): [Service credentials](#).
  - AWS: [Managing access keys](#).

Command	Values/Definition
<code>apicup init</code>	Initializes the apicup installation utility and creates the <code>apiconnect-up.yml</code> configuration file
<code>apicup subsys create ptl portal --k8s</code>	Describes a Portal subsystem in the Kubernetes cluster. The identifier <code>ptl</code> is the name that you have assigned to your Portal service. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character. The command <code>portal</code> indicates that you are creating a Portal subsystem.
<code>apicup subsys set ptl extra-values-file</code>	<code>&lt;path/file name&gt;</code> - Path to the extra values file. One extra values file is permitted. The extra values file requires a <code>.yaml</code> format. It contains the ingress annotations and other settings for the subsystem. For an example of an extra values file for each subsystem, see <a href="#">Creating an extra values file in a Kubernetes environment</a> .
<code>apicup subsys set ptl portal-admin</code>	<code>&lt;padmin&gt;.&lt;hostname&gt;.&lt;domainname&gt;</code> - The management endpoint for the Portal. All endpoints must be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Refer to the deployment overview diagram for information about endpoints and certificates: <a href="#">Deployment overview for endpoints and certificates</a> .
<code>apicup subsys set ptl portal-www</code>	<code>&lt;portal&gt;.&lt;hostname&gt;.&lt;domain name&gt;</code> - The endpoint for the public portal websites. All endpoints must be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Multiple portal-www endpoints may be configured, as described here: <a href="#">Defining multiple portal endpoints for a Kubernetes environment</a> .
<code>apicup subsys set ptl namespace</code>	<code>&lt;namespaceName&gt;</code> - Defines the namespace where the Portal service resides; every subsystem can be in a separate namespace. Valid namespaces are configured in Kubernetes prior to starting the API Connect installation
<code>apicup subsys set ptl storage-class</code>	<code>&lt;storageClass&gt;</code> - Refers to a valid StorageClass object in your Kubernetes cluster that supports dynamic PersistentVolume provisioning. Static or manual PersistentVolume provisioning can be used, but it requires additional steps that are not covered in this documentation. Block storage is required. Select a block storage format of your choosing in order for API Connect components to be supported. GlusterFS is not recommended as a storage option due to severe performance degradation. For AWS, the gp2 and io1 types of Elastic Block Storage (EBS) are supported. Note that NFS is not supported.
<code>apicup subsys set ptl registry</code>	<code>&lt;registry-uri&gt;</code> - The location where you are running the image registry (for example, Artifactory)



Command	Values/Definition
<code>apicup subsys set ptl registry-secret</code>	< <b>registrySecret</b> > - Kubernetes secret with Docker credentials to authenticate with the image registry. The value for <b>registry-secret</b> must match the name of the secret created by using the <code>kubectl create secret</code> command. It contains the Docker credentials for accessing the registry.
<code>apicup subsys set ptl www-storage-size-gb</code>	The default is 5GB. The recommended minimum is 5GB. The actual value depends on your environment. Used to store the php, html, and media files that comprise each Portal web site. Increase this size by at least 1GB for every 20 small sites that you want to host. Note that a single site with a lot of media can take many GB. If there is less than 128MB free, then the Portal will refuse to create new sites.
<code>apicup subsys set ptl backup-storage-size-gb</code>	The default is 5GB. The recommended minimum is 5GB. The actual value depends on your environment. Stores 7 days worth of compressed backups of each Portal website. Increase this by at least 3GB for every 20 small sites that you want to host. Note that a single site with a lot of media can take many GB per backup.
<code>apicup subsys set ptl db-storage-size-gb</code>	The default is 12GB. The recommended minimum is 12GB. The actual value might be higher than this but will depend upon your environment. Used for the database files. Increase this by at least 5GB for every 20 small sites that you want to host. Larger sites require more storage. If there is less than 2GB free, then the Portal refuses to create new sites. Note: For example, a Portal with 5 sites should set this value to 40 GB, or more. Using the default values is not recommended.
<code>apicup subsys set ptl db-logs-storage-size-gb</code>	The default is 2GB, and this size should not be changed from 2 GB. Storage used for the database logs. Reducing this value could result in the volume filling up and the database hanging. Making it larger will have no effect on the number of logs kept.
<code>apicup subsys set ptl admin-storage-size-gb</code>	The default is 1GB, and this size should not be changed from 1GB. Used for the internal state of the admin container. Reducing this value could result in the admin container failing. Making it larger will have no effect
<code>apicup subsys set ptl mode</code>	Determines whether the installation is for development, testing, and demo purposes, or for production. If not explicitly set, the mode will default to <b>dev</b> (development and testing) for versions 2018.4.1.4 and later. For versions 2018.4.1.3 and earlier, the default for mode is <b>standard</b> . <ul style="list-style-type: none"> <li><b>dev</b> - (Default) Use dev mode for development, testing, and demonstration purposes. It allows one instance of each subsystem to be created.</li> <li><b>standard</b> - Use standard mode for production environments. It allows multiple instances of each subsystem to be created.</li> </ul>
<code>apicup subsys set ptl ingress-type</code>	Must be explicitly set to <b>route</b> for an OpenShift environment. For a standard K8s environment, <b>ingress-type</b> defaults to <b>ingress</b> and there is no need to explicitly enter this parameter.
<code>apicup subsys set ptl site-backup-host</code>	The fully qualified domain name of the backup server, in lowercase only. Ensure that the Kubernetes nodes can access this host. If using object storage, enter <b>Endpoint/Region</b> . (The / character between the endpoint and region is required for the object storage setting.)
<code>apicup subsys set ptl site-backup-port</code>	The port for the protocol to connect to the <b>site-backup-host</b> . Defaults to <b>22</b> if not explicitly set. The backup port is not required for object storage.
<code>apicup subsys set ptl site-backup-auth-user</code>	The user name for the server specified in <b>site-backup-host</b> . If using object storage, the user name is the S3 Secret Key ID.
<code>apicup subsys set ptl site-backup-auth-pass</code>	The password for the server specified in <b>site-backup-host</b> . If using object storage, the password is the S3 Secret Access Key parameter. The password is stored in Base64 encoded format, and must not be edited directly in the apiconnect-up.yml file.
<code>apicup subsys set ptl site-backup-path</code>	The full path to the directory where the backup files are stored. For object storage, the path can be set to the <b>bucket</b> value or the <b>bucket/subfolder</b> value.
<code>apicup subsys set ptl site-backup-protocol</code>	The protocol that is used to communicate with your remote backup endpoint. Specify one of the following values: <ul style="list-style-type: none"> <li><b>sftp</b> - for secure file transfer protocol</li> <li><b>objstore</b> - for S3 compatible object storage</li> </ul> The default protocol is <b>sftp</b> . Note: The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <b>x509: certificate signed by unknown authority</b> .
<code>apicup subsys set ptl site-backup-schedule</code>	The schedule for how often automatic Portal backups are run. The format for the schedule is any valid cron string, as follows:  <pre> * * * * * - - - - -                   +----- day of week (0 - 6) (Sunday=0)       +----- month (1 - 12)     +----- day of month (1 - 31)   +----- hour (0 - 23) +----- min (0 - 59) </pre> For example: <code>30 22 * * 1</code> performs backups at 10:30 pm on Mondays. The default backup schedule is <code>0 2 * * *</code> (runs every day at 2 am). The timezone for backups is UTC.
<code>apicup certs set ptl CERT_NAME</code>	[ <b>CERT_FILE</b> <b>KEY_FILE</b> <b>CA_FILE</b> ] [ <b>flags</b> ] - Set the required certificates for the subsystem. Custom certificates might be set, otherwise, default certificates are automatically configured. For information on what certificates are available and how to set them see: <a href="#">Working with certificates</a> .

Command	Values/Definition
<pre>apicup subsys set ptl license- version &lt;license_typ e&gt;</pre>	<p>&lt;license_type&gt; - Specify the type of license, either <b>Production</b> or <b>Nonproduction</b>. If not specified, the default value is <b>Nonproduction</b>. Note that the license type does not impact which Portal image to install. The API Connect Kubernetes Pod annotations <b>productID</b> and <b>productName</b> reflect the selected license.</p>
<pre>apicup subsys install ptl</pre>	<p>Starts the installation of the Portal service.</p>

```
apicup subsys create ptl portal --k8s
apicup subsys set ptl extra-values-file=<path/file name>
apicup subsys set ptl portal-admin=<padmin>.<hostname>.<domainname>
apicup subsys set ptl portal-www=<portal>.<hostname>.<domainname>
apicup subsys set ptl registry=<registry-uri>
apicup subsys set ptl namespace=<namespaceName>
apicup subsys set ptl registry-secret=<registrySecret>
apicup subsys set ptl storage-class=<storageClass>
apicup subsys set ptl www-storage-size-gb=5
apicup subsys set ptl backup-storage-size-gb=5
apicup subsys set ptl db-storage-size-gb=12
apicup subsys set ptl db-logs-storage-size-gb=2
apicup subsys set ptl admin-storage-size-gb=1
apicup subsys set ptl mode=dev
apicup subsys set ptl ingress-type=route <for OpenShift environments only>
```

```
// OPTIONAL: Set the backup parameters
apicup subsys set ptl site-backup-host=<hostname>
apicup subsys set ptl site-backup-port=<22>
apicup subsys set ptl site-backup-auth-user=<username>
apicup subsys set ptl site-backup-auth-pass=<password>
apicup subsys set ptl site-backup-path=</site-backups>
apicup subsys set ptl site-backup-protocol=<sftp, objstore>
apicup subsys set ptl site-backup-schedule=<"0 0 * * *">

// OPTIONAL: Set the certificates
apicup certs set ptl CERT_NAME=[CERT_FILE KEY_FILE CA_FILE] [flags]
```

```
// OPTIONAL: Write the plan to an output directory to inspect myProject/install-plan prior to installation
apicup subsys install ptl --out=ptl-install-plan
// Start the installation from the output directory. Enter the full path to the output directory.
apicup subsys install ptl --plan-dir=./myProject/ptl-install-plan
```

```
// If output file is not used, enter the following command to start the installation
apicup subsys install ptl
```

The Portal endpoints correspond to the values entered when configuring a Portal service in the Cloud Manager. See [Registering a Portal service](#).

- `apicup subsys set ptl portal-admin` - This is the Management Endpoint that is defined in Cloud Manager, which is used for communicating with API Manager.
- `apicup subsys set ptl portal-www` - This is the Portal Website URL defined in Cloud Manager. It determines the URL for the site that is created for each Catalog. It is used for public access to the Portal from a browser.

2. After running the `apicup subsys install` command, run the `apicup subsys`

`get <portal_subsys_name> --validate` command to validate the installation. You must correct any errors that are indicated by the `--validate` command before continuing. The following example output shows how an error in the installation is displayed:

#### Kubernetes settings

=====

Name	Value	
extra-values-file		✓
ingress-type	ingress	✓
mode	dev	✓
namespace	default	✓
registry	apic-dev-docker-local.artifactory.devops.com	✓
registry-secret	apiconnect-image-pull-secret	✓
storage-class	-	✓

#### Subsystem settings

=====

Name	Value	
admin-storage-size-gb	2	✓
backup-storage-size-gb	5	✓
db-logs-storage-size-gb	2	✓
db-storage-size-gb	1	✗ db-storage-size-gb must be 12 or greater
site-backup-auth-pass	mypass	✓
site-backup-auth-user	myuser	✓
site-backup-host	1.2.3.4	✓
site-backup-path	/home/fvtuser/site-backups	✓
site-backup-port	22	✓
site-backup-schedule	0 2 * * *	✓
www-storage-size-gb	5	✓

#### Endpoints

=====

Name	Value
-----	-----

portal-admin	api.portal.default.minikube	✓
portal-www	portal.default.minikube	✓

## What to do next

If you've not already done so, you can continue your API Connect installation by installing the other subsystems.

- [Defining multiple portal endpoints for a Kubernetes environment](#)  
Multiple public facing endpoints (portal-www) can be defined for the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Defining multiple portal endpoints for a Kubernetes environment

Multiple public facing endpoints (portal-www) can be defined for the Developer Portal.

### About this task

You can override the single endpoint definition for portal-www (and the associated portal-www-ingress TLS certificate), in order to support multiple portal-www endpoints.

For information about the endpoints for the Portal, see [Installing the Developer Portal subsystem into a Kubernetes environment](#).

Following are the example endpoints for configuring different sites served by the same Portal service, as configured in this task:

- https://banking.example.com/loans
- https://insurance.example.com/vehicle

These unique endpoints allow portal sites to be defined on the Portal service with different host names and domains. They replace endpoints that distinguish different sites by sub paths, as shown in the following examples:

- https://www.example.com/banking/loans
- https://www.example.com/insurance/vehicle

### Procedure

1. Create TLS secrets for each portal-www endpoint by generating certificates  
Following is an example for how to generate certificates for each portal-www endpoint using openssl:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout banking-tls.key -out banking-tls.crt -subj
"/CN=banking.example.com"
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout insurance-tls.key -out insurance-tls.crt -subj
"/CN=insurance.example.com"
```

2. Store the SSL certificates in a secret.

```
kubectl create secret tls -n <portal-namespace> banking-tls --key banking-tls.key --cert banking-tls.crt
kubectl create secret tls -n <portal-namespace> insurance-tls --key insurance-tls.key --cert insurance-tls.crt
```

Replace `<portal-namespace>` with the Kubernetes namespace that was used to deploy the Portal subsystem.

3. Specify the portal-www endpoints in an extra values file.  
Create an extra values file or append to your current one. Enter the name and secret for each endpoint as an ingress setting in the extra values file. (One extra values file is allowed.) For instructions on creating an extra-values-file, see [Creating an extra values file in a Kubernetes environment](#).

```
apic-portal-www:
  ingress:
    web:
      hosts:
        - name: banking.example.com
          secret: banking-tls
        - name: insurance.example.com
          secret: insurance-tls
```

4. Configure your Portal subsystem to load the extra values file with the following command:

```
apicup subsys set <portal-subsys> extra-values-file=<full-path-to-extra-values-file>
```

5. Install the portal subsystem with the new extra values file using `apicup subsys install portal-subsys`.  
For more information on installing the Portal subsystem, see [Installing the Developer Portal subsystem into a Kubernetes environment](#).
6. If your deployment had existing Portal sites when you configured multiple endpoints, ensure that the Portal site URLs specified in the Manager UI Catalog settings page are consistent with the new endpoint URLs. Access the Catalog setting page, and review the URLs of those existing sites. Modify as appropriate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Installing the Gateway subsystem into a Kubernetes environment

Use the Install Assist utility program to install the Gateway subsystem into your Kubernetes runtime environment.

## Before you begin

Before installing the Gateway subsystem, complete the steps described in [First steps for installing API Connect: Upload files to registry](#). For instructions on using the APICUP installer, see [Tips and tricks for using APICUP](#).

## Procedure

1. Add a gateway service to your Kubernetes cluster by configuring a gateway subsystem.
  - The use of namespaces is required. Namespaces must be configured in Kubernetes before starting the installation process. We do not recommend that you use the default namespace.
  - DNS domain names must be configured for the endpoints. Domain names that are used for endpoints cannot contain the underscore "\_" character. See [Pre-installation requirements](#).
  - Resource allocation sizes are recommended minimums and will depend upon your environment.

Note: The steps in this procedure are not needed if you are using a DataPower appliance as your gateway.

2. The commands for installing the Gateway subsystem are described in the following table. Sample code is also provided.

Note: Host names and DNS entries may not be changed on a cluster after the initial installation.

Command	Values/Definition
<code>apicup init</code>	Initializes the apicup installation utility and creates the <code>apiconnect-up.yml</code> configuration file.
<code>apicup subsys create gw gateway -- k8s</code>	Describes a gateway subsystem in the Kubernetes cluster. The identifier <code>gw</code> is the name you have assigned to your gateway service. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character. The command <code>gateway</code> indicates that you are creating a gateway subsystem.
<code>apicup subsys set gw extra-values-file</code>	<code>&lt;path/file name&gt;</code> - Path to the extra values file. One extra values file is permitted. The extra values file requires a <code>.yaml</code> format. It contains the ingress annotations and other settings for the subsystem. For an example of an extra values file for each subsystem, see <a href="#">Creating an extra values file in a Kubernetes environment</a> .
<code>apicup subsys set gw api-gateway</code>	<code>gw.&lt;hostname&gt;.&lt;domainname&gt;</code> - The endpoint the gateway uses for API traffic. All endpoints have to be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Enter this endpoint as the API Invocation Endpoint when configuring a Gateway Service in Cloud Manager. Refer to the deployment overview diagram for information about endpoints and certificates: <a href="#">Deployment overview for endpoints and certificates</a> .
<code>apicup subsys set gw apic-gw-service</code>	<code>gwd.&lt;hostname&gt;.&lt;domain name&gt;</code> - The endpoint the gateway uses for network communication. All endpoints have to be resolvable by DNS. Domain names that are used for endpoints cannot contain the underscore "_" character. See <a href="#">Pre-installation requirements</a> . Enter this endpoint as the Management Endpoint when configuring a Gateway Service in Cloud Manager.
<code>apicup subsys set gw namespace</code>	<code>&lt;namespaceName&gt;</code> - Defines the namespace where the gateway service resides; every subsystem can be in a separate namespace. Valid namespaces are configured in Kubernetes prior to starting the API Connect installation
<code>apicup subsys set gw registry</code>	<code>&lt;registry-url&gt;</code> - The location where you are running the image registry (for example, Artifactory). If the registry-url setting is used, it takes precedence over <code>&lt;image-repository&gt;</code> and <code>&lt;image-tag&gt;</code> values that may have been set.
<code>apicup subsys set gw registry-secret</code>	For example, <code>&lt;registrySecret&gt;</code> - Kubernetes secret with Docker credentials to authenticate with the image registry. The value for <code>registry-secret</code> must match the name of the secret created using the <code>kubectl create secret</code> command. It contains the Docker credentials for accessing the registry.
<code>apicup subsys set gw image-repository</code>	Points to the repository containing the gateway service image. Must match the values set using the docker tag command for <code>&lt;REGISTRY_HOST&gt;/&lt;IMAGE_NAME&gt;</code> . For example, <code>MyRegistry/MyImage</code> .
<code>apicup subsys set gw image-tag</code>	Points to the image-tag in the local repository, use double-quotes, for example, <code>"2018.4.1.X.999999-release-prod"</code> Note: To install a non-production DataPower image, the image-tag value must be <code>datapower-api-gateway</code> .
<code>apicup subsys set gw monitor-image-repository</code>	Points to the repository containing the gateway service monitor pod image. The monitor pod watches for gateways going up or down, and removes stale gateway peer information from the pods that remain in the cluster. Removal of stale peer information improves the resiliency of the cluster. There is only one monitor pod per gateway service, regardless of how many gateway server pods (replicas) there are in the gateway service. Must match the values set using the docker tag command for <code>&lt;REGISTRY_HOST&gt;/&lt;IMAGE_NAME&gt;</code> . For example, <code>"MyRegistry/k8s-datapower-monitor"</code> . This command is required. If it is omitted, neither the DataPower monitor pod or the gateway pod will start.
<code>apicup subsys set gw monitor-image-tag</code>	Points to the monitor-image-tag in the local repository, use double-quotes, for example, <code>"2018.4.1.7"</code> . This command is required. If it is omitted, neither the DataPower monitor pod or the gateway pod will start.
<code>apicup subsys set gw image-pull-policy</code>	Determines how to obtain an image from the repository when a gateway pod is restarted. Ensures that the correct image is used by the gateway pod. <ul style="list-style-type: none"><li>• <b>Always</b> - Always pull a new image when a pod is restarted. Pulls the image that is referred to by the <code>image-tag</code>.</li><li>• <b>IfNotPresent</b> - If an image cannot be accessed locally, then pull a new image.</li><li>• <b>Never</b> - Never pull a new image. Image must be reloaded manually.</li></ul>
<code>apicup subsys set gw replica-count</code>	3 is a recommended minimum; actual value will depend upon your environment.
<code>apicup subsys set gw max-cpu</code>	The default is 4. The recommended minimum is 4. The actual value will depend upon your environment.

Command	Values/Definition
<code>apicup subsys set gwy max-memory-gb</code>	The default is 6GB. The recommended minimum is 6GB. The actual value will depend upon your environment.
<code>apicup subsys set gwy storage-class</code>	<code>&lt;storageClass&gt;</code> - Refers to a valid StorageClass object in your Kubernetes cluster that supports dynamic PersistentVolume provisioning. Static or manual PersistentVolume provisioning can be used, but it requires additional steps not covered in this documentation. Block storage is required. Select a block storage format of your choosing in order for API Connect components to be supported. GlusterFS is not recommended as a storage option due to severe performance degradation. For AWS, the gp2 and io1 types of Elastic Block Storage (EBS) are supported. Note that NFS is not supported.
<code>apicup subsys set gwy v5-compatibility-mode</code>	Determines which type of gateway to install, either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway. The DataPower Gateway (v5 compatible) is a v5-compatible gateway. The compatibility mode must match the type of image pointed to by the <code>image-tag</code> value. If the image contains a DataPower Gateway (v5 compatible) release, set the compatibility mode to <code>true</code> . If the image contains a DataPower API Gateway release, set the compatibility mode to <code>false</code> . <ul style="list-style-type: none"> <li><code>false</code> - Disables the <code>v5-compatibility-mode</code> when installing a DataPower API Gateway gateway</li> <li><code>true</code> - Enables the <code>v5-compatibility-mode</code> when installing a DataPower Gateway (v5 compatible) gateway</li> </ul> For more information, see <a href="#">Installing a DataPower API Gateway or a DataPower Gateway (v5 compatible)</a>
<code>apicup subsys set gwy enable-tms</code>	Determines whether to enable the OAuth Token Management System (tms). The Token Management System is required for a DataPower API Gateway. When the <code>v5-compatibility-mode</code> is set to <code>false</code> , the <code>enable-tms</code> parameter must be set to <code>true</code> . When the <code>v5-compatibility-mode</code> is set to <code>true</code> , the <code>enable-tms</code> parameter must be set to <code>false</code> . <ul style="list-style-type: none"> <li><code>true</code> - <code>enable-tms</code> will be enabled, required when installing a DataPower API Gateway</li> <li><code>false</code> - <code>enable-tms</code> will be disabled, required when installing a DataPower Gateway (v5 compatible)</li> </ul> For more information, see <a href="#">Installing a DataPower API Gateway or a DataPower Gateway (v5 compatible)</a>
<code>apicup subsys set gwy tms-peering-storage-size-gb</code>	Size of the storage volume for the Token Management System storage. The default is 10GB. The recommended minimum is 10GB. The actual value will depend upon your environment.
<code>apicup subsys set gwy enable-high-performance-peering</code>	Determines whether to enable high performance peering mode for a DataPower API Gateway. Enabling high performance peering is <b>highly recommended</b> for new deployments. <ul style="list-style-type: none"> <li><code>"true"</code> - Enable high performance peering for DataPower API Gateways.</li> <li><code>"false"</code> - Disable high performance peering for DataPower API Gateways.</li> </ul> The values are strings, not booleans, so must be enclosed in double quotes. <p>The requirement for use of this parameter is determined by the value of your <code>v5-compatibility-mode</code> parameter:</p> <ul style="list-style-type: none"> <li>When <code>v5-compatibility-mode</code> is <code>false</code>, the <code>enable-high-performance-peering</code> parameter is <b>required</b> and must be set.</li> <li>When <code>v5-compatibility-mode</code> is <code>true</code>, the <code>enable-high-performance-peering</code> parameter has no effect, and is not required.</li> </ul> For information on enabling high performance peering on existing deployments of version 2018.4.1.7 or later, see <a href="#">Enabling high performance peering for DataPower API Gateway on Kubernetes</a> .
<code>apicup subsys set gwy mode</code>	Determines whether the installation is for development, testing, and demo purposes or for a production environment. If not explicitly set, the mode will default to <code>dev</code> (development and testing) for versions 2018.4.1.4 and later. For versions 2018.4.1.3 and earlier, the default for mode is <code>standard</code> . <ul style="list-style-type: none"> <li><code>dev</code> - (Default) Use dev mode for development, testing, and demo purposes. It allows one instance of each subsystem to be created.</li> <li><code>standard</code> - Use standard mode for production environments. It allows multiple instances of each subsystem to be created.</li> </ul>
<code>apicup subsys set gwy ingress-type</code>	Must be explicitly set to <code>route</code> for an OpenShift environment. For a standard K8s environment, <code>ingress-type</code> defaults to <code>ingress</code> and there is no need to explicitly enter this parameter.
<code>apicup subsys set gwy license-version</code>	<code>&lt;license&gt;</code> - Specify the type of gateway license, one of <code>Developers</code> , <code>Production</code> , or <code>Nonproduction</code> . If not specified, the default value is <code>Developers</code> . The Gateway server image that you are installing must match the license version. Note that this requirement does not apply to the other API Connect subsystems (Management, Analytics, and Developer Portal), because they do not have separate images for each license version.
<code>apicup certs set gwy CERT_NAME</code>	<code>[CERT_FILE KEY_FILE CA_FILE] [flags]</code> - Set the required certificates for the subsystem. Custom certificates may be set, otherwise, default certificates are automatically configured. For information on what certificates are available and how to set them see: <a href="#">Working with certificates</a> .
<code>apicup subsys install</code>	<code>gwy</code> - Starts the installation of the gateway service

```

apicup subsys create gwy gateway --k8s
apicup subsys set gwy extra-values-file=<path/file name>
apicup subsys set gwy api-gateway=gw.<hostname>.<domainname>
apicup subsys set gwy apic-gw-service=gwd.<hostname>.<domainname>
apicup subsys set gwy namespace=<namespaceName>
apicup subsys set gwy registry=<registry-uri>
apicup subsys set gwy registry-secret=<registrySecret>
apicup subsys set gwy image-repository=<ibmcom/datapower>
apicup subsys set gwy image-tag="release_name"

apicup subsys set gwy monitor-image-repository="ibmcom/k8s-datapower-monitor"

apicup subsys set gwy monitor-image-tag="2018.4.1.7"
apicup subsys set gwy image-pull-policy=Always
apicup subsys set gwy replica-count=3
apicup subsys set gwy max-cpu=4
apicup subsys set gwy max-memory-gb=6
apicup subsys set gwy storage-class=<storageClass>
apicup subsys set gwy v5-compatibility-mode=false
apicup subsys set gwy enable-high-performance-peering=true
apicup subsys set gwy enable-tms=true
apicup subsys set gwy tms-peering-storage-size-gb=10
apicup subsys set gwy mode=dev

```

```

apicup subsys set gwy ingress-type=route <for OpenShift environments only>
apicup subsys set gwy license-version=<license>

//Set the certificates
apicup certs set gwy CERT_NAME=[CERT_FILE KEY_FILE CA_FILE] [flags]

// OPTIONAL: Write the plan to an output directory to inspect myProject/install-plan prior to installation
apicup subsys install gwy --out=gwy-install-plan
// Start the installation from the output directory. Enter the full path to the output directory.
apicup subsys install gwy --plan-dir=./myProject/gwy-install-plan

//If output file is not used, enter the following command to start the installation
apicup subsys install gwy

```

### Installing a DataPower API Gateway or a DataPower Gateway (v5 compatible)

The values for the `v5-compatibility-mode` and `enable-tms` parameters determine whether you are installing a DataPower API Gateway or a DataPower Gateway (v5 compatible). The following table shows the required settings for installing both types of gateways:

	<code>v5-compatibility-mode</code>	<code>enable-tms</code>
DataPower API Gateway	false	true
DataPower Gateway (v5 compatible)	true	false

- An alternate deployment scenario is to use a physical DataPower appliance for the gateway. If you are using an appliance for the gateway, do not run any of the APICUP commands described on this page, since you do not need to configure a gateway subsystem in your Kubernetes cluster when using an appliance. For an appliance-based gateway, you must configure two endpoints in DataPower to be used as the Gateway Service Management Endpoint and the API invocation Endpoint. Enter these endpoints in the Configure Gateway Service screen in Cloud Manager. See [Configuring a DataPower Gateway on an appliance](#) for instructions for configuring a DataPower appliance.
- After running the `apicup subsys install` command, run the `apicup subsys get <gateway_subsys_name> --validate` to validate the installation. You must correct any errors indicate by `--validate` prior to continuing. Following is an example for the `--validate` output:

```

Kubernetes settings
=====

Name                               Value
----                               -
extra-values-file                   ✓
ingress-type                        ingress ✓
mode                                standard ✓
namespace                           prod ✓
registry                            apic-dev-docker-local.artifactory.swg-devops.com/apicup-imgs/2018.4.1-550 ✓
registry-secret                     apiconnect-image-pull-secret ✓
storage-class                       rook-block ✓

Subsystem settings
=====

Name                               Value
----                               -
enable-tms                          true ✓
image-pull-policy                    Always ✓
image-repository                    ibmcom/datapower ✓
image-tag                            2018.4.1.x-999999-release-prod ✓
monitor-image-repository            ibmcom/k8-datapower-monitor ✓
monitor-image-tag                   2018.4.1.7 ✓
enable-high-performance-peering     "true" ✓
max-cpu                              4 ✓
max-memory-gb                       6 ✓
replica-count                       3 ✓
tms-peering-storage-size-gb        10 ✓
v5-compatibility-mode              false ✓

Endpoints
=====

Name                               Value
----                               -
api-gateway                         gw.1.23.123.45.sample.io ✓
apic-gw-service                     gwd.1.23.123.45.sample.io ✓

```

## What to do next

Continue your API Connect installation in a Kubernetes runtime environment by installing the other subsystems.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying pods to specific worker nodes in a multi-node cluster

Use node labels to deploy Management, Analytics, and Portal pods to specific worker nodes in a multi-node cluster.

## About this task

A label is a key and value that is attached to an object as an annotation. For a pod to use a NodeSelector, a worker node must exist with the specified label. If no worker node exists with the required label, then the pod is not deployed and is left in the pending state.

## Procedure

1. Label the worker nodes where you want to deploy the subsystem pods:
  - a. For each worker node that you want to label, run the following command:

```
kubectl label nodes <node-name> <Label-Key>=<Label-Value>
```

Use the following <Label-Key>=<Label-Value> pairs to label a node for each subsystem:

- Management: `node-apim=apim`  
For example, to add the `apim` label to the node `worker-1.ibm.com`:  

```
kubectl label nodes worker-1.ibm.com node-apim=apim
```
- Portal: `node-ptl=ptl`  
For example, to add the `ptl` label to the node `worker-3.ibm.com`:  

```
kubectl label nodes worker-3.ibm.com node-ptl=ptl
```
- Analytics: `node-analyt=analyt`  
For example, to add the `analyt` label to the node `worker-2.ibm.com`:  

```
kubectl label nodes worker-2.ibm.com node-analyt=analyt
```

- b. Run the following command to verify that the labels were applied correctly:

```
kubectl get nodes --show-labels
```

2. Deploy the Management subsystem to the labeled node:
  - a. Create the `myapim-extra-value-files.yaml` file with the following contents:

```
cassandra:
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
        - preference:
            matchExpressions:
              - key: beta.kubernetes.io/arch
                operator: In
                values:
                  - amd64
            weight: 3
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: node-apim
                    operator: In
                    values:
                      - apim
            podAntiAffinity:
              preferredDuringSchedulingIgnoredDuringExecution:
                - podAffinityTerm:
                    labelSelector:
                      matchLabels:
                        app: cassandra
                    topologyKey: kubernetes.io/hostname
                weight: 1
  analytics-proxy:
    nodeSelector:
      node-apim: apim
  apim:
    nodeSelector:
      node-apim: apim
  client-downloads-server:
    nodeSelector:
      node-apim: apim
  juhu:
    nodeSelector:
      node-apim: apim
  ldap:
    nodeSelector:
      node-apim: apim
  lur:
    nodeSelector:
      node-apim: apim
  ui:
    nodeSelector:
      node-apim: apim
  nodeSelector:
    node-apim: apim
```

- b. Run the following command to set the extra-values-file for the Management subsystem:

```
./apicup subsys set mgmt extra-values-file myapim-extra-value-files.yaml
```

- c. Run the following command to install the Management subsystem:



```
./apicup subsys install mgmt
```

d. Run the following command to verify that the Management pods are running in only the worker-1.ibm.com node that you labeled:

```
kubectl get po -o wide
```

3. Deploy the Portal subsystem to the labeled node:

a. Create the myportal-extra-value-files.yaml file with the following contents:

```
apic-portal-db:
  nodeSelector:
    node-ptl: ptl
apic-portal-www:
  nodeSelector:
    node-ptl: ptl
```

b. Run the following command to set the extra-values-file for the Portal subsystem:

```
./apicup subsys set ptl extra-values-file myportal-extra-value-files.yaml
```

c. Run the following command to install the Portal subsystem:

```
./apicup subsys install ptl
```

d. Run the following command to verify that the Portal pods are running in only the worker-3.ibm.com node that you labeled:

```
kubectl get po -o wide
```

4. Deploy the Analytics subsystem to the labeled node:

a. Create the myanalytics-extra-value-files.yaml file with the following contents:

```
apic-analytics-client:
  nodeSelector:
    node-analyt: anlyt
apic-analytics-cronjobs:
  nodeSelector:
    node-analyt: anlyt
apic-analytics-ingestion:
  nodeSelector:
    node-analyt: anlyt
apic-analytics-mtls:
  nodeSelector:
    node-analyt: anlyt
apic-analytics-operator:
  nodeSelector:
    node-analyt: anlyt
apic-analytics-storage:
  nodeSelector:
    node-analyt: anlyt
```

b. Run the following command to set the extra-values-file for the Analytics subsystem:

```
./apicup subsys set analytics extra-values-file myanalytics-extra-value-files.yaml
```

c. Run the following command to install the Analytics subsystem:

```
./apicup subsys install analytics
```

d. Run the following command to verify that the analytics pods are running in only the worker-2.ibm.com node that you labeled:

```
kubectl get po -o wide
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Manually creating a CustomResourceDefinition in a Kubernetes environment

Describes the steps for manually creating the CustomResourceDefinitions which are required for the management subsystem. If the `create-crd` command is set to false during installation, the CRDs must be manually created by the Kubernetes administrator.

### Before you begin

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated. These instructions assume you have a working Kubernetes environment and understand how to manage Kubernetes.

Kubernetes is a platform for automated deployment, scaling, and operation of application containers across clusters of hosts, providing container-centric infrastructure. For more information, see <https://kubernetes.io>.

### About this task

If the `apicup subsys set mgmt create-crd` command is set to `false` in the Install Assist script (perhaps to disallow cluster-level access during installation), then the CRDs must be created manually prior to deploying API Connect in a Kubernetes environment.



## Procedure

1. Ensure you have not already installed the CRDs by executing the following command: `kubectl get crd`
2. Download the `cassandra-crds_lts_v2018.4.1.20.yaml` CRD from IBM Fix Central at the following link: [http://www.ibm.com/support/fixcentral/quickorder?product=ibm%2FWebSphere%2FIBM+API+Connect&fixids=cassandra-crds\\_lts\\_v2018.4.1.20.yaml&source=SAR](http://www.ibm.com/support/fixcentral/quickorder?product=ibm%2FWebSphere%2FIBM+API+Connect&fixids=cassandra-crds_lts_v2018.4.1.20.yaml&source=SAR)
3. Execute `kubectl create -f <CustomResourceDefinition.yaml>` (use the same file name as the one created to define the CRDs). This installs the two required CRDs, `cassandraclusters.apic.ibm.com` and `cassandraclusterbackups.apic.ibm.com`.
4. Verify the installation by entering `kubectl get crd`. You should receive results similar to the following:

NAME	AGE
<code>cassandraclusterbackups.apic.ibm.com</code>	79d
<code>cassandraclusters.apic.ibm.com</code>	79d

5. After installing the CRDs manually, install the management subsystem following the instructions at [Installing the Management subsystem into a Kubernetes environment](#). Set the `apicup subsys set mgmt create-crd` command to `false`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating an extra values file in a Kubernetes environment

Describes the steps for creating an extra values file for configuration parameters that are not set by Install Assist, such as ingress annotations and resource limits. An example `.yaml` file is provided with entries for each subsystem.

### Before you begin

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated. These instructions assume you have a working Kubernetes environment and understand how to manage Kubernetes.

Kubernetes is a platform for automated deployment, scaling, and operation of application containers across clusters of hosts, providing container-centric infrastructure. For more information, see <https://kubernetes.io>.

### About this task

The APICUP installer exposes the most important parameters for configuring a subsystem using the `subsys set` commands. The extra values file is a `.yaml` file used to set additional configuration parameters contained in the Helm chart, further to those parameters that are set using APICUP. The `.yaml` format is required. One extra values file per subsystem is allowed. The sample file is named, for example, `myExtraValues.yaml`.

Syntax:

```
apicup subsys set <subsystem_name> extra-values-file <name_of_extra_values_file>
```

The `<name_of_extra_values_file>` parameter can include a directory path.

Example for `mgmt` subsystem:

```
apicup subsys set mgmt extra-values-file myExtraValues.yaml
```

## Procedure

- **Logging level** - If you want reduced logging for the `apim`, `lur`, and `ldap` pods, set a `DEBUG` environment variable using the extra-values file. The value for `modes` must match the mode you are using (`demo` or `standard`). In the following example, `modes` is set to `standard`:

```
apim:
  modes:
    standard:
      env:
        DEBUG: "audit,bhendi:error,bhendi:probe,bhendi:flags,apim:server,apim:error"
lur:
  modes:
    standard:
      env:
        DEBUG: "audit,bhendi:error,bhendi:probe,bhendi:flags,lur:server,lur:error"
ldap:
  modes:
    standard:
      env:
        DEBUG: "audit,bhendi:error,bhendi:probe,bhendi:flags,ldap:server,ldap:error"
```

- **Optimize subsystem sync time** - You can use the extra-values file to set the pagination limit for records that are fetched by a database query. Add the flag `VELOX_CASSANDRA_LIMIT` to the extra-values file. The recommended value is `500`. Use this setting to reduce the time needed to sync subsystems. This setting is available in Version 2018.4.1.6.iFix3, and in Version 2018.4.1.7 or later. For example:

```
apim:
  modes:
    standard:
      env:
        VELOX_CASSANDRA_LIMIT: "500"
```

```

lur:
  modes:
    standard:
      env:
        VELOX_CASSANDRA_LIMIT: "500"

```

- **Management subsystem** - The following example contains entries for the extra values file to set Kubernetes ingress annotations and resource limits for the Management subsystem. By default, the APICUP installer does not specify limits for most resources. If you are using quota enforcement when sharing Kubernetes cluster resources, resource limits may be required. Adjust the values given in this example to support your resource constraints, quotas, and expected load/capacity requirements:

Note: Setting of resource requests and limits on OpenShift is not supported for the API Connect Analytics subsystem.

```

global:
  ingress:
    # cloud-admin-ui endpoint
    cm:
      annotations:
        kubernetes.io/ingress.class: "nginx"
    # api-manager-ui endpoint
    apim:
      annotations:
        kubernetes.io/ingress.class: "nginx"
    # platform-api endpoint
    platformAPI:
      annotations:
        kubernetes.io/ingress.class: "nginx"
    # consumer-api endpoint
    consumerAPI:
      annotations:
        kubernetes.io/ingress.class: "nginx"

```

```

analytics-proxy:
  resources:
    limits:
      cpu: 100m
      memory: 128Mi

```

```

apim:
  resources:
    limits:
      cpu: 1000m
      memory: 2Gi

```

```

cassandra:
  resources:
    limits:
      cpu: 2100m
      memory: 9Gi

```

```

client-downloads-server:
  resources:
    requests:
      cpu: 10m
    limits:
      cpu: 100m
      memory: 32Mi

```

```

juhu:
  resources:
    limits:
      cpu: 200m
      memory: 512Mi

```

```

ldap:
  resources:
    limits:
      cpu: 100m
      memory: 256Mi

```

```

lur:
  resources:
    limits:
      cpu: 200m
      memory: 256Mi

```

```

ui:
  resources:
    limits:
      cpu: 200m
      memory: 64Mi

```

- **Gateway subsystem** - Refer to the following example that contains entries for the extra values file for setting Kubernetes ingress annotations, the license version (applies to v2018.4.1.4 - v2018.4.1.6 only), and an optional *customDatapowerConfig* setting to point to the *ConfigMap* for baked-in policies for the Gateway subsystem.

Note: The `licenseVersion` setting shown in the `datapower` section applies to versions v2018.4.1.4 - v2018.4.1.6 only. For v2018.4.1.7 and later versions, the license version is set using APICUP when installing the Gateway subsystem. Delete the license version setting from the extra-values-file when upgrading to v2018.4.1.7 and configure the license version using APICUP.

```

ingress:
  # api-gateway endpoint
  gateway:
    annotations:
      kubernetes.io/ingress.class: "nginx"
  # apic-gw-service endpoint
  gwd:

```

```

    annotations:
      kubernetes.io/ingress.class: "nginx"

datapower:
  licenseVersion: "version - Note: Remove this setting if using v2018.4.7 or later"
  customDatapowerConfig: "custom-dp-config"

```

- **Portal subsystem** - The following example is an extra values file that sets Kubernetes ingress annotations for the Portal subsystem and shows an example for defining multiple Portal endpoints:

```

global:
  ingress:
    # portal director endpoint. used for apim -> portal comms
    portal:
      annotations:
        kubernetes.io/ingress.class: "nginx"
    # portal web site endpoint. used to access the portal web site(s)
    web:
      annotations:
        kubernetes.io/ingress.class: "nginx"

apic-portal-www:
  ingress:
    web:
      hosts:
        - name: banking.example.com
          secret: banking-tls
        - name: insurance.example.com
          secret: insurance-tls

```

- **Analytics subsystem** - The following example is an extra values file that sets Kubernetes ingress annotations for the Analytics subsystem:

```

global:
  ingress:
    client:
      annotations:
        kubernetes.io/ingress.class: "nginx"
    ingestion:
      annotations:
        kubernetes.io/ingress.class: "nginx"

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring user-defined policies on the API Gateway in a Kubernetes deployment

For a Kubernetes deployment that uses the DataPower API Gateway, user-defined Policies are externally configured. To distribute the user-defined policies on the DataPower API Gateway, you create a Kubernetes ConfigMap that is installed using the extra values file. The ConfigMap ensures the policies are available to the Management server.

### About this task

When to use the instructions in this task

Use the instructions in this topic for distributing user-defined policies when **both** of the following two conditions are met:

- Your Gateway services is deployed on Kubernetes. The Management service is also usually deployed on Kubernetes.
- Your topology uses the DataPower API Gateway, not the DataPower Gateway (v5 Compatible).

For an explanation of gateway types, see [API Connect gateway types](#).

For a Kubernetes deployment with the DataPower® API Gateway, you can configure user-defined policies on the Gateway subsystem for distribution to API Manager as part of the deployment process, or you can configure user-defined policies at any time, on any platform, after the deployment process is complete. With the DataPower Gateway (v5 compatible) you can configure user-defined policies only after the deployment process is complete. For details of how to configure user-defined policies after deployment with either gateway type, see [Authoring policies](#).

To configure user-defined policies on the Gateway subsystem for distribution to API Manager during deployment, you create a Kubernetes ConfigMap containing the code for the policy and then add an entry in the extra values file to point to the ConfigMap. When the APICUP installation command is executed, the externally-configured policies will be loaded into the DataPower configuration at start-up and then shared with the API Manager, which makes them available in the assembly palette. A Kubernetes ConfigMap ensures that the externally configured user-defined policies are available when the Gateway is started, across upgrades and restarts of the Gateway, and when scaling the number of pods in a cluster. The policies are uploaded directly to the `apic-gw-service` object, which was installed using the APICUP installer. See [Installing the Gateway subsystem in a Kubernetes environment](#) for instructions on installing the Gateway, including how to install the extra values file.

In cases where non-standard charts or other types of virtual or physical deployments, these policies must be mounted or placed on all the members of the Gateway service.

For topologies using a virtual or physical gateway with the DataPower API Gateway

If your topology uses a virtual (OVA) or physical DataPower Gateway and the DataPower API Gateway, then refer to the DataPower documentation for configuring an assembly function available here: [Configuring an assembly function](#).

## Procedure

- Create a configuration files containing the user-defined policies. In the following example, a configuration file named `my-echo-policy.cfg` defines a policy that sets the content of a response body from the value of a parameter.

1. Define the policy in a file named, for example, `my-echo-policy.cfg`.

The following DataPower configuration commands create an assembly-function object that represents the user-defined policy that will be advertised to the Management server in the ConfigMap. This example writes the value of the `msg` parameter provided as input to the user-defined policy to the `message.body` that is returned on the API invocation.

```
assembly-setvar "my-echo-policy_1.0.0_assembly-setvar"
  variable
    action set
      name "message.body"
      type string
      value "$(local.parameter.msg)"
    exit
  variable
    action set
      name "message.headers.content-type"
      type string
      value "text/plain"
    exit
  exit
exit

api-rule "my-echo-policy_1.0.0_api-rule"
  action my-echo-policy_1.0.0_assembly-setvar
exit

assembly "my-echo-policy_1.0.0_assembly"
  rule my-echo-policy_1.0.0_api-rule
exit

assembly-function "my-echo-policy_1.0.0"
  summary "my-echo-policy_1.0.0"
  title "My Echo Policy"
  parameter
    name "msg"
    description "The message string"
    value-type string
  exit
  assembly my-echo-policy_1.0.0_assembly
exit

apic-gw-service
  user-defined-policies my-echo-policy_1.0.0
exit
```

- Add the configuration file to the ConfigMap.
  1. To make the `.cfg` files available to the Gateway subsystem, create a Kubernetes ConfigMap named, for example, `custom-dp-config`:

```
kubectl create configmap custom-dp-config --from-file=my-echo-policy.cfg
```

- Optionally, you can add your configuration files directly to the Helm chart.
  1. You create a directory for `dynamic-gateway-service/config` and place your configuration into a compressed tar file.
  2. Use `apicup` commands as shown below to place `<your_config>` into the Helm chart, so that it can be referenced by the `values.yaml`.  
For example:

Figure 1. Using `apicup` to install configuration into the Helm Chart

```
mkdir -p dynamic-gateway-service/config
cp <your_config> dynamic-gateway-service/config/
apicup subsys install gateway --out <dir>
gunzip <dir>/helm/dynamic-gateway-service-X.X.XX.tar.gz
tar -rf <dir>/helm/dynamic-gateway-service-X.X.XX.tar dynamic-gateway-service
gzip <dir>/helm/dynamic-gateway-service-X.X.XX.tar
mv <dir>/helm/dynamic-gateway-service-X.X.XX.tar.gz <dir>/helm/dynamic-gateway-service-X.X.XX.tar.gz
```

3. In the extra values file, specify:

```
datapower:
  additionalConfig:
  - domain: <some-domain>
    config: config/<your config>
```

4. Run the following commands to install the gateway with the specified configuration:

```
apicup subsys set gateway extra-values-file <extra values file>
apicup subsys install gateway --plan-dir <dir>
```

5. If you are updating an existing install, you also need to restart the pod by running the following command:

```
kubectl delete pod/<gateway_pod_name> -n <namespace>
```

The new gateway pod is then automatically created with the new configuration.

- Optionally, you can add local files directly to the Helm charts.  
Local files can be added into the gateway deployment (Helm chart) by use of the `datapower.additionalLocalTar` value. This value is a path to a tar file which contains all the files you wish to add. This tar file should be a well formatted DataPower `local`: directory where files intended for the `default` domain are on the top level and all files intended for a different domain are in a subdirectory named for that domain.

Use the same task flow as shown in [Figure 1](#), but for the local path use `local.tar.gz`.

- Optionally, you can add certificates to the ConfigMap.

Certificates and other crypto files can be added to the `cert:` directory by use of the `datapower.additionalCerts` value. This value takes the form of a list of domain-secret pairs. The secrets are Kubernetes secrets which contain the crypto files you wish to use. To create the secrets from an existing crypto key-cert pair:

1. Create the secrets from a specified file:

```
kubectl create secret generic my-secret --from-file=/path/to/key.pem --from-file=/path/to/cert.pem
```

2. Reference the secret in the extra values file by specifying:

```
datapower:
  additionalCerts:
  - domain: "default"
    secret: "some-default-cert-secret"
  - domain: "apiconnect"
    secret: "some-apiconnect-cert-secret"
```

- Install the ConfigMap containing the policies using the extra values file and the APICUP installer.
  1. In order to configure the Gateway to use the custom policies in the `custom-dp-config` ConfigMap, set the `customDatapowerConfig` value to point to the ConfigMap in the extra values file:  
For information on the extra values file, see [Creating an extra values file in a Kubernetes environment](#).

```
datapower:
  customDatapowerConfig: "custom-dp-config"
```

2. Point to the extra values file from the Gateway subsystem by entering `apicup subsys set gwy extra-values-file path/file`.
  3. Execute the `apicup subsys install gwy` command to install the Gateway with the extra values file containing the ConfigMap entry.
- Use the externally-configured user-defined policy in an API.
    1. To use the externally-configured policy in the example from an API, create an assembly that resembles the following:

```
assembly:
  execute:
  - my-echo-policy:
    title: Echo msg
    version: 1.0.0
    msg: ${request.parameters.msg}
```

2. A request against that API should have `?msg=<text>` as a query parameter, and the response body should be equal to `<text>`, which is the value of the `msg` query parameter.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing the toolkit

You can install the toolkit that provides CLI commands, and the API Designer user interface, for IBM® API Connect.

### About this task

The toolkit is provided as executable files, so no actual installation is necessary, you just need to download the required compressed file and extract the contents.

There are two toolkit options available:

- CLI: provides a command line environment for working with IBM API Connect.
- CLI + LoopBack + Designer: provides a command line environment for working with IBM API Connect, including LoopBack® support, and the API Designer user interface.

To install the toolkit, download the compressed file that is appropriate for your chosen toolkit option and platform, then extract the contents to a chosen location on your local machine. The compressed file contains an executable file for running CLI commands and, if you choose the CLI + LoopBack + Designer option, an executable file for launching the API Designer user interface.

You can download the toolkit compressed file in either of the following ways:

- From IBM Fix Central.
- From either the Cloud Manager or API Manager user interface.

The following table identifies the name of the compressed file that you need to download, depending on your chosen toolkit option and platform:  
Table 1. Toolkit file names, by option and platform

Toolkit option	Mac OS X	Linux®	Windows
CLI	toolkit-mac.zip	toolkit-linux.tgz	toolkit-windows.zip
CLI + LoopBack + Designer	toolkit-loopback-designer-mac.zip	toolkit-loopback-designer-linux.tgz	toolkit-loopback-designer-windows.zip

---

## Procedure

To install and run the toolkit, complete the following steps:

1. Download the toolkit compressed file.
  - To download the toolkit from IBM Fix Central, complete the following steps:
    - Open the [IBM Fix Central site](#) in your browser.

- In the Product selector field, enter `API Connect`, then select IBM API Connect from the drop down list.
  - Select your installed 2018.x.y version from the Installed Version list, then click Continue. If you do not know your installed IBM API Connect version, contact your administrator.
  - In the Text field, enter `toolkit`, then click Continue.
  - Select the required file, as identified in [Table 1](#).  
Note: When you download from IBM Fix Central, the release number is appended to the file name.
  - Click Continue, then follow the instructions to complete the download operation.
  - To download the toolkit from either the Cloud Manager or API Manager user interface, complete the following steps:
    - Cloud Manager or API Manager user interface.
    - Select Help ( ? ) in the navigation.
    - Select Install API Connect CLI & API Designer.
    - Select CLI or CLI + LoopBack + Designer according to your preferred option.
    - Select your platform to download the toolkit compressed file.
    - Close the Install API Connect CLI & API Designer window.
2. Extract the contents of the toolkit compressed file to a folder of your choice.  
The contents of the file depend on the your chosen toolkit option and platform, as follows:

Table 2. Toolkit compressed file contents, by option and platform

Toolkit option	Mac OS X	Linux	Windows
CLI	apic-slim	apic-slim	apic-slim.exe
CLI + LoopBack + Designer	apic api_designer-mac.zip: contains the API Designer user interface application.	apic api_designer-linux	apic.exe api_designer-win.exe

The apic-slim or apic-slim.exe file is the CLI for IBM API Connect.

The apic or apic.exe file is the CLI for IBM API Connect including LoopBack support.

Tip: If you are using the CLI option, then if you rename the apic-slim file to apic, or the apic-slim.exe file to apic.exe, you can run the CLI commands exactly as documented, copy and paste sample commands from the documentation, and use any command scripts as-is if you later move to the CLI + LoopBack + Designer option.

The `api_designer-platform` file is the API Designer user interface application for the specified platform.

3. Run the CLI.
- For the Mac OS X or Linux platforms, complete the following steps:
    - Open a terminal instance and navigate to the folder where you extracted the contents of the toolkit compressed file.
    - Make the CLI file an executable file by entering the following command:

```
chmod +x download_name
```

Where `download_name` is the name of the toolkit file that you downloaded, either apic or apic-slim.

- Run CLI commands as follows:

```
./apic command_name_and_parameters
```

or

```
./apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

- For the Windows platform, complete the following steps:
  - Open a terminal window and navigate to the folder where you extracted the contents of the toolkit compressed file.
  - Run CLI commands as follows:

```
apic command_name_and_parameters
```

or

```
apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

Tip: Add the folder location of your CLI file to your PATH variable so that you can run CLI commands from anywhere in your file system.

4. Launch the API Designer user interface by running the application from the location to which you extracted the contents of the toolkit compressed file.

Note:

- To uninstall the API Designer application on a Windows platform with a non Administrator account, complete the following steps:
  - In Windows File Explorer, navigate to the `USER_HOME\AppData\Local\Programs\api-designer` folder.
  - Run the **Uninstall API Designer application** application. Do **not** use the **Add or remove programs** window.
- To uninstall the API Designer application on a Windows platform with an Administrator account, you can either run the **Uninstall API Designer application** application, or you can use the **Add or remove programs** window.

## Results

The IBM API Connect toolkit CLI and, if selected, the API Designer user interface application are installed on your local system.

For information on using the API Designer user interface, see [Developing your APIs and applications](#).

For information on using the toolkit CLI, see [Using the developer toolkit command-line tool](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Upgrading API Connect in a Kubernetes environment

Upgrades are performed from the same project directory used for the initial installation.

## About this task

---

Review the requirements before starting an upgrade. Follow the procedures in the following links to complete an upgrade.

Note: Version 2018.4.1.7 introduced a new monitor pod for gateway. If you are upgrading the gateway from Version 2018.4.1.6 or earlier to Version 2018.4.1.7 or later, you must use separate instructions to install a new monitor pod during the upgrade. See the following links.

- [Requirements for upgrading](#)  
Before upgrading API Connect on Kubernetes, ensure that your deployment meets all the upgrade requirements.
- [Upgrading API Connect subsystems](#)  
Complete the following steps to upgrade API Connect subsystems.
- [Adding a monitor pod during gateway upgrade](#)  
Version 2018.4.1.7 introduced a new Gateway monitor pod that is required during an upgrade.
- [Verifying Portal upgrade](#)  
You can monitor the progress of existing Portal sites being upgraded to the new version.
- [Troubleshooting the upgrade](#)  
Troubleshoot problems with the upgrade.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Requirements for upgrading

Before upgrading API Connect on Kubernetes, ensure that your deployment meets all the upgrade requirements.

The instructions for upgrading apply to the *latest* Fix Pack version. Ensure that you are upgrading to the latest Fix Pack version. For information on the available Fix Packs, see [What's New in the latest release](#).

Note: For more information if you are moving to v2018.4.1.10, see the Tech note at [IBM Support](#).  
See the requirements for the version you are upgrading from:

- [Requirements for upgrading from v2018.4.1.8 or later](#)  
Review the requirements for upgrading from Version 2018.4.1.8 or later.
- [Requirements for upgrading from v2018.4.1.5, v2018.4.1.6, or v2018.4.1.7](#)  
Review the requirements for upgrading from Version 2018.4.1.5, Version 2018.4.1.6, or Version 2018.4.1.7
- [Requirements for upgrading from Version 2018.4.1.4 or earlier](#)  
Review the requirements for upgrading from Version 2018.4.1.4 or earlier on Kubernetes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Requirements for upgrading from v2018.4.1.8 or later

Review the requirements for upgrading from Version 2018.4.1.8 or later.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- Verify the API Connect deployment is fully operational. See [Checking cluster health on Kubernetes](#).  
When upgrading to v2018.4.1.10 or later, this step is optional because a health check will be run automatically as part of the `apicup subsys install` command. Note that you might still want to run the health check when preparing for the upgrade, to ensure the health of the backup image you must create prior to running `apicup subsys install`.
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.  
  
For details on creating a backup, see [Backing up the management database in a Kubernetes environment](#), [Backing up and restoring the Developer Portal](#) and [Backing up and restoring the analytics database](#).
- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a Kubernetes deployment](#).
- Helm version 2.8.2 or higher is required when upgrading API Connect in a Kubernetes environment.

- The same distribution file for each API Connect subsystem is used for either upgrades or fresh installations. For an example list of the files for Kubernetes upgrades, see [First steps for installing API Connect: Upload files to registry](#).
- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading from v2018.4.1.5, v2018.4.1.6, or v2018.4.1.7

Review the requirements for upgrading from Version 2018.4.1.5, Version 2018.4.1.6, or Version 2018.4.1.7

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- Verify the API Connect deployment is fully operational.
  - For Version 2018.4.1.6 or later, see [Checking cluster health on Kubernetes](#).  
When upgrading *from* v2018.4.1.6 or later *to* v2018.4.1.10 or later, this step is optional because a health check will be run automatically as part of the `apicup subsys install` command. Note that you might still want to run the health check when preparing for the upgrade, to ensure the health of the backup image you must create prior to running `apicup subsys install`.
  - For Version 2018.4.1.5, see [Determining status of a cluster on Kubernetes](#).
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management database in a Kubernetes environment](#), [Backing up and restoring the Developer Portal](#) and [Backing up and restoring the analytics database](#).

- The upgrade of the management server *from* Version 2018.4.1.7 or earlier may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data, such as greater than 10 GB.

To reduce the amount of time required for the upgrade of the management database, use the `apicops` command to truncate the subscriber event table and remove unused snapshots:

```
apicops subscriber-queues:clear
apicops snapshots:clean-up
```

Use the latest release of the `apicops` command. Download it from <https://github.com/ibm-apiconnect/apicops/releases>.

- When upgrading an OpenShift deployment to API Connect v4.1.6, the OpenShift router may reject the renamed route for the analytics proxy. This is caused by a bug in OpenShift. If the analytics proxy routes are rejected with the error `HostAlreadyClaimed`, restart the OpenShift router using the following command:

```
oc delete po -n default <router-x-xxxxx>
```

where `router-x-xxxxx` is the name of the router.

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a Kubernetes deployment](#).
- Helm version 2.8.2 or higher is required when upgrading API Connect in a Kubernetes environment.
- The same distribution file for each API Connect subsystem is used for either upgrades or fresh installations. For an example list of the files for Kubernetes upgrades, see [First steps for installing API Connect: Upload files to registry](#).
- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.



- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading from Version 2018.4.1.4 or earlier

Review the requirements for upgrading from Version 2018.4.1.4 or earlier on Kubernetes.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- Verify the API Connect deployment is fully operational. For Version 2018.4.1.5 or earlier, see [Determining status of a cluster on Kubernetes](#).
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management database in a Kubernetes environment](#), [Backing up and restoring the Developer Portal](#) and [Backing up and restoring the analytics database](#).

- The upgrade of the management server from Version 2018.4.1.7 or earlier may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data, such as greater than 10 GB.

To reduce the amount of time required for the upgrade of the management database, use the `apicops` command to truncate the subscriber event table and remove unused snapshots:

```
apicops subscriber-queues:clear
apicops snapshots:clean-up
```

Use the latest release of the `apicops` command. Download it from <https://github.com/ibm-apiconnect/apicops/releases>.

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a Kubernetes deployment](#).
- Helm version 2.8.2 or higher is required when upgrading API Connect in a Kubernetes environment.
- The same distribution file for each API Connect subsystem is used for either upgrades or fresh installations. For an example list of the files for Kubernetes upgrades, see [First steps for installing API Connect: Upload files to registry](#).
- For v2018.4.1.4 and later, the default setting for `mode` is `dev`. For v2018.4.1.3 and earlier the default setting for `mode` is `standard`. Thus, if your deployment was running in `standard` mode prior to performing the upgrade, you will need to explicitly set the mode to `standard` for each subsystem using the following command in the APICUP installer: `apicup subsys set SUBSYS mode=standard`. Note that this requirement primarily affects deployments of v2018.4.1.3 or earlier that accepted the default mode of `standard`. If you are upgrading from a v2018.4.1.3 or earlier deployment, on which you accepted the default mode of `standard`, be aware of the change in default behavior, and be sure when upgrading to the latest release to set the mode to `standard`.
- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, `myProject`) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Upgrading API Connect subsystems

Complete the following steps to upgrade API Connect subsystems.

### Before you begin

- Ensure that you have met the requirements for upgrading API Connect subsystems in a Kubernetes environment. See [Requirements for upgrading](#).
- Ensure that you are upgrading to the *latest* Fix Pack version. These instructions apply to upgrading to the latest Fix Pack version. To access the Fix Packs, see the link in [What's New in the latest release](#).

Important: If you are upgrading from 2018.4.1.19 or earlier, to 2018.4.1.20 or later, on OpenShift then you must upgrade API Connect before upgrading OpenShift to version 4.10.

### About this task

When you apply an upgrade, the new level of the subsystem overwrites the existing level. Your user configuration, APIs, Products, and subsystem configurations (Management, Analytics, and Developer Portal) are retained.

Note: **Upgrading to 2018.4.1.13 - iFix3.0 or later:** Permissions are added to the pre-supplied API Connect user roles as follows:

- A role that had the **Api-Drafts:View** permission, but not the **Api-Drafts:Manage** permission, has the **Product-Drafts:View** permission added if not already present.
- A role that has both the **Api-Drafts:View** and **Api-Drafts:Manage** permissions has the **Product-Drafts:View** and **Product-Drafts:Manage** permissions added if not already present.

The permission settings for custom roles are not changed.

For more information on user roles, and assigning permissions to roles, see [API Connect user roles](#) and [Creating custom roles](#).

### Procedure

1. Verify the API Connect deployment is healthy and fully operational:

Version to upgrade from	Instructions
v2018.4.1.6 or later	See <a href="#">Checking cluster health on Kubernetes</a> . When upgrading from v2018.4.1.6 or later to v2018.4.1.10 or later, this step is optional because a health check is run automatically as part of the <code>apicup subsys install</code> command in step <a href="#">11</a> . Note that you might still want to run the health check now, as preparation for the manual backup in step <a href="#">2</a> .
v2018.4.1.5 or earlier	See <a href="#">Determining status of a cluster on Kubernetes</a>

2. Complete a manual backup of the API Connect subsystems. See [Backing up and restoring](#).

3. **Upgrading to Version 2018.4.1.15 or later:**

Helm v3 is required for Kubernetes upgrades. Versions of APIC prior to v2018.4.1.15 use Helm v2, so Helm v3 must be installed prior to upgrading the API Connect files. The following instructions apply to Kubernetes v1.18.x:

Note: Use the Helm version corresponding to your Kubernetes version. Review the Helm compatibility chart: [https://helm.sh/docs/topics/version\\_skew/](https://helm.sh/docs/topics/version_skew/).

- a. Backup ConfigMaps

Helm release data is stored in these ConfigMaps, and is required to go back to Helm v2.

- i. List the ConfigMaps to backup:

```
kubectl get configmaps --namespace=kube-system --selector=OWNER=TILLER --output=jsonpath='{range .items[*]}{.metadata.name} {"\n"} {end}'
```

- ii. Save each ConfigMap from the list:

```
kubectl get configmap [CONFIGMAP_NAME] -oyaml > [CONFIGMAP_NAME].yaml
```

- b. Follow the steps outlined in the official Helm documentation to migrate from Helm v2 to Helm v3. See <https://helm.sh/blog/migrate-from-helm-v2-to-helm-v3/>.

Note: The cleanup command will remove the Helm v2 Configuration, Release Data and Tiller Deployment. It cleans up all releases managed by Helm v2. It will not be possible to restore them if you haven't made a backup of the releases. Helm v2 will not be usable afterwards.

- c. Verify that `helm version` reports `v3.x.x` and `helm ls` lists all the correct releases.

4. If necessary, prepare your management database for the upgrade:

Version to upgrade from	Instructions
v2018.4.1.8 or later	No preparation required. Skip this step, go directly to Step <a href="#">5</a>

Version to upgrade from	Instructions
v2018.4.1.7 or earlier	<p>Due to schema changes, upgrade of the management database takes longer than previous upgrades. For long established deployments with a large amount of data, such as over 10 GB, upgrade can take as long as several hours. To reduce this time, use the <b>apicops</b> interface to truncate the subscriber event table and remove unused snapshots.</p> <p>Download the latest release of the <b>apicops</b> interface from <a href="https://github.com/ibm-apiconnect/apicops/releases">https://github.com/ibm-apiconnect/apicops/releases</a>, and run the following commands:</p> <pre>apicops subscriber-queues:clear apicops snapshots:clean-up</pre>

- Download the appropriate images from IBM® Fix Central. To access the Fix Packs, see the link in [What's New in the latest release](#). On the Fix Pack page, select the version you want to install. When the version contents are displayed, access the files by clicking the link **Status : Available**.

Note that for Kubernetes deployments, there is one distribution file for each API Connect subsystem. The same file is used for either upgrade or new installation.

- Change directories to your existing project directory, for example, `/myProject`, and copy the new version of the APICUP installer from the zip file into the `/myProject` directory, overwriting the previous version.
- Upload the zip files to the image registry by entering the following command: `apicup registry-upload <subsystem> <image_tgz> <registry_uri>` where:
  - subsystem\_type** is the type of the subsystem for the file you are uploading, one of **management**, **portal**, **analytics**. (The gateway image is obtained from a different location, for example, **ibmcom/datapower**).
  - image\_tgz** is the name of the tar file you are uploading.
  - registry\_uri** is the path to the registry for the tar files. Enter a new subpath within the registry for the tar files when performing an upgrade. Do not reuse the same path where you initially uploaded the tar files.
- Enter the `apicup subsys set <subsystem> registry <newImageTarget>` command from the `myProject` directory for each subsystem:

```
cd ./myProject
apicup subsys set <subsystem> registry <newImageTarget>

// apicup subsys set mgmt registry <newImageTarget>
// apicup subsys set analyt registry <newImageTarget>
// apicup subsys set ptl registry <newImageTarget>
```

**newImageTarget** refers to the registry location of new image tarballs of each subsystem. This location is same as the registry upload location (**registry\_uri**).

Warning:

The installer may request that you change the **cassandra-volume-size** during the upgrade. IBM® API Connect does not support resizing **cassandra-volume-size**. You can bypass the validation on volume sizes by using the `--no-verify` option. Before starting the upgrade, make sure that the PVC size capacities and storage class of the Cassandra pod match the values mentioned in `apiconnect-up.yml`. See the instructions in [Troubleshooting the upgrade](#) to determine the PVC size capacity and storage class for the Cassandra pod.

- For the gateway, complete the following steps:

Note:

When you upgrade a cluster of gateway pods in Kubernetes, a small number of API transactions may fail. During the upgrade, Kubernetes removes the pod from the load balancer configuration, deletes the pod and then starts a new pod. The steps are repeated for each pod. Socket hang ups occur on transactions that are in process at the time the pod is killed.

The number of transactions that fail depends on the rate of incoming transactions and the length of time needed to complete each transaction. Typically the number of failures is a very small percentage. This behavior is expected during an upgrade. If the failure level is not acceptable, schedule the upgrade during an off-hours maintenance window.

- Specify the *repository* and *image-tag* for the upgrade by entering the following commands:

```
apicup subsys set gwy image-repository=<name of repository>
apicup subsys set gwy image-tag="release_name"
```

- If you are upgrading DataPower API Gateway from Version 2018.4.1.6 or earlier to 2018.4.1.7 or later, set **enable-high-performance-peering**. High performance peering is a feature that was added for DataPower API Gateway in Version 2018.4.1.7, and must be configured during the upgrade. Version 2018.4.1.7 (and later) contains a new setting, **enable-high-performance-peering=""**. Note that the setting has no value by default.

If your deployment uses DataPower API Gateway, as specified by the configuration setting `apicup subsys set gwy v5-compatibility-mode=false`, then you must set **enable-high-performance-peering="false"** during this upgrade process:

```
apicup subsys set gwy enable-high-performance-peering="false"
```

Use of high performance peering is optional, but highly recommended. If you do not plan to use high performance peering, no further configuration is needed. If you want to use high performance peering, you must first complete the API Connect upgrade, and then follow the link in Step [15](#) at the end of these instructions.

- When upgrading a high-availability cluster, ensure that you meet the requirements:

- Gateways must be updated one at a time.
- Before starting the upgrade, a single gateway must be running as primary for all gateway-peering definitions.
- When upgrading multiple gateways, the primary gateway must be upgraded last.
- For upgrading to version 2018.4.1.9 or later, ensure that the pod with a name like `r*-dynamic-gateway-service-0` is the primary because it is the last node to be upgraded.

To determine which gateway is running as primary, use either the `show gateway-peering-status` command in the DataPower CLI, or use the Gateway Peering Status display in the WebGUI in the API Connect application domain. To move the primary to the DataPower on which you're currently working, you can issue the `gateway-peering-switch-primary <peering-object-name>` command.

To access the DataPower CLI or WebGUI, use:

```
kubectl attach -it <podname>
```

For more information, see [Configuring the API Connect Gateway Service](#).

d. To upgrade the gateway monitor pod, point to the `monitor-image-tag` for the upgrade by entering the following command:

Important: If you are upgrading from 2018.4.1.6 or earlier to 2018.4.1.7 or later, do not use this step. Instead, complete the instructions in [Adding a monitor pod during gateway upgrade](#).

```
/apicup subsys set gwy monitor-image-tag="<new_monitor_version_number>"
```

Note that you must specify an updated value on the `monitor-image-tag`. The value must differ from the value that was originally used at deployment, as described in [Installing the Gateway subsystem into a Kubernetes environment](#). For example, if the original value was `2018.4.1.7`, do not use `2018.4.1.7`, even if the contents of the tag are different.

When you run `apicup subsys install` command again, `apicup` detects the changed value and restarts the monitor pod.

e. **Version 2018.4.1.9:** All gateway extensions and gateway script policies which were in place prior to the upgrade to Version 2018.4.1.9 are automatically moved from `local://` to `temporary://`. Therefore, any references to `local` in those extensions or policies must be changed to refer to `temporary`.

10. Apply the new Cassandra CRDs before upgrading the management subsystem.

a. Download the `cassandra-crds_lts_v2018.4.1.20.yaml` CRD from [IBM Fix Central](#).

b. Run the following command to renew the CRDs:

```
kubectl apply -f <CustomResourceDefinition.yaml>
```

Where `<CustomResourceDefinition.yaml>` is the same file name as the one created to define the CRDs. This renews the two required CRDs, `cassandraclusters.apic.ibm.com` and `cassandraclusterbackups.apic.ibm.com`.

11. Run `apicup subsys install` for each subsystem. Upgrade a single subsystem at a time and check the health status of each subsystem before upgrading the next subsystem. See [Checking cluster health on Kubernetes](#).

```
cd ./myProject
apicup subsys install <subsystem>
```

```
// apicup subsys install mgmt
// apicup subsys install analyt
// apicup subsys install ptl
// apicup subsys install gwy
```

12. **Version 2018.4.1.9 iFix1.0 and later:** After completion of the upgrade, verify that all tasks are running.

Due to a known limitation in versions prior to v2018.4.1.9 iFix1.0, some tasks may have stopped running. Complete the following steps:

a. Download the `apicops` utility from <https://github.com/ibm-apiconnect/apicops/releases>.

b. Run the following command to remove any pending tasks:

```
$ apicops task-queue:fix-stuck-tasks
```

c. Run the following command to verify that the returned list (task queue) is empty.

```
$ apicops task-queue:list-stuck-tasks
```

13. Upgrade Kubernetes or OpenShift if needed:

- Kubernetes: If the your version of Kubernetes is not supported by the version of API Connect that you upgraded to, obtain a supported version and install it. To review the list of supported versions of Kubernetes, follow the instructions in [IBM API Connect Version 2018 software product compatibility requirements](#).

- OpenShift: If you upgraded from 2018.4.1.19 or earlier, to 2018.4.1.20 or later, upgrade OpenShift now to the level supported by your new version of API Connect.

For example, if you upgraded from 2018.4.1.19 on OpenShift 4.6 to 2018.4.1.20, you must now upgrade OpenShift to 4.10.

14. If you upgraded the Portal, see [Verifying Portal upgrade](#).

15. If you want to enable high performance peering for a DataPower API Gateway, see [Enabling high performance peering for DataPower API Gateway on Kubernetes](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding a monitor pod during gateway upgrade

Version 2018.4.1.7 introduced a new Gateway monitor pod that is required during an upgrade.

### About this task

The Gateway monitor pod feature was introduced in Version 2018.4.1.7. The monitor pod is required on all Gateway deployments for Version 2018.4.1.7 and later. When upgrading from Version 2018.4.1.6 or earlier, you must install (add) a new monitor pod to your deployment. The procedure in this topic is for this upgrade scenario.

Note: If you have already deployed Version 2018.4.1.7 or later, and are upgrading to a newer version, do not use this procedure. In this case, your deployment includes an existing gateway monitor pod.

### Procedure

1. Download the distribution file.

The monitor pod is distributed in a separate tar file on IBM Fix Central, along with the other API Connect distribution files. For example, the file for Version 2018.4.1.7 is `dpm2018417.1ts.tar.gz`. For more information, see [First steps for installing API Connect: Upload files to registry](#).

2. Load the Kubernetes DataPower Monitor image to your local docker registry:

```
docker load -i dpm2018417.1ts.tar.gz
```

For these examples, assume the version number is 2018.4.1.7. The image is automatically tagged in the docker registry to

```
ibmcom/k8s-datapower-monitor:2018.4.1.7
```

3. Re-tag the image for the new registry with the registry host, image name, and tag in the following format: `<REGISTRY_HOST>/<IMAGE_NAME>:<TAG>`:

```
docker tag <existing REGISTRY_HOST>/<IMAGE_NAME>:<TAG>> <new REGISTRY_HOST>/<IMAGE_NAME>:<TAG>>
```

For example:

```
docker tag ibmcom/k8s-datapower-monitor:2018.4.1.7 MyRegistry/k8s-datapower-monitor:2018.4.1.7
```

The parameters are defined as follows:

- `<REGISTRY_HOST>/<IMAGE_NAME>` - Enter your registry host name and an image name, for example, *My Registry/k8s-datapower-monitor*. The `<REGISTRY_HOST>/<IMAGE_NAME>` matches the value entered for installation in the `apicup subsys set gwy monitor-image-repository` command.
  - `<TAG>` - Enter a tag for the image, for example, `2018.4.1.7`. The `<TAG>` value must match the value entered for the gateway image tag during installation using the `apicup subsys set gwy monitor-image-tag` command. You can look up required tag using the `./apicup version --images` command to list all images required by APICUP. The tag must match the tag for the Kubernetes DataPower Monitor. For example: `2018.4.1.7`.
4. Push the monitor pod image to the new registry:

```
docker push <REGISTRY_HOST>/<IMAGE_NAME>:<TAG>
```

For example:

```
docker push MyRegistry/k8s-datapower-monitor:2018.4.1.7
```

5. Use `apicup` to configure the monitor pod:

For example:

```
apicup subsys set gwy monitor-image-repository="ibmcom/k8s-datapower-monitor"
```

```
apicup subsys set gwy monitor-image-tag="2018.4.1.7"
```

When you run `apicup subsys install` command, `apicup` includes the monitor pod.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Verifying Portal upgrade

You can monitor the progress of existing Portal sites being upgraded to the new version.

### Procedure

---

1. After completing an upgrade of the Portal subsystem, there may be a delay while the existing sites are upgraded to the new platform version. Once all Portal pods have completed the upgrade, you can monitor the progress of the sites being upgraded using the following commands:
  - `apicup subsys exec portal list-sites sites` - Displays the status of the sites that are being upgraded to the new platform version. Any sites currently upgrading will be listed with an `UPGRADING` status. When a site is finished upgrading, it will have an `INSTALLED` status, with the new platform version listed.
  - `apicup subsys exec portal list-sites platforms` - Confirm that the new version of the platform is listed for all sites with `INSTALLED` status.
2. The installer may request that you change storage sizes during the upgrade. Unless your persistent storage solution supports manual resizing, do not change the size of the volumes. You can bypass the validation on volume sizes using the `--no-verify` option. The following storage size settings may be affected:

```
www-storage-size-gb  
backup-storage-size-gb  
db-storage-size-gb  
db-logs-storage-size-gb
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting the upgrade

Troubleshoot problems with the upgrade.

### Health check error for Cassandra pod

---

API Connect does not support resizing the storage capacity or modifying the storage class of the Cassandra database.

After completing the upgrade, if the health check of the management system returns an **Error**: for any component, and the problem persists, use the following steps to debug the cluster.

1. Verify that all upgrade jobs have completed. Upgrades can take some time, depending on the version. You can monitor the health check to ensure that all upgrade jobs and pods are no longer running. See [Checking cluster health on Kubernetes](#).
2. If upgrade pods are completed and all pods are up and running, verify the health status of the Cassandra pods. You might see a problem similar to the following:

```
./apicup subsys health-check <mgmt-subsystem>
```

```
Error: Problem checking cassandrapod(<namespace>.<cassandra-pod>): Expect pod image(<current-pod-image>) == <Expected-image>
```

The error described means that the Cassandra pod image is not up to date with expected image. If the error message persists, it means that the Cassandra operator failed to upgrade the Cassandra servers. In most cases, the failure occurs because `cassandra-volume-size-gb` or `storage-class` changed. To determine if either of these settings changed:

- a. Look for errors in the `cassandra-operator` logs. For example:

```
time="2019-06-17T15:30:56Z"
level=error msg="Error while processing work item:
Unable to process pt/r59a99aa884-apiconnect-cc:statefulset.
Giving up: StatefulSet.apps \"r59a99aa884-apiconnect-cc\" is invalid: spec:
Forbidden: updates to statefulset spec for fields other than 'replicas', 'template',
and 'updateStrategy' are forbidden."
```

- b. Compare the value for `cassandra-volume-size-gb` in `apiconnect-up.yml` with the actual Cassandra PersistentVolumeClaim (PVC) capacity. Run the following command to determine the PVC capacity:

```
kubectl get pvc -n <namespace> | grep -e 'apiconnect-cc' -e 'CAPACITY'
```

- c. Compare the value of `storage-class` in `apiconnect-up.yml` with actual Cassandra PVC storage. Run the following command to determine the `storage-class` value for the PVC attached to the Cassandra pod:

```
kubectl get pvc -n <namespace> | grep -e 'apiconnect-cc' -e 'STORAGECLASS'
```

3. **Workaround:** If necessary, use the following steps to upgrade the Cassandra servers:

- a. Ensure that the value of `cassandra-volume-size-gb` in `apiconnect-up.yml` matches the `apiconnect-cc` PVC claim capacity.
- b. Ensure that the value of `storage-class` in `apiconnect-up.yml` matches the `apiconnect-cc` PVC `storage-class`.
- c. If values don't match, revert back changes in `apiconnect-up.yml` so that they match the `apiconnect-cc` PVC capacity and `storage-class`.
- d. Update the management subsystem:

```
apicup subsys install <management-subsystem>
```

Cassandra servers will be automatically upgraded by the Cassandra operator.

- e. Allow sufficient time for the upgrade to complete, and then check the status of the cluster.

## Issues when installing Drupal 8 based custom modules or sub-themes into the Drupal 9 based Developer Portal

From IBM® API Connect 2018.4.1.17, the Developer Portal is based on the Drupal 9 content management system. If you want to install Drupal 8 custom modules or sub-themes into the Drupal 9 based Developer Portal, you must ensure that they are compatible with Drupal 9, including any custom code that they contain, and not using any deprecated APIs, for example. There are tools available for checking your custom code, such as [drupal check](#) on GitHub, which checks Drupal code for deprecations.

For example, any Developer Portal sites that contain modules or sub-themes that don't contain a Drupal 9 version declaration will fail to upgrade, and errors like the following output will be seen in the `admin` logs:

```
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: Checking theme: emeraldgreen
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '' found for emeraldgreen
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: Checking theme: rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '8.x' found for rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Found themes incompatible with
Drupal 9: emeraldgreen rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: /tmp/restore_site.355ec8 is NOT
Drupal 9 compatible
...
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: Checking module: custom_mod_1
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '' found for custom_mod_1
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: Checking module: custom_mod_2
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '8.x' found for custom_mod_2
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Found modules incompatible with
Drupal 9: emeraldgreen rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: site1.com is NOT Drupal 9
compatible
```

To fix version compatibility errors, all custom modules and sub-themes should declare a `core_version_requirement` key in their `*.info.yml` file that indicates Drupal 9 compatibility. For example:

```
name: Example module
type: module
description: Purely an example
core: 8.x
core_version_requirement: '^8 || ^9'
package: Example module
```

```
# Information added by Drupal.org packaging script on 2020-05-31
version: '8.x-1.3'
project: 'example_module'
datestamp: 1590905415
```

This example specifies that the module is compatible with all versions of Drupal 8 and 9. For more information, see [Let Drupal know about your module with an .info.yml file](#) on the drupal.org website.

If you have a backup of a site that you need to restore, and are getting the version compatibility error, but the module or theme \*.info.yml file cannot be changed easily, then you have two options. Either:

- Add an environment variable for the `www` pod of the `admin` container stating `SKIP_D9_COMPAT_CHECK: "true"`. However, if you choose this method, you must be positive that all of the custom modules and themes for your sites are Drupal 9 compatible, as otherwise the sites may end up inaccessible after the upgrade or restore. Create an extra values file to contain the environment variable, as follows:

```
apic-portal-www:
  admin:
    env:
      SKIP_D9_COMPAT_CHECK: "true"
```

Save the file as `d9compat.yaml`, and run the following command:

```
apicup subsys set <portal_subsystem_name> extra-values-file d9compat.yaml
```

Then, update the portal with the updated setting by running the following command:

```
apicup subsys install <portal_subsystem_name>
```

Or:

- Extract the site backup, edit the relevant files inside it, and then tar the backup file again. Note that this procedure will overwrite the original backup file, so ensure that you keep a separate copy of the original file before you start the extraction. For example:
  1. `mkdir /tmp/backup`
  2. `cd /tmp/backup`
  3. `tar xfz path_to_backup.tar.gz`
  4. Edit the custom module and theme files to make them Drupal 9 compatible, and add the correct `core_version_requirement` setting.
  5. `rm -f path_to_backup.tar.gz`
  6. `tar cfz path_to_backup.tar.gz`
  7. `cd /`
  8. `rm -rf /tmp/backup`

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deleting the API Connect deployment in a Kubernetes runtime environment

To delete the Kubernetes deployment of API Connect, you delete the Helm charts, Custom Resource Definitions, the Persistent Volumes, and the namespace.

### About this task

To delete the API Connect deployment in a Kubernetes runtime environment, you will need the Kubernetes command-line tool, `kubectl`. For more information, see [kubectl](#).

### Procedure

Repeat the following steps for each namespace containing an API Connect subsystem:

1. List the Helm charts in your namespace by entering: `helm ls --namespace=<namespace>`. You will perform a delete for each chart. The charts will look similar to the following:
  - `dynamic-gateway-service-*`
  - `cassandra-operator-*`
  - `dynamic-gateway-*`
  - `apic-analytics-*`
  - `apiconnect-*`
  - `apic-portal-*`
2. Delete each Helm chart using the `--purge` option: `Helm delete --purge <chart-name>`. If the reclaim policy of a Persistent Volume is `Delete`, the PV is automatically deleted when the Helm chart is deleted.
3. List and delete the Custom Resource Definitions (CRDs). Enter `kubectl get crd` to list the CRDs. Delete the following API Connect CRDs by entering the following command: `kubectl delete crd <name>`:
  - `cassandraclusters.apic.ibm.com`
  - `cassandraclusterbackups.apic.ibm.com`
4. Check if there are remaining Persistent Volumes associated with API Connect by entering: `kubectl get pv`. If there are any remaining PV, delete them manually using the following command: `kubectl delete pv <name>`.
5. Delete the namespace if it is no longer needed. Ensure there are no other resources deployed in the namespace before deleting it. Enter the following command to delete the namespace: `kubectl delete namespace <name>`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Maintaining a Kubernetes deployment

You can use utilities to complete maintenance tasks such as backup, restore, and certificate management on Kubernetes.

Note: When maintaining API Connect, do not use `kubectl exec` or `oc exec` commands to access API Connect pods unless advised by IBM.

- **[Backing up and restoring](#)**  
You can backup and restore API Connect subsystems.
- **[Disabling the Analytics subsystem on Kubernetes](#)**  
During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Backing up and restoring

You can backup and restore API Connect subsystems.

Note:

- Backups are intended for recovery of the Management, Portal, and Analytics subsystems onto the same deployment from which they were taken, or onto a new replacement installation in the same environment for disaster recovery. The same environment means the same network configuration and project directory as the original installation.
- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.
- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.
- **[Backing up the management database in a Kubernetes environment](#)**  
Backups of the management database can be performed based upon a cron-like schedule or on-demand from the command line.
- **[Restoring the management database in a Kubernetes environment](#)**  
The management database can be restored as a complete restoration. Partial restorations are not supported.
- **[Backing up and restoring the Developer Portal in a Kubernetes environment](#)**  
How to backup and restore your Developer Portal service in your Kubernetes environment.
- **[Backing up and restoring the analytics database](#)**  
The analytics database can be backed up and restored from an S3 repository. S3 compatible object storage is required, for example, IBM Cloud Object Storage.
- **[Using APICUP to reconfigure](#)**  
You can use APICUP to change configuration of a subsystem after completion of initial installation.
- **[Configuring maximum size of client requests to the Management subsystem](#)**  
You can configure the maximum allowed size of the client request body for requests made to the Management subsystem.
- **[Increasing memory allocation for Cassandra operator on OpenShift](#)**  
If the Cassandra operator pods restart due to out of memory errors, you can increase the memory allocation for the Cassandra operator.
- **[Setting rate limits for public APIs on the management service for a Kubernetes environment](#)**  
Describes the procedure for setting a rate limit for public APIs on the management service. Rate limits provide protection from DDoS (distributed denial of service) attacks.
- **[Enabling high performance peering for DataPower API Gateway on Kubernetes](#)**  
Enabling high performance peering on DataPower API Gateway deployments optimizes API transaction performance.
- **[Dynamically re-registering and reconfiguring a Gateway service in a Kubernetes deployment](#)**  
In API Connect, Gateway services do not persist their configuration settings by default. Instead, the master configuration is stored on the Management server and the *Dynamic Reregistration and Reconfiguration* (DRR) mechanism resynchronizes configuration data when needed. The DRR process is used when proper High Availability/Disaster Recovery (HA/DR) is not configured, or if a manual resynchronization is required.
- **[Monitoring and logging a Kubernetes deployment](#)**  
You can do health checks to determine program status, and can gather logs for use in troubleshooting.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Backing up the management database in a Kubernetes environment

Backups of the management database can be performed based upon a cron-like schedule or on-demand from the command line.



## About this task

Database backups can be used to restore the database for disaster recovery or for transferring data during an upgrade.

- For scheduled backups, see [Configuring scheduled backups](#)
- For on-demand backups, see [Performing on-demand backups](#)

We strongly recommend that you configure a backup schedule for your management database using the `cassandra-backup-schedule` setting during installation of the management subsystem. If you did not do so when you installed API Connect in your Kubernetes runtime environment, you have the option to perform an on-demand backup. We also recommend that you perform a backup (either scheduled or on-demand) of the management database prior to upgrading.

Both scheduled backups and on-demand backups require that the `cassandra-backup-x` settings be configured when installing the management subsystem. Automatic scheduled backups are performed according to the cron-like job configured by the `cassandra-backup-schedule` setting. On-demand backups also require the backup settings for protocol, host, etc., but are run on-demand from the command line.

Note:

- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.
- The original project directory created with APICUP during the initial product installation (for example, myProject) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from where it can always be retrieved.

## Procedure

### • Configuring scheduled backups

1. To configure scheduled backups, enter the `cassandra-backup-x` settings when installing the management subsystem as described in [Installing the Management subsystem into a Kubernetes environment](#). The `cassandra-backup-x` parameters are also described in the following table:

The parameters are as follows:

Parameter	Description
<code>cassandra-backup-protocol</code>	The backup protocol. Specify one of the following values: <ul style="list-style-type: none"> <li>• <code>sftp</code> - for secure file transfer protocol</li> <li>• <code>objstore</code> - for S3 compatible object storage</li> </ul> Note: <ul style="list-style-type: none"> <li>• For the management subsystem, IBM Cloud and Amazon Web Services (AWS) are supported S3 object store providers.</li> <li>• The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code>.</li> </ul>
<code>cassandra-backup-path</code>	The full path to the directory where the backup files will be stored. This must point to a directory on the backup server. For object storage ( <code>objstore</code> ), the path can be set to the <code>bucket</code> value or the <code>bucket/subfolder</code> value.
<code>cassandra-backup-host</code>	The fully qualified domain name of the backup server. Ensure that the Kubernetes nodes can access this host. If using object store, enter <code>Endpoint/Region</code> . (The "/" character between the endpoint and region are required for this setting.)
<code>cassandra-backup-port</code>	The port for the protocol to connect to the <code>cassandra-backup-host</code> . The backup port is not required for object storage.
<code>cassandra-backup-schedule</code>	Cron like schedule for performing automatic backups. The format for the schedule is: <ul style="list-style-type: none"> <li>• <code>*****</code></li> <li>• <code>-----</code></li> <li>• <code>     </code></li> <li>• <code>     +-----</code> day of week (0 - 6) (Sunday=0)</li> <li>• <code>   +-----</code> month (1 - 12)</li> <li>• <code>  +-----</code> day of month (1 - 31)</li> <li>• <code> +-----</code> hour (0 - 23)</li> <li>• <code>+-----</code> min (0 - 59)</li> </ul> The backup schedule defaults to <code>0 0 * * *</code> . This means a backup is run every day at midnight and minute zero. The timezone for backups is UTC. <p>When you configure a host, if you do not specify a value for <code>cassandra-backup-schedule</code>, the default backup schedule is automatically set. Note that the default backup schedule is not set, and scheduled backups not enabled, until host configuration is completed.</p> Note: Cassandra <b>repair</b> cron schedule is set to <code>00 1 * * 0,2,4,6</code> . This means the repair runs at 01:00 on Sunday, Tuesday, Thursday, and Saturday. By default, the Cassandra <b>backup</b> cron schedule should not run within one hour of the repair cron schedule. Please make sure to modify the current backup configuration as needed. If backups and repairs run at the same time, backup processes can fail intermittently.
<code>cassandra-backup-auth-user</code>	The username for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Key ID.
<code>cassandra-backup-auth-pass</code>	The password for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Access Key parameter. The password will be stored in Base64 encoded format. <p>For example:</p> <pre>apicup subsys set mgmt cassandra-backup-auth-pass '&lt;password&gt;'</pre> <p>Note that you cannot use the '=' sign to assign the password to <code>cassandra-backup-auth-pass</code>.</p>

For example:

```
cassandra-backup-protocol: sftp
cassandra-backup-host: mybackuphost.com
cassandra-backup-port: 22
cassandra-backup-path: /backups
cassandra-backup-schedule: 0 0 * * *
cassandra-backup-auth-user: myusername
cassandra-backup-auth-pass: mypassword
```

- These settings will be activated by the `apicup subsys install <SUBSYS_NAME>` command.
- To verify that automatic backups are in progress, you will see a pod labeled `<cassandra cluster name>-backup` at the scheduled time for the backup.
- When the scheduled backup is complete, the backup files are stored at the location specified by the `cassandra-backup-path` parameter. The file name varies depending upon the API Connect version that performed the backup, but the file name must be compatible with the version used for restoring the database. For details about the file name, see [Restoring the management database in a Kubernetes environment](#).
- List the backups by entering: `apicup subsys exec <SUBSYS_NAME> list-backups`. You can view a list of backups with the ID and Status in the output. Following is an example:

Cluster	Namespace	ID	Timestamp	Status
rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC	Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC	Complete

- #### Performing on-demand backups

- Configure the backup parameters. (The same parameters are required for on-demand backup that are required for a scheduled backup, but you will bypass the schedule.) If you did not enter the `cassandra-backup-x` parameters when you installed API Connect, you will need to enter them from the command line and re-install the management subsystem before performing an on-demand backup.
- After the backup parameters have been specified, run the `apicup subsys install <SUBSYS_NAME>` command to activate the new parameters. This step is not required if you specified these parameters during the initial installation.
- Enter the following command: `apicup subsys exec <SUBSYS_NAME> backup` where `SUBSYS_NAME` is the name of your management subsystem.
- When the back is completed, the backup files are stored at the location specified by the `cassandra-backup-path` parameter. The file name varies depending upon the API Connect version that performed the backup, but the file name must be compatible with the version used for restoring the database. For details about the file name for the backups, see [Restoring the management database in a Kubernetes environment](#).
- List the backups by entering: `apicup <subsys exec <SUBSYS_NAME> list-backups`. You can view a list of backups with the ID and Status in the output. Following is an example:

Cluster	Namespace	ID	Timestamp	Status
rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC	Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC	Complete

## Results

Use the ID generated for the backup files to restore the database from the backup. Usually the database is restored from a backup file after an upgrade is performed and to recover from a disaster. See [Restoring the management database in a Kubernetes environment](#) and [Upgrading API Connect in a Kubernetes environment](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Restoring the management database in a Kubernetes environment

The management database can be restored as a complete restoration. Partial restorations are not supported.

### Before you begin

Before you being restoring a management database, ensure your deployment and your process meets these requirements:

- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- Restoring the Management Service requires database downtime and is a destructive process that deletes current data and copies backup data. During the restoration process, external traffic must be stopped.
- In a Disaster Recovery scenario, do not log in to the administration UI or attempt to configure or change any settings prior to restoring the backup. Restore the backup immediately after installing the subsystem.
- To restore the management database, you must use the original project directory that was created with `apicup` during the initial product installation. You cannot restore the database without the initial project directory because it contains pertinent information about the cluster. The endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored system. Note that successful restoration depends on use of a *single apicup* project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.
- Map the DNS entries from the source cluster to the corresponding IP addresses on the target cluster. Record the DNS entries for each endpoint before starting the restore.
- When restoring the management database, the endpoints (on the new cluster which is the target for the restoration) have to be the same as those on the old cluster (the source of the backup). This includes all the endpoints for API Connect: `api-manager-ui`, `cloud-admin-ui`, `consumer-api`, `platform-api`; `api-gateway`, `api-gw-service`; `analytics-ingestion`, `analytics-client`; and `portal-admin`, `portal-www`.
- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.

### About this task

Follow the procedure on this page to restore your management database. You must complete the prerequisite steps before beginning the restore.

Note that in a disaster recovery scenario you must first re-establish the management subsystem. The procedure includes an optional first step for disaster recovery.

If you encounter errors, see [Troubleshooting restoration of management database](#). Note that the troubleshooting page includes [Overview of restore process for management database](#).

## Procedure

- If the restoration is for a disaster recovery scenario, complete this step to first install a new Management subsystem. If the restoration is *not* for a disaster recovery (meaning that you have a *running* Management subsystem to use for restoration), skip this step and go directly to Step 2.
  - Copy the project folder that corresponds to the backup files into a new location.
  - If, for the previous installation, you have redirected the configuration to an optional output folder using the `apicup subsys install mgmt --out=mgmt-out` command as explained in [Installing the Management subsystem into a Kubernetes environment](#), delete all output folders prior to starting the new installation.
  - (Optional) If the image registry location or the secret has changed, update the image registry and registry-secret for each subsystem, as follows:

- `apicup subsys set <SUB_SYS> registry <REPOSITORY>`

The location where you are running the image registry (for example, Artifactory). `<SUB_SYS>` is the name of the subsystem. `<REPOSITORY>` is the URL to the repository.

- `apicup subsys set <SUB_SYS> registry-secret <SECRET>`

Kubernetes secret with Docker credentials to authenticate with the image registry. The value for `registry-secret` must match the name of the secret created using the `kubectl create secret` command. The secret contains the Docker credentials for accessing the registry. `<SUB_SYS>` is the name of the subsystem. `<SECRET>` is the repository secret.

- Perform a fresh installation of the management subsystem using the following command:

```
apicup subsys install <SUB_SYS>
```

Important:

Do not make any other changes in the project directory nor run any other `apicup` operations before proceeding to the next step.

- Verify that your deployment meets the prerequisites for restoring a management database:
  - Verify that you are about to restore onto a deployment that has the same number of Cassandra pods as the deployment where the backup was created. For example, if the management service backup deployment had 3 Cassandra pods, the restore deployment must have 3 Cassandra pods. You cannot restore onto a deployment with fewer Cassandra pods because the Cassandra data is sharded across the pods. To successfully restore, first create the matching number of Cassandra pods.  
To view the number of pods, enter the `kubectl get pods` command.
  - Ensure that all Cassandra pods are on-line and running normally.
- Restoration of a management subsystem requires access to backup tar files. Obtain your backup files, as follows:
  - Enter `apicup subsys exec <MANAGEMENT_SUBSYS_NAME> list-backups`. The output lists the current backups in your namespace with Backup ID and status.

Cluster Status	Namespace	ID	Timestamp
rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC Complete

The backup files are stored at the location specified by the `cassandra-backup-path` parameter.

Examine the backup filename to identify the backup ID for use with the restore command.

Table 1. Backup file naming convention

Backup file name	BackupID (for use with restore command)
<code>&lt;backupId&gt;-&lt;pod index&gt;-&lt;# of pods&gt;.tar.gz</code>	<code>[backupId]</code>
Example:	Example:
<code>1534954510365016356-0-3.tar.gz</code>	<code>1534954510365016356</code>

Note: **Disaster Recovery:** If you are restoring on a new deployment, you will not have any backups. You will have to find the `backupID` by going to your backup host and looking for the backup files from previous deployments.

- Confirm that you have a backup file for each Cassandra pod in your cluster. If you have  $n$  pods, you must have the same number of backup files.
- Optional but recommended:** Verify the integrity of the backup tar files.  
Ensure that the tar files are not corrupt. For example, on Linux:

```
tar -tzf <backup_file>
```

- Optional but recommended:** Determine whether your environment has sufficient space to perform the restore. You need enough free space such that the size of the backup file, when multiplied by four, does not exceed 85% of the available free space.  
For example, on Linux, you can use the following steps:

- Exec a bash terminal on the Cassandra pod:

```
kubectl exec -it <Cassandra-pod> --bash
```

- Paste the following script to calculate space available for restore:

```
# current available space
avail=$(df /var/db/ | awk 'NR==2{print $4}')

#space occupied by /var/db/data/
db_data=$(du -s /var/db/data/ | awk '{print$1}')

# Estimated available space after cleanup (avail + db_data) and a buffer of 15%
total_space_avail=$((($avail + $db_data) * 1024) * 85 / 100)
echo $total_space_avail
```

The value obtained in the above script is in bytes and must be calculated for every pod, and compared against  $4x$  the value, where  $x$  is the backup tar size of the corresponding backup file.

Each backup file is in format `<backup-id>-<ordinal-of-cassandra pod>-<number-of-cassandra-pods in cluster>.tar.gz`

In the example script above, if the backup tar size of `<backup-id>-0-3.tar.gz` is `15*1024*1024` bytes, the value for `$total_space_avail` in Cassandra `cc-0` pod must be around `60*1024*1024` bytes.

If free space is insufficient, create additional free space before starting the restore.

- Restore the management database by entering:

```
apicup subsys exec <MANAGEMENT_SUBSYS_NAME> restore <backupID>
```

The `restore` command will restore all backups from files with the same backupID. You must have the same number of management database (Cassandra) pods running as the number of backup files that match the backupID.

- Verify that the restore process completed successfully.

Ensure that the restore job is marked as completed. Note, however, that it is possible for the restore job to be marked complete, but the Cassandra restore is not complete.

The best way to ensure that the Cassandra restore is complete is to review the `CassandraRestoreStatus` field in the `CassandraClusters` Custom Resource. When the `CassandraRestoreStatus` is `completed`, the Cassandra database is successfully restored.

Example command flow to verify the restore:

- Examine the restore job and pod:

```
# kubectl get jobs | grep restore
restore-plnfs 0/1 71s 71s
```

```
# kubectl get pods | grep restore
restore-plnfs-hr2rh 0/2 Init:0/2 0 54s
```

- Since the status of the restore pod is in `Init:0/2`, the first init container (Container `restore`) is being executed. In this case, watch the `ClusterRestoreStatus` inside `CassandraCluster` (cc) Custom Resource to see the current status of the Cassandra restore process.

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Running Retrieve checks on backup file 1583268283534609276-1-3.tar.gz for pod rdd94fb4a21-
apiconnect-cc-1
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Running Retrieve checks on backup file 1583268283534609276-2-3.tar.gz for pod rdd94fb4a21-
apiconnect-cc-2
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Restore prelim checks passed for rdd94fb4a21-apiconnect-cc-2
```

- When the restore process is complete, examine the restore pod and job status. Make sure `ClusterRestoreStatus` is marked as `completed`.

```
kubectl get jobs | grep restore
restore-plnfs 1/1 10m 7h17m
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: completed
```

If you encounter errors, see [Troubleshooting restoration of management database](#).

- Version 2018.4.1.9 iFix1.0 and later:** After completion of the restore, verify that all tasks are running. Complete the following steps:

- Download the `apicops` utility from <https://github.com/ibm-apiconnect/apicops/releases>.

- Run the following command to remove any pending tasks:

```
$ apicops task-queue:fix-stuck-tasks
```

- Run the following command to verify that the returned list (task queue) is empty.

```
$ apicops task-queue:list-stuck-tasks
```

- [Troubleshooting restoration of management database](#)

You can troubleshoot problems with restoring the management database.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting restoration of management database

You can troubleshoot problems with restoring the management database.

Review the information on this page to understand the steps you can take to troubleshoot a failed restore. Be sure to first read the [Overview of restore process for management database](#) to understand the logging and error reporting.

Note that for Version 2018.4.1.10 there is a known limitation with error reporting. See [Known limitation: failed restores not reporting properly](#).

- [Overview of restore process for management database](#)
- [Error: invalid backup host credentials](#)
- [Error: insufficient disk space](#)
- [Error: failure on preliminary backup checks on cc-1 and cc-2](#)
- [Known limitation: failed restores not reporting properly](#)
- [Frequently asked questions](#)

If you cannot resolve a failed restore, contact IBM Support for assistance.

## Overview of restore process for management database

The API Connect restore process is started by the command `apicup subsys exec <management_subsys> restore <backupID>`. The apicup installer starts a Kubernetes job named `restore-<id>`, which in turn starts a pod named `restore-<job-id>-<pod-id>`.

API Connect restore process

The management restoration pod consists of:

- `cassandra-restore` -- The main responsibility of this init container is to perform the Cassandra database restore. If this container encounters an error, the restoration job is Failed with pod status of `InitError (0/2)`.  
Note: For Version 2018.4.1.10 and earlier, this container is called `job-container`.
- `Cassandra-health-check` - An init container that verifies that the Cassandra health status is in a healthy state. .
- `Lur-upgrade-job` - This container checks for schema mismatches between restored data and the currently running installation. If necessary, the container performs a schema upgrade.
- `Apim-upgrade-job` - This container checks for schema mismatches between restored data and the currently running installation. If necessary, the container performs a schema upgrade. The container also resyncs all the gateway services.

The init containers start sequentially. The second init container starts only once the first init container has succeeded. Containers inside the pod start immediately after all the init containers in the pod are started. For more info, see the Kubernetes documentation: [Understanding pod status](#)

Cassandra restore process

The Cassandra restore process is performed by the init container `cassandra-restore`. Backups are required from each Cassandra pod. The container process flow is:

1. Perform preliminary retrieval checks.
2. Download the backup tar file onto the Cassandra container.
3. Verify the backup tar file integrity.
4. Stop the Cassandra process.
5. Perform the Cassandra restore.
6. Start the Cassandra process with restored data.

If any of the preliminary checks fail, error messages are returned. The error conditions must be fixed, and then the restore process can be run again.

Note that API Connect Version 2018.4.1.10 (and later) performs extensive preliminary checks prior to beginning a restore. Running the checks extends the time required to complete a restore. In particular, restoration of large backup files (larger than 5 GB) take longer.

Logging

Both init container and container logs are available during the restore process. To obtain logs from a restore pod:

```
kubectl logs <restore-pod> -n <namespace> -c <init container/container name>
```

The `apicup` command produces logs from all init and main containers sequentially as soon they finish, either successfully or with errors.

Cassandra restore logging

The logs for the init container are updated only upon completion (success or failure). To get accurate status information for the current state of the Cassandra restore process, review the `ClusterRestoreStatus` field in the `CassandraClusters` Custom Resource:

```
kubectl get cc -n <namespace> -o yaml | grep -A 2 ClusterRestoreStatus
```

## Error: invalid backup host credentials

Example output when this problem occurs:

```
./apicup subsys exec mgmt restore 158326828353460927

Cluster          Namespace      Backup Name      Backup Retrieval Timeout(hrs)  Status
rdd94fb4a21-apiconnect-cc  niharns1      1583268283534609276  24                               Started

Restore failed
Error: rpc error: code = Unknown desc =
Pod name: rdd94fb4a21-apiconnect-cc-0

Error:
ssh: Could not resolve hostname 9.30.251.186.xxx: Name or service not known
Couldn't read packet: Connection reset by peer
**** [ Wed Mar  4 04:04:25 UTC 2020 ]prelimRetrieve 0: Preliminary Retrieve checks          FAILED ****
[ Wed Mar  4 04:04:25 UTC 2020 ] Preliminary retrieve checks failed on all 1 attempts.      ABORTING restore

Error: unable to get log stream for container cassandra-health-status, pod restore-hhrsc
-2slbm, job restore-hhrsc: container "cassandra-health-status" in pod "restore-hhrsc
-2slbm" is waiting to start: PodInitializing
```

ClusterRestoreStatus in Cassandra Clusters CR:

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: 'Restore Failed: Retrieve preliminary checks failed for rdd94fb4a21-apiconnect-cc-0'

kubectl get pods | grep restore
restore-hhrsc-2slbm          0/2    Init:Error  0          26m

kubectl get jobs | grep restore
restore-hhrsc                0/1          27m        27m
```

## Error: insufficient disk space

**Issue:** cc-0 prelim checks can reject restore process complaining about insufficient disk space.

**Workaround:** Increase the disk space (4X size of Cassandra backup tar file size) allocated to all Cassandra nodes on a fresh install and re-attempt restore. See [Freespace check](#).

## Error: failure on preliminary backup checks on cc-1 and cc-2

**Issue:** Failure on preliminary backup checks on cc-1 and cc-2 can leave the system in non-ready state

**Workaround:**

1. Always look at `ClusterRestoreStatus` in `CassandraClusters` Custom Resource for failed or completed Cassandra restore status, regardless of restore job status. In this case, the restore job is stuck on health status check, and Cassandra cc-0 will be in a non-ready state.
2. Execute the following command on each Cassandra pod sequentially starting with cc-0. Wait for the Cassandra pod to become ready (1/1) before executing on other Cassandra pods. If the Cassandra pod is already in ready state (1/1), you do not need to wait, just run the command.

```
kubect1 exec -it <cassandra-pod-X> -n <namespace>
-- sh -c 'rm -rf /var/db/.restore && rm -rf /var/db/restore/*'
```

In the above command the X in `<cassandra-pod-X>` stands for the numerical value (ordinal) of the Cassandra pod, such as (0,1,2).

3. Review the Cassandra operator logs to figure out why restore has failed and fix the problem.  
To see an example of this type of failure, review [Example 2: Corrupted backup tar file cc-1 not reporting properly](#).

## Known limitation: failed restores not reporting properly

**Issue:** Failure on preliminary backup checks on cc-0, such as a corrupt tar file or incomplete download of a backup tar file, can cause the restore job to complete, but the underlying `ClusterRestoreStatus` is marked as `Restore Failed`: `<Reason>`.

**Workaround:** Always check `ClusterRestoreStatus` in the `CassandraClusters` custom resource for failed or completed Cassandra restore status, regardless of restore job status. Consult the Cassandra operator logs to figure out why restore failed, and fix the problem.

See:

- [Example 1: Corrupted backup tar file cc-0 not reporting properly](#)
- [Example 2: Corrupted backup tar file cc-1 not reporting properly](#)

Example 1: Corrupted backup tar file cc-0 not reporting properly

In this example, restoration was started with `apicup`:

```
./apicup subsys exec restore 1583268283534609276
```

1. View the initial status of restore job and restore pod:

```
kubect1 get jobs | grep restore
restore-xs85r                0/1          38s          38s

kn get pods | grep restore
restore-xs85r-st554          0/2          Init:1/2     0            92s
```

2. Note that the restore completes, according to the job and pod status:

```
kubect1 get pods | grep restore
restore-xs85r-st554          0/2          Completed    0            3m43s

kubect1 get jobs | grep restore
restore-xs85r                1/1          2m51s        4m1s
```

3. Observe, however, that the `apicup` restore command output includes the following failure status:

```
./apicup subsys exec mgmt restore 1583268283534609276
Cluster      Namespace   Backup Name      Backup Retrieval Timeout(hrs)  Status
rdd94fb4a21-apiconnect-cc  niharns1     1583268283534609276  24                               Started

Cluster      Namespace   Backup Name      Status
rdd94fb4a21-apiconnect-cc  niharns1     1583268283534609276  Restore Failed: Backup retrieve checks failed
```

4. Check the value of `ClusterRestoreStatus` in the Custom Resource. Note that `ClusterRestoreStatus` is marked as `Restore Failed`.

```
kubect1 get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: 'Restore Failed: Backup retrieve checks failed'
```

5. Check the Cassandra operator pod logs and see errors:

```
time="2020-03-04T04:30:33Z" level=error msg="Restore Failed: Backup retrieve checks failed with an error: \n
Pod name: rdd94fb4a21-apiconnect-cc-0\n
Error: \n
gzip: stdin: not in gzip format\n
tar: Child returned status 1\n
tar: Error is not recoverable: exiting now\n
retrieveCheck 0: RetrieveCheck of backup file 1583268283534609276-0-3.tar.gz FAILED\n
**** [ Wed Mar  4 04:30:32 UTC 2020 ] retrieveCheck 0: Retrieve process END ****\n\n"
time="2020-03-04T04:30:33Z" level=info msg="Updating status to Restore Failed: Backup retrieve checks failed"
```

**Explanation:** Even though restore job says it completed, Cassandra restore never completed due to corrupt backup tar file. The restore job believes it reached completion, but in this case the error logging is incorrect. Therefore, you must check `ClusterRestoreStatus` in `CassandraCluster` Custom Resource to see if restore really completed. Note that `CassandraClusterRestore` status does not state exactly why restore failed, hence you must look at the Cassandra operator pod logs.

Table 1. Limitation with error reporting when Cassandra tar file for cc-0 is corrupted

Expected flow	Actual flow
1. Cassandra operator detects that cc-0 backup is corrupt	1. Cassandra operator detects that cc-0 backup is corrupt
2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code>	2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code>
3. Operator passes an error message back to restore init container (cassandra-restore).	3. Operator <i>does not</i> pass an error message back to restore init container (cassandra-restore)
4. <code>apicup</code> restore command line should exit with an error message on why restore init container (cassandra-restore) failed. Restore pod status should be <code>Init:Error</code> .	4. Restore pod moves forward and executes other init container and upgrade containers.
5. Restore job is marked as Failed	5. Restore pod is marked as completed

6. **Workaround:** The Cassandra cluster remains up and running as restore process failed internally, so you must review the Cassandra operator logs to figure out why exactly Cassandra restore failed. In this case, Cassandra restore failed due to corrupt backup tar file. Locate and use a non-corrupted Cassandra backup.

Example 2: Corrupted backup tar file cc-1 not reporting properly

In this example, restoration was started with `apicup`:

```
./apicup subsys exec restore 1583268283534609276
```

1. View the initial status of restore job and restore pod

```
kubectl get jobs | grep restore
restore-v7nvt 0/1 81s 81s

kubectl get pods | grep restore
restore-v7nvt-2jdzq 0/2 Init:0/2 0 101s
```

2. Continue to view pods, and observe that the restore pod is stuck.

The restore pod first init container (cassandra-restore), which performs the Cassandra restore, is marked as complete. The second init container is stuck waiting for the Cassandra cluster to become healthy.

```
kn get pods | grep restore
restore-v7nvt-2jdzq 0/2 Init:1/2 0 4m8s
```

The status `Init:1/2` means that first init container completed, but the process is waiting on second init container to finish.

3. Note that the output from the `apicup` restore command gives a status of `Restore Failed`:

```
./apicup subsys exec mgmt restore 1583268283534609276
Cluster          Namespace      Backup Name      Backup Retrieval Timeout(hrs)  Status
rdd94fb4a21-apiconnect-cc  niharns1      1583268283534609276  24                               Started

Cluster          Namespace      Backup Name      Status
rdd94fb4a21-apiconnect-cc  niharns1      1583268283534609276  Restore Failed: Backup retrieve checks failed
```

4. View the `ClusterRestoreStatus` in the Cassandra Cluster Custom Resource:

```
kn get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: 'Restore Failed: Backup retrieve checks failed'
```

5. View the Cassandra operator pod logs:

```
time="2020-03-04T05:04:18Z" level=error msg="Restore Failed: Backup retrieve checks failed with an error: \n Pod name: rdd94fb4a21-apiconnect-cc-1\n Error: \ngzip: stdin: not in gzip format\n tar: Child returned status 1\n tar: Error is not recoverable: exiting now\n retrieveCheck 0: RetrieveCheck of backup file 1583268283534609276-1-3.tar.gz FAILED\n **** [ Wed Mar 4 05:04:18 UTC 2020 ] retrieveCheck 0: Retrieve process END ****\n\n time="2020-03-04T05:04:18Z" level=info msg="Updating status to Restore Failed: Backup retrieve checks failed"
```

6. View the Cassandra pod status:

```
rdd94fb4a21-apiconnect-cc-0      0/1      Running      3      11h
rdd94fb4a21-apiconnect-cc-1      1/1      Running      2      11h
rdd94fb4a21-apiconnect-cc-2      1/1      Running      2      11h
```

**Explanation:** In this scenario, the restore process is stuck waiting for the Cassandra cluster to become healthy. Cassandra pod cc-0 is in a non-ready state. Due to a known limitation with error logging, the only way to accurately determine whether the restore is complete is to view `ClusterRestoreStatus` in `CassandraCluster` Custom Resource. Since `CassandraClusterRestore` status does not state exactly why the restore failed, you must look at Cassandra operator pod logs.

Table 2. Limitation with error reporting when corrupted Cassandra backup tar file cc-1 is corrupted

Expected flow	Actual flow
1. Cassandra operator detects that cc-1 backup is corrupt	1. Cassandra operator detects that cc-1 backup is corrupt
2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code> .	2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code> .
3. Operator cleans up the restore process content and makes sure that Cassandra cluster is healthy using the existing data.	3. Operator <i>does not</i> perform any cleanup and <i>does not</i> makes sure that the Cassandra cluster is in a healthy state
4. Operator passes an error message back to restore init container (cassandra-restore) to indicate that the restore has failed.	4. Operator <i>does not</i> pass an error message back to restore init container (cassandra-restore) to indicate that the restore has failed.



Expected flow	Actual flow
5. The <code>apicup</code> restore command line should exit with a proper error message on why restore init container (cassandra-restore) failed. The restore pod status should be <code>Init:Error</code> . The restore job should be marked as Failed.	5. Restore pod moves forward from the init container but becomes stuck on second init container.

7. **Workaround:**

- a. Since the Cassandra cluster is in a degraded state, and the Cassandra pod cc-0 is in a non-ready state (0/1), run the following command to bring the Cassandra cluster up and running.

```
kubect1 exec -it <cassandra-pod-X> -n <namespace> -- sh -c
'rm -rf /var/db/.restore && rm -rf /var/db/restore/*'
```

Note that you must run this command sequentially on all Cassandra pods, starting with cc-0. Make sure that the cc-x status is (1/1), before running this command on the next pod (cc-x+1).

- b. Review the Cassandra operator logs to determine why the Cassandra restore failed. In this scenario, since the Cassandra restore failed due to a corrupt backup tar file, the solution is to choose a non-corrupted Cassandra backup.

Note that this Cassandra restore issue applies not only specifically to a corrupted backup tar file, but also to any of the restore issues which may happen on Cassandra-x+1 pods (x is a numerical value starting with 0) and can leave previous Cassandra pods in a non-ready state.

## Frequently asked questions

Where can I find the status of the restore process?

See: [Cassandra restore logging](#)

In what cases can I re-run the restore process on existing installations?

The answer depends on which stage of the restore failed. If the restore process failed because of a corrupted tar file, you can re-initiate the restore process using a different backup ID.

In what cases I need to redeploy a whole new cluster before re-attempting a failed restore?

If the restore failed due to space allocation problems or any other reasons except corrupted backup tar, a complete new installation is needed. You must fix the problem reported by the restore process.

What action to take if I get the error message: Not enough space to download the tar file?

You must re-install the management subsystem, allocating sufficient disk space. Please look at system requirements section for recommended disk sizes.

What if the downloaded backup tar file is corrupted?

Use a different backup. Run backup tar integrity checks prior to attempting the restore.

If the backup tar integrity succeeds in your local system but restore process is failing, gather the required logs and contact IBM Support.

What if there is not enough space to perform restore using the downloaded backup tar file?

You must re-install the management subsystem, allocating sufficient disk space. See [Error: insufficient disk space](#).

What if the restore job finishes but I still don't see any restored data?

See [Known limitation: failed restores not reporting properly](#).

What if the restore process remains stuck on the health check status for a very long time?

This might be due to a known limitation. See [Example 2: Corrupted backup tar file cc-1 not reporting properly](#).

What if the restore process fails on Lur-upgrade-job and Apim-upgrade-job containers?

Run the restore process again. If the error persists, contact IBM Support.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Backing up and restoring the Developer Portal in a Kubernetes environment

How to backup and restore your Developer Portal service in your Kubernetes environment.

### About this task

It is strongly recommended that you configure the backup parameters for your portal service during installation. If you did not do so when you installed API Connect in your runtime environment, you must configure the backups for the Developer Portal before performing an upgrade. These backups can then be used to restore the Developer Portal if required. When your Developer Portal subsystem is running, you can also make on-demand backups by using the command line.

The default Developer Portal backup schedule is once every 24 hours, but the schedule can be changed in the backup settings. The Developer Portal saves all system and site backups locally, and also saves them remotely based on the configured SFTP and s3 settings.

The local backups are automatically maintained so that the latest three backups of each site and of the system are kept, and older backups are removed. This maintenance means that the Developer Portal retains the latest three backups for each site and for the system however old they are, but there is no deletion of the old backups on the remote server. If a site is deleted, then all of the local backups for that site are also deleted, as otherwise the backup volume might become full of old site backups. For remote backups, you can configure a retention policy on your remote server to remove the old backup files as required.

These instructions cover the following backup and restore actions:

- [How to configure the location and timing of your backups](#)
- [How to run on-demand backups](#)
- [How to restore a Developer Portal service](#)

Note:

- The backup and restore procedures are the same for both Kubernetes and VMware environments.
- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.



- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.
- The backup secret is a Kubernetes secret that contains your username and password for your backup database (sftp/s3). Only password-based authentication is supported for sftp and s3, not authentication based on public certificates and private keys. Password-based authentication for s3 requires that you generate an access key and secret. For example:
  - IBM (Cloud Object Storage): [Service credentials](#).
  - AWS: [Managing access keys](#).

## Procedure

- How to configure the location and timing of your backups
  1. Open your API Connect installation project directory.
  2. Run the following commands to set the location and timing of your backups:

```
apicup subsys set pt1 site-backup-host mybackuphost.com
apicup subsys set pt1 site-backup-port 22
apicup subsys set pt1 site-backup-auth-user mybackupauthusername
apicup subsys set pt1 site-backup-auth-pass mybackupauthpassword
apicup subsys set pt1 site-backup-path /site-backups
apicup subsys set pt1 site-backup-protocol sftp
apicup subsys set pt1 site-backup-schedule "0 2 * * *"
```

The backup parameters are detailed in the following table.

Table 1. Portal backup parameters

Parameter	Description
<code>site-backup-host</code>	The fully qualified domain name of the backup server, in lowercase only. Ensure that the Kubernetes nodes can access this host. If using object storage, enter <i>Endpoint/Region</i> . (The / character between the endpoint and region is required for the object storage setting.)
<code>site-backup-port</code>	The port for the protocol to connect to the <code>site-backup-host</code> . Defaults to 22 if not explicitly set. The backup port is not required for object storage.
<code>site-backup-auth-user</code>	The user name for the server specified in <code>site-backup-host</code> . If using object storage, the user name is the S3 Secret Key ID.
<code>site-backup-auth-pass</code>	The password for the server specified in <code>site-backup-host</code> . If using object storage, the password is the S3 Secret Access Key parameter. The password is stored in Base64 encoded format, and must not be edited directly in the <code>apiconnect-up.yml</code> file.
<code>site-backup-path</code>	The full path to the directory where the backup files are stored. For object storage, the path can be set to the <i>bucket</i> value or the <i>bucket/subfolder</i> value.
<code>site-backup-protocol</code>	The protocol that is used to communicate with your remote backup endpoint. Specify one of the following values: <ul style="list-style-type: none"> <li>• <code>sftp</code> - for secure file transfer protocol</li> <li>• <code>objstore</code> - for S3 compatible object storage</li> </ul> The default protocol is <code>sftp</code> . Note: The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code> .
<code>site-backup-schedule</code>	The schedule for how often automatic Portal backups are run. The format for the schedule is any valid cron string, as follows: <pre>* * * * * - - - - -                   +----- day of week (0 - 6) (Sunday=0)       +----- month (1 - 12)     +----- day of month (1 - 31)   +----- hour (0 - 23) +----- min (0 - 59)</pre> For example: <code>30 22 * * 1</code> performs backups at 10:30 pm on Mondays. The default backup schedule is <code>0 2 * * *</code> (runs every day at 2 am). The timezone for backups is UTC.

3. Optional: At any time you can view the current Portal subsystem values by running the following command:

```
apicup subsys get pt1
```

where `pt1` is the name that you assigned to your Portal service. The output from this command lists all of the subsystem settings, including backup, and indicates whether there are any errors. You must fix any errors before continuing.

4. Validate the installation with the new backup parameters by running the following command:

```
apicup subsys get pt1 --validate
```

where `pt1` is the name that you assigned to your Portal service. The output from this command lists all of the subsystem settings, including backup, and indicates whether the values are valid or invalid. You must correct any invalid values before continuing.

5. Activate the backup settings by running the following command:

```
apicup subsys install pt1
```

where `pt1` is the name that you assigned to your Portal service.

- How to run on-demand backups

You can make on-demand backups of your Developer Portal system, sites, and complete service, by running the following `exec` commands in your API Connect installation project directory.

- To backup the Portal system configuration (and not the sites), run the following command:

```
apicup subsys exec pt1 backup-system
```

- To backup a specific Portal site or all sites, run the following command:

```
apicup subsys exec pt1 backup-site arg
```

where `arg` is a specific site UUID or URL, or to backup all sites use the argument `installed`.

- To backup the entire Portal service (the system and all installed sites), run the following command:

```
apicup subsys exec pt1 backup
```

Where `pt1` is the name that you assigned to your Portal service.

Note: To restore a Developer Portal service, you need backups of both the Portal system and the installed sites.

- How to restore a Developer Portal service

You can restore your Developer Portal service by using the backups that exist on your remote server, by running the following `exec` commands in your API Connect installation project directory. Note that before you can run these commands, you must have configured your remote backup server details, and have valid backups of both the Portal system and the installed sites (see sections [How to configure the location and timing of your backups](#) and [How to run on-demand backups](#)).

Note:

- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- Restoration requires a functioning Developer Portal. In a disaster recovery scenario, you might need to reinstall the Developer Portal subsystem before you can restore the backed-up data. To reinstall, refer to [Installing the Developer Portal subsystem into a Kubernetes environment](#).
- To see what Portal system and site backups that you currently have on your remote backup server, run the following command. (This command may be useful when performing some of the following commands.)

```
apicup subsys exec pt1 list-backups remote
```

- To see what restore actions are performed when the `apicup subsys exec pt1 restore-all` command is run, you can use the following command:

```
apicup subsys exec pt1 restore-all dry backup_from
```

where `backup_from` can be `now`, so the latest backup file that is available is used. Or you can specify a timestamp in the format of `YYYYMMDD.HHMMSS` to retrieve the nearest backup file to a specified time, searching backwards from the timestamp given.

Note: From IBM® API Connect Version 2018.4.1.17, the timestamp format changed to `YYYYMMDD.HHMMSS`. For Version 2018.4.1.16 and earlier, the timestamp format is `YYYY-MM-DD`

`HH:MM:SS`.

- To restore your Portal service, run the following command:

```
apicup subsys exec pt1 restore-all run backup_from
```

where `backup_from` can be `now`, so the latest backup file that is available is used. Or you can specify a timestamp in the format of `YYYYMMDD.HHMMSS` to retrieve the nearest backup file to a specified time, searching backwards from the timestamp given. This command executes the portal restore process, which downloads the system backup and all of the site backups from the remote server, and installs them within the Portal stack. This process will then restore the system configuration from the found backup, and restore all of the sites.

Note: Note that if a backed up site is already installed on the current stack, then the site is reinstalled by using the backup (the site is overwritten by the backup).

If there are multiple sites to restore, then these sites are queued for restoring. You can track the restoration process within the Portal `www admin` logs.

- To view the list of installed and restoring sites, run the following command:

```
apicup subsys exec pt1 list-sites sites
```

Note that any sites that are pending restore and still in the queue do not appear in this list.

- To restore just the Portal system configuration, run the following command:

```
apicup subsys exec pt1 restore-system arg
```

where `arg` can be the system backup `tgz` file name, to restore the system by using the specified backup file, or use the argument `latest` to restore the system by using the latest backup file on the remote server. Note that if Portal system configuration information exists on the current stack, running this command will overwrite that configuration.

- To restore just the Portal sites, run the following command:

```
apicup subsys exec pt1 restore-site arg
```

where `arg` can be the site backup `tgz` file name or URL, to restore a particular site by using the specified backup file (or the latest backup file found if using a URL). Or you can use the argument `all` to restore all of the Portal sites that have backup files on the remote server. Note that if any of the backed up sites are already installed on the current stack, then they are reinstalled by using the backup (the sites are overwritten by the backup).

## What to do next

---

For a full list of the Developer Portal `exec` commands, see [List of the Developer Portal exec commands](#).

- [List of the Developer Portal exec commands](#)

You can use the `exec` commands to backup, restore, and list some of the environment details of your Developer Portal service.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## List of the Developer Portal `exec` commands

You can use the `exec` commands to backup, restore, and list some of the environment details of your Developer Portal service.

The following Developer Portal `exec` commands can be run in your API Connect installation project directory, where `pt1` is the name of your Portal service.

Tip: Running `exec` commands leaves completed jobs and pods on your subsystem. Therefore, it is recommended that you regularly delete those jobs and pods that have completed successfully, and whose logs are not required.

```

$ apicup subsys exec ptl backup-system
Backup the portal system to the remote server

$ apicup subsys exec ptl backup-site <arg>
Backup a portal site to the remote server.
valid args:
  site uuid/url - will backup that site only to the remote server
  installed    - will backup all installed sites to the remote server

$ apicup subsys exec ptl backup
Backup the portal system and all its sites to the remote server

$ apicup subsys exec ptl list-backups <arg>
List the portal backups either locally or on the remote server.
valid args:
  local    - will list backups present on the pod's filesystem
  remote   - will list backups on the remote server

$ apicup subsys exec ptl list-platforms
List the platforms that exist on the portal

$ apicup subsys exec ptl list-sites <arg>
List the installed portal sites
valid args:
  sites      - list only the sites
  platforms  - list the sites and their associated platform

$ apicup subsys exec ptl restore-system <arg>
Restore the portal system from the remote server
valid args:
  system backup-1234.tgz - restores the system from the given file which should exist on the pod's local filesystem or
the remote server
  latest                 - restores from latest system backup found from either local filesystem or remote server

$ apicup subsys exec ptl restore-site <arg>
Restore a portal site
valid args:
  site-1234.tgz - restore the site using this filename which should can be copied from the output of the list backups
command
  myportal.com  - find the latest backup for this URL (locally or remotely) and restore it.
  all           - restore all sites found on the remote server
NOTES:
If you specify a site TGZ / URL, the command runs restore_site with -f so overwrites any existing site
If you specify 'all', any installed sites will be reinstalled from the found backup

$ apicup subsys exec ptl restore-all <arg1> <arg2>
Restore the portal system and all backed up sites from the remote server
valid arg1:
  dry - executes a dry-run of the command, returning the actions that will be performed
  run - executes the command, restoring and replacing the system files and all sites. Any existing sites will be
reinstalled
valid arg2:
  now - use the latest backups available
  <TIMESTAMP> - in 'YYYYMMDD.HHMMSS' format, specify a timestamp to retrieve the backup from. The nearest backup,
searching backwards from this timestamp, will be used.

Note: IBM API Connect 2018.x was EOS after 30 April 2023. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x and later product documentation.

```

---

## Backing up and restoring the analytics database

The analytics database can be backed up and restored from an S3 repository. S3 compatible object storage is required, for example, IBM Cloud Object Storage.

### Before you begin

If you configure Analytics for offloading and your endpoint requires a particular certificate, add trust certificates to the Analytics subsystem before attempting to back up or restore the Analytics database. For information on adding certificates to Analytics for back-up and restore, see [Adding certificates for Analytics for back-up and restore](#).

Tip: The [Procedure](#) section includes examples that illustrate how the back-up and restore commands work.

### About this task

These commands apply to OVA, pure Kubernetes, and IBM Cloud Private (ICP) deployments. To back up and restore the analytics database, you will need S3 compatible object storage. Backups are created on an as-needed basis and cannot be automated in API Connect.

Important: All arguments are required. Replace an argument with empty quotes ("") to use the default setting.

Command	Values/Definition
---------	-------------------

Command	Values/Definition
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; create-s3-repo</pre>	<p>Create the S3 repository to store analytics backups. These settings are identical to those used when creating a repository in Elasticsearch. The arguments are:</p> <ul style="list-style-type: none"> <li>• REPO_NAME - Name of repository to be created.</li> <li>• REGION - The region where bucket is located. Defaults to US Standard.</li> <li>• BUCKET - The name of the bucket to be used for snapshots. Note: Analytics backup and restore supports virtual-host style bucket access, such as <code>bucket.s3-example.com</code>, but does not support path style bucket access such as <code>s3-example.com/bucket</code>.</li> <li>• ENDPOINT - The endpoint to the S3 API.</li> <li>• ACCESS_KEY - The access key to use for authentication.</li> <li>• SECRET_KEY - The secret key to use for authentication.</li> <li>• BASEPATH - The path name within the bucket where backup information is stored. The default is the root path. For S3 storage, you must include the port using the following format: <code>BASEPATH:PORT</code>.</li> <li>• COMPRESS_TRUE_FALSE - Default is true. Determines whether metadata files are stored in compressed format.</li> <li>• CHUNK_SIZE_GB - Default is 1GB. Large files can be stored as chunks when the snapshot is created. This setting specifies the size of the chunks as GB, MB, or KB.</li> <li>• SERVER_SIDE_ENCRYPTION_TRUE_FALSE - Default is false. Determines whether files are encrypted. When set to true, files are encrypted on the server side using AES256.</li> </ul> <p>If the <code>create-s3-repo</code> command results in the following error, complete the steps in <a href="#">Adding certificates for Analytics for back-up and restore</a> to add the certificate to the Analytics subsystem and then run the command again:</p> <pre>PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</pre>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; list-repos</pre>	<p>List repositories for analytics backups. There are no arguments or defaults for the <code>list-repos</code> command.</p>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; delete-repo</pre>	<p>Delete specified analytics backup repository. The argument is:</p> <ul style="list-style-type: none"> <li>• REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> </ul> <p>Examples:</p> <ul style="list-style-type: none"> <li>• If only one repo exists: <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; delete-repo ""</code></li> <li>• If multiple repos exist: <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; delete-repo REPO_NAME</code></li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; backup</pre>	<p>Perform a backup of analytics data on an as-needed basis.</p> <ul style="list-style-type: none"> <li>• INDICES - Default is <code>all</code> - All indices will be backed up. The INDICES arguments can consist of a series of index names, a single argument of comma-separated indices, or one or more keywords. Multiple keywords must be comma-separated. Values with whitespace must be enclosed in double-quotes. If no indices are specified, then all indices will be backed up. The keywords are mapped to indices or aliases in Elasticsearch. The keywords are as follows: <ul style="list-style-type: none"> <li>◦ <code>all</code> - Backup all data.</li> <li>◦ <code>apievents</code> - Backup all analytics data.</li> <li>◦ <code>ui</code> - Backup all UI visualizations and dashboards.</li> <li>◦ <code>config</code> - Backup all configuration information, such as retention period.</li> </ul> </li> <li>• BACKUP_NAME - Enter the name of the backup.</li> <li>• REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> <li>• IGNORE_UNAVAILABLE_TRUE_FALSE - Default is false. If false, the restore will fail if an index is missing. If true, missing indices will be skipped and the restore will continue.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; list-backups</pre>	<p>List analytics backups per S3 repo. The argument is:</p> <ul style="list-style-type: none"> <li>• REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; details-backup</pre>	<p>Get details on specified analytics backup The arguments are:</p> <ul style="list-style-type: none"> <li>• BACKUP_NAME - Defaults to <code>backup-&lt;indices&gt;-&lt;time date&gt;</code> if not explicitly set. The name must be all lowercase. For example, if the backup command was run as <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; backup ui apievents</code> then the backup name would be <code>backup-ui-apievents-2018-10-10t16:04:25z</code>. If more than three indices are specified, the backup name is truncated to <code>backup-&lt;time date&gt;</code>.</li> <li>• REPO_NAME - Sets the name of the repository where the backups will be stored. Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; delete-backup</pre>	<p>Delete specified analytics backup The arguments are:</p> <ul style="list-style-type: none"> <li>• BACKUP_NAME - Enter the name of the backup to be deleted. You can view the names of backups using <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; list-backups</code>.</li> <li>• REPO_NAME - Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> </ul>

Command	Values/Definition
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; restore</pre>	<p>Restore analytics data</p> <ul style="list-style-type: none"> <li>INDICES - Default is <code>all</code> - All indices will be restored. The INDICES arguments can consist of a series of index names, a single argument of comma-separated indices, or one or more keywords. Multiple keywords must be comma-separated. Values with whitespace must be enclosed in double-quotes. If no indices are specified, then all indices will be backed up. The keywords are mapped to indices or aliases in Elasticsearch. The keywords are as follows: <ul style="list-style-type: none"> <li><code>all</code> - Restore all data.</li> <li><code>apievents</code> - Restore all analytics data.</li> <li><code>ui</code> - Restore all UI visualizations and dashboards.</li> <li><code>config</code> - Restore all configuration information, such as retention period.</li> </ul> </li> <li>BACKUP_NAME - Enter the name of the backup to be restored.</li> <li>REPO_NAME - Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> <li>IGNORE_UNAVAILABLE_TRUE_FALSE - Default is false. If false, the restore will fail if an index is missing. If true, missing indices will be skipped and the restore will continue.</li> <li>OVERRIDE_TRUE_FALSE - Default is false. If set to true, then the restore operation will override existing indices.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; restore-status</pre>	<p>Displays analytics status for determining the progress of the restore process. Note: The <code>restore-status</code> command is available in Version 2018.4.1.7 or later.</p>

Attention: Naming conventions for indices and back ups follow the Elasticsearch rules:

- Lowercase only
- Cannot include \, /, \*, ?, ", <, >, |, ` (space character), ,, #
- Colons (:) are not supported in 7.0+ (indices prior to 7.0 could contain a colon)
- Cannot start with -, \_, +
- Cannot be . or ..
- 255 byte limit (multi-byte characters will reach the 255 limit sooner)

## Procedure

This section contains example commands with output. The name of the analytics subsystem in the examples is *analytics*.

- How to create a backup
  1. Create the S3 repository. The example creates a repository with the following values:
    - REPO\_NAME - myrepo
    - REGION - US
    - BUCKET - bucket
    - ENDPOINT - myrepo.s3repo.com
    - ACCESS\_KEY - access\_key
    - SECRET\_KEY - secret\_key
    - BASEPATH - my\_folder
    - COMPRESS\_TRUE\_FALSE - "" uses default of true
    - CHUNK\_SIZE\_GB - "" uses default of 1GB.
    - SERVER\_SIDE\_ENCRYPTION\_TRUE\_FALSE - "" sets to the default of false.

```
apicup subsys exec analytics create-s3-repo myrepo US bucket myrepo.s3repo.com access_key secret_key my_folder "" ""
""
OUTPUT:
Creating repository myrepo. List repos to see if creation completed, or check logs for errors.
```

Note:

With virtual host style repositories, the Analytics backup and restore process connects to the hostname formed by combining the values for `BUCKET` and `ENDPOINT`. For example, using the values given in this step, the host is `bucket.myrepo.s3repo.com`.

2. List the repositories. For example:

```
apicup subsys exec analytics list-repos
OUTPUT:
Name      Repo Type  Bucket   BasePath  Region  Endpoint                Chunk Size  Compress  Server Side
Encryption
myrepo    s3        bucket   my_folder US        myrepo.s3repo.com      1gb         true
```

3. Create a backup with the following values:
  - INDICES - all (or "" for the default of all)
  - BACKUP\_NAME - mybackup
  - REPO\_NAME - myrepo (use "" if only one repo exists)
  - IGNORE\_UNAVAILABLE\_TRUE\_FALSE - "" sets to the default of false.

```
apicup subsys exec analytics backup all mybackup myrepo ""
OUTPUT:
Successfully created backup mybackup.
```

4. List backups in the repo named *myrepo*.

```
apicup subsys exec analytics list-backups myrepo
OUTPUT:
Name      Start Time                End Time                State
mybackup  2019-02-20T17:05:56.415Z  2019-02-20T17:06:09.833Z  SUCCESS
```

5. Display details for a backup named *mybackup* in the repo named *myrepo*.

```
apicup subsys exec analytics details-backup mybackup myrepo
OUTPUT:
Backup Name: mybackup
State: SUCCESS
Failures:
Shards Failed: 0
Shards Successful: 13
Start Time: 2019-02-20T17:05:56.415Z
End Time: 2019-02-20T17:06:09.833Z
Version: 5.6.8
Indices: .export-status, apic-api-2019.02.19-1, apic-api-2019.02.20-000002, .apic-config, .kibana-6
```

6. Delete a backup named *mybackup* in the repo named *myrepo*.

```
apicup subsys exec analytics delete-backup mybackup myrepo
OUTPUT:
Deleting backup mybackup. List backups to see the status, or check logs for errors.
```

- How to restore a backup

Note: Restoration requires a functioning Analytics subsystem. In a disaster recovery scenario, you might need to install the Analytics subsystem before you can restore the backed-up data. To install, refer to [Installing the Analytics subsystem into a Kubernetes environment](#).

1. Restoring a backup

- INDICES - all (or "")
- BACKUP\_NAME - mybackup
- REPO\_NAME - myrepo (or "")
- IGNORE\_UNAVAILABLE\_TRUE\_FALSE - "" sets to the default of false.
- OVERRIDE\_TRUE\_FALSE - true

```
apicup subsys exec analytics restore all mybackup myrepo "" true
```

2. Check the status of the restore process. Enter the following command:

```
apicup subsys exec analytics restore-status
```

Note: The `restore-status` command is available in Version 2018.4.1.7 and later.

Following is example output:

Status	Active Primary Shards	Active Shards	Initializing Shards	Unassigned Shards
green	104	312	0	0

The restore process is successful when the status is green and there are no unassigned shards and no initializing shards. Note that the results are different if you are running in dev mode, or in standard mode with less than three nodes. For dev mode or standard mode with less than three nodes, the restore process will be finished when the initializing shards drops to 0. However, the status will remain yellow and unassigned shards will not drop to 0.

- Troubleshooting

Look for information in the `apicup` logs:

- Locate the analytics-operator pod with: `kubectl get pods`
- To view the logs, run: `kubectl logs <analytics-operator-pod>`

- [Adding certificates for Analytics for back-up and restore](#)

If you offload IBM API Connect Analytics data to an endpoint that is secured with a self-signed or private certificate, add the certificate to the Analytics subsystem to ensure connectivity with the endpoint during back-up and restore operations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding certificates for Analytics for back-up and restore

If you offload IBM® API Connect Analytics data to an endpoint that is secured with a self-signed or private certificate, add the certificate to the Analytics subsystem to ensure connectivity with the endpoint during back-up and restore operations.

### About this task

If you connect to an endpoint without using the required self-signed or private certificate, the `create-s3-repo` command results in the following error:

```
PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
```

Resolve the error by adding the endpoint's certificate to the Analytics subsystem as explained in the following steps.

### Procedure

1. Obtain trust certificates that allow a client to connect securely to the offload endpoint.  
The certificate files should be PEM encoded. There will typically be one or two trust certificates required to complete the secure connection.
2. Create a separate file for each of the trust certificates, naming them cert1.pem and cert2.pem.
3. Verify that you can connect to the endpoint:
  - a. If you have multiple files, combine the certificates into a single file to use for validation by running the following command:

```
cat cert1.pem cert2.pem > certificates.pem
```

- b. Execute the following cURL command to connect to the endpoint using the certificates:

```
curl https://endpoint:port --cacert certificates.pem
```

A certificate verification failure indicates that the certificates obtained in step 1 failed to establish trust. Return to step 1 and try again. You cannot proceed to step 4 until the certificates are successfully verified.

- c. When the certificate verification is successful, proceed to step 4.
4. Create a Kubernetes secret to contain the certificates, and then apply it to the cluster where the Analytics subsystem runs.
- a. Log in to a server that has kubectl access to the Kubernetes cluster where API Connect Analytics is deployed.  
For OVA deployments, you must log in to one of the running Analytics VMs as a user with kubectl access.

- b. Encode each of the certificate files from step 2 in base64.

There are several ways to encode the file. If `cat` and `base64` are available, then you can run the following command to encode each file:

```
cat cert1.pem | base64 -w 0
```

Copy the output for each `cat` command so that you can paste it in the next step.

- c. Create a Kubernetes secret and add the certificate.

- i. Create a YAML file to contain the secret.

```
apiVersion: v1
kind: Secret
metadata:
  # Change value of name to be whatever you want the secret to be called
  name: analytics-br-cert
data:
  cert1.pem: output of base64 encoded cert1.pem
  cert2.pem: output of base64 encoded cert2.pem if required
```

- ii. In the file's `data` section, create a field for each certificate, and paste the corresponding encoded certificate as shown in the example. The value is a single line, so you do not need to enclose it in quotation marks.

- iii. Save the file.

- d. Update the cluster with the new certificate.

Run the following command to update the cluster:

```
kubectl apply -f analytics-br-cert.yaml -n your_namespace
```

where:

- `analytics-br-cert.yaml` is the secret's file name.
- `your_namespace` is the name of your deployment's namespace. The namespace is only needed for Kubernetes deployments.  
For OVA deployments, you can omit the entire `-n your_namespace` parameter from the command. By default, kubectl is already configured to match the namespace where your Analytics resources are deployed.

5. Update the Analytics subsystem so that it can use the new secrets.

- a. Create an `extra-values.yaml` file that specifies the names of your certificate and environment variables secret files.

Include the following lines in the `extra-values.yaml` file, making sure to use the name of your Kubernetes secret (the value of the `name` field in the `metadata` section of the secret, which might not match the file's name):

```
apic-analytics-storage:
  backupSecretCerts: analytics-br-cert
```

- b. Update the Analytics subsystem.

- i. Log in to the server where you run `apicup`, and navigate to the project directory.  
In Kubernetes deployments this is probably the same server that you used for step 1, but for OVA deployments it is a different server.
- ii. Run the following `apicup` commands to reinstall the Analytics subsystem using the configuration settings in the `extra-values.yaml` file.

```
apicup subsys set analytics extra-values-file=~/.extra-values.yaml
apicup subsys install analytics
```

6. Verify that the certificates were added in the storage logs.

Log in to the host node and use `sudo` to run the following command:

```
kubectl logs -n namespace analytics-storage-pod
```

For OVA deployments, you can omit the namespace parameter and value.

The following confirmation message displays when the pod starts up:

```
Adding /etc/velox/backup-certs/ca-analytics-backup.pem contents to the system keystore for backup and restore.
Certificate cert1.pem was added to keystore
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using APICUP to reconfigure

You can use APICUP to change configuration of a subsystem after completion of initial installation.



The Install Assist utility (APICUP) is used to configure and complete initial installation of all subsystems, in both Kubernetes and VMware deployments. After initial installation, you can use the same `apicup` commands to reconfigure some of the settings from your existing deployment without having to completely reinstall the subsystem.

For example, if you did not configure optional features such as backup/restore or logging during initial installation, you can configure them later.

Note: Both Kubernetes and VMware deployments include subsystem settings that cannot be reconfigured without a complete redeployment. For example, for the Management Service these settings include hosts, endpoints, interfaces (`public-iface`, `traffic-iface`), IP ranges of the Kubernetes pod and the service networks (`k8s-pod-network`, `k8s-service-network`), and backup volume size `cassandra-volume-size-gb`. Before reconfiguring, review the installation instructions for your subsystem to determine which settings are optional during initial installation and thus able to be updated later. See the section [Installing and upgrading on Kubernetes](#).

For both initial installation and reconfiguration, you specify configuration values by using `apicup subsys set` commands, and then activate them with `apicup subsys install`.

```
apicup subsys set <subsystem_name> <parameter_name>=<parameter_value>
```

```
apicup subsys install <subsystem_name>
```

For example, the following commands configure backup for the management subsystem `mgmt`, either during initial install or after initial install:

```
apicup subsys set mgmt cassandra-backup-auth-user=MyUsername cassandra-backup-auth-pass=MyPassword
apicup subsys set mgmt cassandra-backup-host=<hostname>
apicup subsys set mgmt cassandra-backup-path=</backups>
apicup subsys set mgmt cassandra-backup-port=<22>
apicup subsys set mgmt cassandra-backup-protocol=<sftp, objstore>
apicup subsys set mgmt cassandra-backup-schedule=<"0 0 * * *">
apicup subsys set mgmt cassandra-max-memory-gb=16 cassandra-cluster-size=3
apicup subsys set mgmt cassandra-volume-size-gb=<50>
apicup subsys set mgmt create-crd=<true>
apicup subsys set mgmt external-cassandra-host=<hostname>
```

```
apicup subsys install mgmt
```

On initial installation, the APICUP utility installs a Helm chart along with configuration values contained in `apiconnect-up.yml`. On subsequent invocations of `apicup subsys install`, APICUP checks to see if a saved combination of Helm chart and configuration values exists. If it does, APICUP checks to see if the configuration values have been updated, and if so, then a Helm update is triggered. If the combination of Helm chart and values is unchanged, then the subsystem is not changed. In this way, you can reconfigure as needed without having to completely redeploy and reinstall the subsystem.

- [Converting installation mode](#)  
You can switch an API Connect installation between `dev` mode and `standard` mode.
- [Increasing the memory allocation for the management database](#)  
You can increase the maximum memory allocation for the management database in a running deployment.
- [Enabling Analytics ingestion-only on Kubernetes](#)  
Enable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` configuration setting.
- [Disabling Analytics ingestion-only on Kubernetes](#)  
Disable the ingestion-only configuration by redeploying the IBM API Connect Analytics subsystem with the `ingestion-only` value set to false.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Converting installation mode

You can switch an API Connect installation between `dev` mode and `standard` mode.

### About this task

The supported installation mode options are: `dev` and `standard`. Mode is set for each subsystem type: Management, Analytics, Developer Portal, Gateway Service.

- Development mode (`dev`) deploys a subsystem with the scale of one; a single node, non-HA subsystem. The recommended use for `dev` mode is for development and testing.  
Note: While it is possible to use `dev` mode installations in a production deployment, it is not recommended. The `dev` mode does not provide the failover resiliency nor the downtime guarantee that is needed to maintain high availability (HA) production deployments.
- The `standard` mode deploys in HA mode for a production environment. The `standard` mode is supported for installation of production environments that consist of three or more nodes. The `standard` mode is not supported for less than three nodes.
- If not explicitly set, the installation mode defaults to `dev` (development). To specify `standard` mode:

```
apicup subsys set [subsys-name] mode=standard
```

Note: Prior to Version 2018.4.1.4, the default mode was `standard`.

- If you installed in `standard` mode, but have less than three nodes, you can convert to `dev` mode. You must convert each subsystem.
- If you installed in `dev` mode, you can convert to `standard` mode. You must convert each subsystem.

Complete the following steps to determine your mode and, if necessary, convert the mode:

### Procedure

1. Review the output from the command: `apicup subsys get <subsystem_name>`.



For example, `apicup subsys get mgmt:`

- VMware (OVA) environment output:

```
Appliance settings
=====
Name                               Value                               Description
----                               -
.
.
mode                               standard
.
.
```

- Kubernetes output:

```
Kubernetes settings
=====
Name                               Value                               Errors
----                               -
.
.
mode                               standard
.
.
```

2. If you need to convert the mode, following the appropriate instructions:

- [Converting dev mode to standard mode on VMware](#)
- [Converting dev mode to standard mode on Kubernetes](#)
- [Converting standard mode to dev mode on VMware](#)
- [Converting standard mode to dev mode on Kubernetes](#)
- [Converting dev mode to standard mode on Kubernetes](#)  
You can convert your API Connect installation on Kubernetes from `dev` mode to `standard` mode.
- [Converting standard mode to dev mode on Kubernetes](#)  
You can convert your API Connect installation on Kubernetes from `standard` mode to `dev` mode.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Converting dev mode to standard mode on Kubernetes

You can convert your API Connect installation on Kubernetes from `dev` mode to `standard` mode.

### About this task

To review the installation modes, see [Converting installation mode](#).

Make sure the Kubernetes cluster where the subsystem is deployed has at least 3 nodes. .

### Procedure

1. Verify that the Kubernetes cluster where the subsystem is deployed has at least 3 nodes.  
Deploying a subsystem in `standard` mode on a cluster with less than 3 nodes is not supported.

2. For each subsystem in your deployment:

- a. Use `apicup` to convert to `standard` mode.

```
apicup subsys set <subsys> mode=standard
```

- b. Run the command:

```
apicup subsys install <subsys>
```

This command propagates the configuration change into the Helm charts that are deployed and updated by the `apicup` installer.

To review installation instructions for each subsystem on Kubernetes, see:

- [Installing the Management subsystem into a Kubernetes environment](#)
- [Installing the Analytics subsystem into a Kubernetes environment](#)
- [Installing the Developer Portal subsystem into a Kubernetes environment](#)

3. To verify your installation mode, use `apicup subsys get <subsystem_name>`.

For example, `apicup subsys get mgmt:`

```
Kubernetes settings
=====
```

Name	Value	Errors
mode	standard	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Converting standard mode to dev mode on Kubernetes

You can convert your API Connect installation on Kubernetes from **standard** mode to **dev** mode.

### About this task

To review the installation modes, see [Converting installation mode](#).

### Procedure

Complete the following steps for each subsystem:

1. Use the following `apicup` command to convert to Dev mode:

```
# ./apicup subsys set [subsys-name] mode=dev
# ./apicup subsys install [subsys-name]
```

2. Select one of the following actions, based on your deployment:

- If your Kubernetes deployment has two or more nodes, you do not have to remove replicaset. Continue to step 3
- If your Kubernetes deployment has only one node, you must remove replicaset. Complete the following:
  - Find the Helm release name and namespace for the subsystem that you are switching to dev mode:

```
helm ls
```

Sample output:

NAME	REVISION	UPDATED	STATUS	CHART
NAMESPACE				
r04bac65d2f	1	Mon Feb 11 14:51:08 2019	DEPLOYED	dynamic-gateway-service-1.0.16
apiconnect				
r1001d56a76	1	Mon Feb 11 14:50:46 2019	DEPLOYED	apic-analytics-2.0.0
apiconnect				
r14613f8fca	1	Mon Feb 11 14:50:26 2019	DEPLOYED	apic-portal-2.0.0
apiconnect				
raeb5658fc0	1	Mon Feb 11 14:51:27 2019	DEPLOYED	dynamic-gateway-service-1.0.16
apiconnect				
rbcb2382b5e	1	Mon Feb 11 14:50:06 2019	DEPLOYED	apiconnect-2.0.0
apiconnect				
rf19e62b277	1	Mon Feb 11 14:50:03 2019	DEPLOYED	cassandra-operator-1.0.0
apiconnect				

- Use the following command to delete the replicaset for that release in its namespace:

```
kubectl delete rs -n <namespace> -l release=<release_name>
```

For example, using the previous sample output:

```
kubectl delete rs -n apiconnect -l release=rbcb2382b5e
```

3. To review your installation mode, use `apicup subsys get`

`<subsystem_name>`.

For example, `apicup subsys get mgmt`:

```
Kubernetes settings
=====
```

Name	Value	Errors
mode	dev	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Increasing the memory allocation for the management database

You can increase the maximum memory allocation for the management database in a running deployment.

### Before you begin

---

When increasing the memory allocation, complete a manual backup of the management database and Portal subsystem just prior to changing the allocation. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management database in a Kubernetes environment](#) and [Backing up and restoring the Developer Portal](#).

### About this task

---

Starting with API Connect version 2018.4.1.3, you can increase the maximum memory allocation for the management database in a running deployment. This procedure works for both Kubernetes and OVA environments. All the pods in the API Connect cluster must be up and running to increase the maximum memory value in a running deployment.

### Procedure

---

1. Ensure all pods are up and running using the `kubectl get pods` command.
2. Use the following `apicup` command to increase the maximum memory:

```
apicup subsys set <MANAGEMENT_SUBSYS_NAME> cassandra-max-memory-gb <new_value>
```

3. Reinstall the management subsystem to push the change to the cluster. This may take several minutes as each pod updates.

```
apicup subsys install <MANAGEMENT_SUBSYS_NAME>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Enabling Analytics ingestion-only on Kubernetes

Enable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` configuration setting.

### Before you begin

---

Only complete this task if you want to offload all analytics data and disable the Analytics feature in the user interface. If you want to retain the data or allow users to access the Analytics user interface, skip this task and instead see [Configuring analytics offload for API Connect](#).

### About this task

---

If you choose to offload all analytics data to third-party services and have no need to retain the data in API Connect, you can configure the Analytics service with the ingestion-only setting. If you offload data and then separately disable the Analytics UI, the associated components are still deployed and require system resources. Configuring Analytics for ingestion-only removes unused Analytics components (such as analytics-storage and analytics-client) from the topology and only deploys the components required for offloading data (such as analytics-ingestion and analytics-mq-kafka). The reduced topology requires less CPU, memory, and storage.

### Procedure

---

1. Remove the Analytics service from your API Connect deployment by completing the following steps.
  - a. Unassociate the Analytics service from all gateway services as explained in step 7 in the topic, [Associating an analytics service with a gateway service](#).
  - b. Unregister the Analytics service from Cloud Manager.  
You can unregister the service from the Topology page: click the Options menu next to the service name and select Delete. For information on registering services, see [Registering an analytics service](#).
2. Create an extra values file to specify the URL of the offload endpoint where all analytics data will be routed.  
You can choose the name for the file, but you must use `.yaml` as the file-name extension. Format the file as shown in the following example, making sure to include required settings for the third-party system.

```
apic-analytics-ingestion:  
  outputOffload: |-  
    elasticsearch {  
      hosts => "http://offload_endpoint:port"  
      index => "api-call"  
    }  
  }
```

For information about configuring analytics offloading, see [Configuring analytics offload for API Connect](#).

3. Run the following `apicup` commands to reinstall the Analytics subsystem using the configuration settings in the extra values file:

```
apicup subsys set analyt ingestion-only=true
apicup subsys set analyt extra-values-file=extra-values.yaml
apicup subsys install analyt
```

4. Add the new Analytics service to your API Connect deployment by completing the following steps.
  - a. Register the new Analytics service with the Cloud Manager.

Attention: When you register an Analytics service that is configured for ingestion-only, you must use the ingestion endpoint and the ingestion TLS profile instead of the client endpoint and TLS profile.

For information on registering a service with the Cloud Manager, see [Registering an analytics service](#).
  - b. Associate the Analytics service with a gateway service as explained in [Associating an analytics service with a gateway service](#).

## Results

---

All analytics data is routed directly to the offloading endpoint, the Analytics UI is disabled, and no data is retained in API Connect.

In addition, the following Analytics configuration settings will not be validated or used when `ingestion-only` is set to enabled:

- `coordinating-max-memory-gb`
- `data-max-memory-gb`
- `data-storage-size-gb`
- `master-max-memory-gb`
- `master-storage-size-gb`
- `analytics-client`
- `es-storage-class`

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Disabling Analytics ingestion-only on Kubernetes

Disable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` value set to false.

### About this task

---

When you previously configured Analytics for ingestion-only, you deployed only a subset of the Analytics components. If you want to disable the ingestion-only setting, redeploy the subsystem to ensure that all components are available.

### Procedure

---

1. Remove the Analytics service from your API Connect deployment by completing the following steps.
  - a. Unassociate the Analytics service from all gateway services as explained in step 7 in the topic, [Associating an analytics service with a gateway service](#).
  - b. Unregister the Analytics service from Cloud Manager.

You can unregister the service from the Topology page: click the Options menu next to the service name and select Delete. For information on registering services, see [Registering an analytics service](#).
2. Generate the `analytics-client-ingress` certificate (if you initially created this certificate, it was removed when you deployed using the ingestion-only configuration).

For information, see [Working with certificates](#).
3. Run the following `apicup` commands to install the complete Analytics subsystem:

```
apicup subsys set analyt ingestion-only=false
apicup subsys install analyt
```
4. Add the new Analytics service to your API Connect deployment by completing the following steps.
  - a. Register the new Analytics service with the Cloud Manager.

For information on registering a service with the Cloud Manager, see [Registering an analytics service](#).
  - b. Associate the Analytics service with a gateway service as explained in [Associating an analytics service with a gateway service](#).

## Results

---

The Analytics subsystem is deployed with all components, the Analytics user interface is enabled, and the deployment is configured to route analytics data from the Gateway to API Connect, where it is stored. If you want to offload analytics data, configure offloading as explained in [Configuring analytics offload for API Connect](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring maximum size of client requests to the Management subsystem

You can configure the maximum allowed size of the client request body for requests made to the Management subsystem.

## About this task

---

The default maximum size is 8 megabytes. You can increase this value.

Increasing the setting can be useful if you get errors when using the CLI to import a large API. Example error:

```
HTTP/1.1 413 Request Entity Too Large
```

## Procedure

---

1. Add an entry to an extra-values-file:

```
juhu:
  clientMaxBodySize: "12m"
```

Note that the default value is "8m".

For information on working with extra-values-files, see [Creating an extra values file in a Kubernetes environment](#)

2. Run `apicup` to update the settings in the deployed Management subsystem.

```
apicup subsys install <management-subsystem>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Increasing memory allocation for Cassandra operator on OpenShift

If the Cassandra operator pods restart due to out of memory errors, you can increase the memory allocation for the Cassandra operator.

## About this task

---

On OpenShift 4.6, Cassandra operator pods can sometimes restart due to out of memory errors. The problem shows up as a number of **Restarts** for the Cassandra pod in the output from:

```
kubectl get pods -n <namespace>
```

You can see the out of memory errors in the output of `dmesg` on the worker node. For example:

```
[256264.442631] Memory cgroup out of memory: Killed process 1899490 (cop) total-vm:2427012kB, anon-rss:244376kB, file-
rss:23748kB, shmem-rss:0kB, UID:108
[256266.030417] oom_reaper: reaped process 1899490 (cop), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
[256830.705886] cop invoked oom-killer: gfp_mask=0x6000c0 (GFP_KERNEL), nodemask=(null), order=0, oom_score_adj=-998
```

The problem has been seen on API Connect v2018.4.1.13 and later, running on OpenShift 4.6.9, with underlying Kubernetes 1.19 and Helm version 3.4.2.

To fix the problem:

## Procedure

---

1. Get the current deployment:

```
oc get deployment -n <namespace>
```

2. Edit the Cassandra deployment:

```
oc edit deployment <r9c835a18ba-cassandra-operator> -n <namespace>
```

3. Change the memory limits and memory requests value to **512Mb**:

```
resources:
  limits:
    cpu: 100m
    memory: 512Mi
  requests:
    cpu: 100m
    memory: 512Mi
```

4. Save the edits.
5. Verify that the Cassandra pod has been recreated. Check the output of:

```
kubectl get pods -n <namespace>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Setting rate limits for public APIs on the management service for a Kubernetes environment

Describes the procedure for setting a rate limit for public APIs on the management service. Rate limits provide protection from DDoS (distributed denial of service) attacks.

## Before you begin

---

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated. These instructions assume you have a working Kubernetes environment with the kubectl command-line tool installed, and that you understand how to manage Kubernetes. For more information, see <https://kubernetes.io>.

## About this task

---

Rate limits can be set for public APIs on the management service. Rate limits on APIs help provide protection from DDoS (distributed denial of service) attacks. Without a rate limit, API calls from public APIs are unlimited.

The rate limit configuration requires that the header contains the actual client IP address. Any load balancer or proxy (for example, HAProxy) that is installed in front of the management service must be configured to pass the actual client IP address.

This procedure must be performed on a running API Connect deployment.

This feature is available in API Connect versions 2018.4.1.1 and higher.

Rate limits are calculated as requests per seconds per client.

## Procedure

---

- Configure the ingress-nginx-ingress-controller to pass the real client IP address:
  1. Set the `use-proxy-protocol` parameter to `true` by entering the following command:

```
kubectl edit ConfigMap -n kube-system ingress-nginx-ingress-controller
```

2. In the ConfigMap, set the following: `use-proxy-protocol: "true"`

- Set a rate limit:

1. Add the following entries to an extra-values-file:

```
juhu:
  rateLimitPerClient: 10
  limitRequestOption: "burst=10 nodelay"
```

In this example, the first option sets the rate to 10 requests per second (10r/s). The second option allows 10 requests boosted without delay within < 1 second. You can customize as needed.

The `rateLimitPerClient` property sets `rate`, and `limitRequestOption` sets `[burst=number] [nodelay | delay=number]` in the following nginx configuration:

```
limit_req_zone key zone=name:size rate=rate;
limit_req zone=name [burst=number] [nodelay | delay=number];
```

- Note that `zone` has been pre-defined and can't be configured. For details, see the nginx.org doc [Module ngx\\_http\\_limit\\_req\\_module](#).
- If you don't have an extra-values-file, you can create a new one. See [Creating an extra values file in a Kubernetes environment](#).

2. Run `apicup` to update the settings in the deployed Management subsystem.

```
apicup subsys install <management-subsystem>
```

Note: If the rate limit has been reached on the management subsystem, the client will get an HTTP error: **429 Too Many Requests**.

3. Validate that the juhu pod has restarted by listing the pods:

```
kubectl get pods -n <namespace> | grep juhu
```

4. Check the AGE column to ensure a new juhu pod has started.
5. If the older juhu pod is still running, delete it with the following command:

```
kubectl delete pods -n <namespace> <old-juhu-pod>
```

- Disable a rate limit:

1. Validate that the juhu pod has restarted by listing the pods:

```
kubectl get pods -n <namespace> | grep juhu
```

2. Check the AGE column to ensure a new juhu pod has started.
3. If the older juhu pod is still running, delete it with the following command:

```
kubectl delete pods -n <namespace> <old-juhu-pod>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Enabling high performance peering for DataPower API Gateway on Kubernetes

Enabling high performance peering on DataPower API Gateway deployments optimizes API transaction performance.

## About this task

---

You can configure DataPower API Gateways to have dedicated `gateway-peering` objects for the API Connect Gateway Service, subscription data, and rate-limit data. By default, the gateway service is configured to share a single `gateway-peering` object for all three. In some scenarios, use of a single object can impact API performance. Enabling of high performance peering is highly recommended.

High performance peering is available in API Connect Version 2018.4.1.7 and later.

Use the procedure in this topic to reconfigure an existing API Connect deployment to enable high performance peering.

Important: Enabling high performance peering on an existing deployment requires an outage to API transaction processing during configuration.

- If you are upgrading the Gateway from Version 2018.4.1.6 to version 2018.4.1.7 or later, complete the upgrade first and then use the instructions in this topic to reconfigure peering. For upgrade instructions, see [Upgrading API Connect in a Kubernetes environment](#).
- If you are deploying DataPower API Connect externally to Kubernetes, such as part of a physical or virtual DataPower appliance, do not use this topic. Instead, follow the configuration instructions in [Sample configuration for multiple peering objects on gateway services external to Kubernetes](#)

It is important that all gateway instances are down at the same time, in order for dynamic reconfiguration to take place. In this case, dynamic reconfiguration is achieved by scaling down gateway services, using `apicup` to install, then scaling up the gateway services. To review the DataPower Gateway dynamic reconfiguration process, see [Dynamically re-registering and reconfiguring a Gateway service in a Kubernetes deployment](#).

## Procedure

---

1. Set the high performance peering option in the `apicup` gateway configuration.

```
apicup subsys set gwy enable-high-performance-peering="true"
```

You must run `apicup` in the same project directory as where the gateway was originally configured using `apicup`.

2. Obtain the name of the StatefulSet, and scale the cluster down to a single gateway.

```
kubectl get statefulset
kubectl scale statefulset r600e4a32ef-dynamic-gateway-service --replicas=1
```

3. Use `apicup` to redeploy the gateway pod.

```
apicup subsys install gwy
```

4. Optional: Log in to the DataPower CLI and use the `show gateway-peering-status` status to confirm that multiple `gateway-peering` objects have been configured.

Confirm that the status shows separate entries for API Connect Gateway Service (`gwd`), API Gateway rate limit (`rate-limit`), and API Gateway subscriptions (`subs`). For example:

```
idg# sw apiconnect
idg[apiconnect]# config
Global configuration mode
idg[apiconnect](config)# show gateway-peering-status
```

Address	Configuration name	Pending updates	Replication offset	Link status	Primary
192.168.0.127	gwd	0	0	ok	yes
192.168.0.127	rate-limit	0	0	ok	yes
192.168.0.127	subs	0	0	ok	yes

5. When the first gateway pod is marked ready, scale the StatefulSet to the original number of gateways. For example:

```
kubectl scale statefulset r600e4a32ef-dynamic-gateway-service --replicas=3
```

Be sure to scale up the same StatefulSet that you scaled down in Step 2.

6. Wait for management server heartbeat to detect that reconfiguration is required.

Heart beat processing usually detects the need for reconfiguration within 5 minutes, but the process may take a few more minutes to begin. When the `show api-collection` command returns data, the reconfiguration has begun. The length of time to required to reach completion depends on the number of catalogs, products, APIs, subscriptions, etc.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Dynamically re-registering and reconfiguring a Gateway service in a Kubernetes deployment

In API Connect, Gateway services do not persist their configuration settings by default. Instead, the master configuration is stored on the Management server and the *Dynamic Reregistration and Reconfiguration* (DRR) mechanism resynchronizes configuration data when needed. The DRR process is used when proper High Availability/Disaster Recovery (HA/DR) is not configured, or if a manual resynchronization is required.

If a Gateway service is not configured properly for resiliency and is restarted, the gateways in the Gateway service will lose the configuration from the Management server. Configuration data from the Management server is maintained on the gateway service according to the gateway peering configuration on the gateways.

---

## Preventing the loss of configuration data

For high availability in production environments, use a minimum of three gateways in the Gateway service.

---

## Forcing re-population of the configuration data

To repopulate the configuration data, complete the steps in the following section to trigger an outage-less DRR.

---

## Triggering an outage-less DRR

To force a Dynamic Reregistration and Reconfiguration across a peer group from within the gateway service without requiring a restart, complete the following steps using the CLI. This process flushes the primary peering object configured for the apic-gw-service. This task can only be performed with CLI, and requires the default admin ID for running the `diag` command.

1. Access the gateway service:

- For a virtual gateway, run the following command to access the DataPower CLI:

```
kubectl attach -it <pod_name>
```

- For a physical gateway, use `ssh` to connect to the gateway service.

2. Switch to the API Connect application:

```
switch <APIC_APP_DOMAIN>
```

3. Show the current gateway-peering-status:

```
show gateway-peering-status
```

4. If the server that you connected to is not primary for the apic-gw-service gateway-peering object, force this server to become primary:

```
config; gateway-peering-switch-primary <gateway-peering-name-for-apic-gw-service>;
```

5. Flush the apic-gw-service gateway-peering object:

```
diag; gateway-peering-flush <gateway-peering-name-for-apic-gw-service>; exit
```

When prompted, confirm the operation.

6. Disable and then re-enable the apic-gw-service object for every member of the gateway service:

```
config; apic-gw-service; admin-state disabled; exit  
apic-gw-service; admin-state enabled; exit; exit
```

7. Confirm that the apic-gw-service was flushed and is now waiting for gateway service registration:

Look in the debug log target for the apic-gw-service, located in `logtemp:///` and check for a message similar to the following example:

```
20200618T232339.332Z [0x88e000d7][apic-gw-service][notice]  
apic-gw-service(default): tid(2653): Waiting for gateway service registration.
```

The DRR will be triggered by the next arrival of a webhook event. Starting with API Connect Version 2018.4.1.2, the Management server sends a heartbeat to the gateway at five-minute intervals, prompting the gateway to check whether it has lost its configuration and if so, trigger a DRR. In Version 2018.4.1.8 and later, if the Management server has previously marked a Catalog or cloud as being unavailable for a particular gateway service, a successful heartbeat triggers synchronization for that Catalog or cloud on that gateway service.

8. If you attached to a virtual gateway, you can detach by pressing Ctrl+P and then pressing Ctrl+Q.

9. (Optional) Force a BAU webhook event to be sent from API Manager to trigger the DRR:

If you do not want to wait for the Management server to send a heartbeat, you can trigger the DRR manually by completing the following steps:

- a. Open the API Manager.
- b. Open the Catalog.
- c. Click Settings > Edit.
- d. Update the Summary field with some text so that it is modified.
- e. Click Save.

When the event is received, the DRR is initiated.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## Monitoring and logging a Kubernetes deployment

You can do health checks to determine program status, and can gather logs for use in troubleshooting.

- [Checking cluster health on Kubernetes](#)  
You can use `apicup` to check the health of the API Connect clusters in your Kubernetes deployment.
- [Determining status of a cluster on Kubernetes](#)  
You can determine the health of an API Connect cluster on Kubernetes.
- [Obtaining simple health check data of Developer Portal sites by using a REST API call](#)  
Call a simple health check API from your external load balancer to dynamically determine whether a specific Developer Portal site in a cluster is working. This API call can be used by a load balancer to help determine where to route traffic.
- [Gathering logs for a Kubernetes environment](#)  
The `generate_postmortem.sh` script gathers all logs for troubleshooting and diagnostics.
- [Changing logging levels](#)  
You can enable logging for entry and exit trace and for large payloads for `apim-v2` pods.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Checking cluster health on Kubernetes

You can use `apicup` to check the health of the API Connect clusters in your Kubernetes deployment.

Note: The `apicup` health check command is available on Version 2018.4.1.6 or later. For deployments on Version 2018.4.1.5 or earlier, use the instructions on [Determining status of a cluster on Kubernetes](#).

The `health-check` command checks a number of criteria to determine the health of their cluster. When all criteria are successfully met, the command displays no output, and exits with a status of 0. When one or more criteria are not met, the command stops processing and displays a message with the failure and exits with a status of 1.

The command takes no arguments. The only option is the `--verbose` flag. If `--verbose` is set the command prints all checks that were performed.

The `health-check` is run against the namespace that is specified in `apiconnect-up.yaml`.

The health check is run on the specified subsystem. For each expected Helm release, the command:

- Verifies that the deployed Chart version matches the `apicup` version.
- Validates that all Deployments, ReplicaSets, and DaemonSets are fulfilled.

Syntax:

Usage:

```
apicup subsys health-check <SUBSYS> [flags]
```

Flags:

```
-h, --help                help for health-check
--kubeconfig string      (optional) absolute path to the kubeconfig file (default "/Users/<username>/.kube/config")
-v, --verbose            Verbose output
```

Global Flags:

```
--accept-license  Accept the license for API Connect
--debug           Enable debug logging
```

Example usage, for a subsystem named `mgmt`:

```
../apicup subsys health-check mgmt
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Determining status of a cluster on Kubernetes

You can determine the health of an API Connect cluster on Kubernetes.

---

### About this task

At any time, you can review the status of the API Connect cluster. You should review the status before and after doing configuration tasks such as upgrading.

Note: For Version 2018.4.1.6 or later, you can also use `apicup subsys health-check`. See [Checking cluster health on Kubernetes](#)  
Do not use these instructions on VMware. See [Determining status of a cluster on VMware](#).

## Procedure

1. Check that the subsystem Helm release was successful.

The release names are generated off the subsystem chart plus namespace. First determine the release name for the subsystem by running `helm ls --namespace <namespace>`. Next use the following chart to determine which release corresponds to which subsystem.

Subsystem	Helm release
management	apiconnect-2.0.0, cassandra-operator-1.0.0
analytics	apic-analytics-2.0.0
portal	apic-portal-2.0.0

Example:

```
$ helm ls --namespace e2etest
NAME REVISION UPDATED STATUS CHART APP
VERSION NAMESPACE
r3b3275c7fb 1 Mon Apr 22 09:20:13 2019 DEPLOYED dynamic-gateway-service-1.0.23 1.0
demo-namespace
r7430a60641 1 Mon Apr 22 08:44:57 2019 DEPLOYED apiconnect-2.0.0
demo-namespace
r85ba975aee 1 Mon Apr 22 09:23:06 2019 DEPLOYED apic-portal-2.0.0
demo-namespace
re5c7f54039 1 Mon Apr 22 09:20:32 2019 DEPLOYED apic-analytics-2.0.0
demo-namespace
rf99c951215 1 Mon Apr 22 08:44:45 2019 DEPLOYED cassandra-operator-1.0.0 1.0.1
demo-namespace
```

In the previous example, the release types are as follows:

Value in NAME column	Release type
r7430a60641 and rf99c951215	Management
r85ba975aee	Portal
re5c7f54039	Analytics

If any release is not **DEPLOYED**, open a ticket with IBM Support, to obtain assistance with troubleshooting the release.

2. Check that the deployed charts are the correct version.

This check needs to be performed on only one node. See the previous steps to determine the Helm release name(s) for the subsystem type.

- Version 2018.4.1.4 or later  
Run the command `helm get [release-name] | grep productVersion | head -1`

```
# helm get values apiconnect | grep productVersion | head -1
productVersion: 2018.4.1.4
```

If the previous command returns `productVersion: ""`, the deployed charts are for a version prior to 2018.4.1.4. See the following instructions.

- Version 2018.4.1.3 or earlier
  - Management subsystem

- Run `helm get apiconnect | grep apiconnect/apim[: ] | head -1`. Compare output to the following table to determine chart `productVersion`:

Image output	Chart productVersion
image: "apiconnect/apim:2018.4.1-223-5e98505"	2018.4.1.0
image: "apiconnect/apim:2018.4.1-267-7151665"	2018.4.1.1
image: "apiconnect/apim:2018.4.1-339-0d1c113"	2018.4.1.2
image: "apiconnect/apim:2018.4.1-391-e1df735"	2018.4.1.3

- Run `helm get cassandra-operator | grep apiconnect/cassandra-operator[: ] | head -1`. Compare output to the following table to determine chart `productVersion`.

Image output	Chart productVersion
image: "image: apiconnect/cassandra-operator:2018-10-27-11-34-27-b30560eb3775c558f8438abb8e084ed97ca8a34f"	2018.4.1.0
image: "image: apiconnect/cassandra-operator:2018-11-13-14-39-43-cac5b0e471f011258f38b92c6959e4c30bc21b08"	2018.4.1.1
image: "image: apiconnect/cassandra-operator:2019-01-07-17-52-36-a571b8b69248e7afd5aac06ca5196ffeb8fd7875"	2018.4.1.2
image: "image: apiconnect/cassandra-operator:2019-02-19-01-57-00-77482ccb867d08277e9985a3a75cb86385720523"	2018.4.1.3

- Portal subsystem  
Run `helm get apic-portal | grep apiconnect/portal-web[: ] | head -1`. Compare output to the following table to determine chart `productVersion`.

Image output	Chart productVersion
image: apiconnect/portal-web:2018.4.1-385444eb87824c93e42c85412af8088eac8c7bb9-335	2018.4.1.0
image: apiconnect/portal-web:2018.4.1-ea9a86af8cf6f9e241f29f9a7ad860991e017d29-514	2018.4.1.1
image: apiconnect/portal-web:2018.4.1-8e3c67a1ec93f9eda7411286b75e19becaab0fb0-773	2018.4.1.2

Image output	Chart productVersion
image: apiconnect/portal-web:2018.4.1-31b0250446fd1660cc4622f44b7f8317391c536d-948	2018.4.1.3

- Analytics subsystem  
Run `helm get apic-analytics | grep apiconnect/analytics-client[:]`  
| `head -1`. Compare output to the following table to determine chart `productVersion`:

Image output	Chart productVersion
image: "apiconnect/analytics-client:2018-10-18-17-32-08-ba640482b2659c37905271d6a318fa0fc902aebd"	2018.4.1.0
image: "apiconnect/analytics-client:2018-11-21-11-19-56-ba640482b2659c37905271d6a318fa0fc902aebd"	2018.4.1.1
image: "apiconnect/analytics-client:2019-01-07-15-16-59-9f8c1b7bb4a8f9a0bf9529bab166f752821b62e4"	2018.4.1.2
image: "apiconnect/analytics-client:2019-02-12-12-52-40-700d61ffb1c2ff2c9860d6d75f1620fbceea64d6"	2018.4.1.3

3. Review the results of the previous step, and ensure that the product version matches the `apicup` version and the appliance version. If the chart version is wrong it is likely that either the wrong version of Install Assist was used for `apicup subsys install`, or an old `--plan-dir` was passed to the install command. Verify the correct version of `apicup` is being used and try to re-install by using `apicup subsy install`. It is safe to run the command multiple times.

4. Use `kubect1` to check the deployments and validate that the number of deployments matches for `DESIRED`, `CURRENT`, and `AVAILABLE`. If the numbers don't match, it is possible that one of the pods is still starting, or is having problems. Another possibility is that Kubernetes encountered problems trying to scale down an old replica set. Check for replica-sets that are for the same deployment and have a number count greater than 0 for `DESIRED`. Note that the replica-sets for a deployment start with the same name.

You can use `kubect1 get rs` to try to determine why the replica-set is not scaling down. For example, the following output shows a cluster in a non-healthy state.

```
# kubect1 get deployments
NAME                                DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
apiconnect-a7s-proxy                3        3        3            1          2h
apiconnect-apim-v2                  3        3        3            3          2h
apiconnect-client-dl-srv            2        3        2            1          2h
apiconnect-juhu                     3        4        2            2          2h
apiconnect-ldap                     3        4        2            2          2h
apiconnect-lur-v2                   3        4        2            2          2h
apiconnect-ui                        2        3        2            1          2h
cassandra-operator-cassandra-operator 1        1        1            1          2h

# kubect1 get rs
NAME                                DESIRED  CURRENT  READY  AGE
apiconnect-a7s-proxy-56c9f975c      3        3        1      9m
apiconnect-apim-v2-55d77dff65       3        3        3      2h
apiconnect-client-dl-srv-56cfd98dd5  2        2        0      9m
apiconnect-client-dl-srv-6bc7d48477  1        1        1      2h
apiconnect-juhu-744db85f             2        2        0      9m
apiconnect-juhu-744f9cbc89           2        2        2      2h
apiconnect-ldap-6c855d4b             2        2        2      2h
apiconnect-ldap-d599cb954           2        2        0      9m
apiconnect-lur-v2-79556d69c          2        2        0      9m
apiconnect-lur-v2-7f97c7b8b5         2        2        2      2h
apiconnect-ui-69c8df97               2        2        0      9m
apiconnect-ui-84c97657c5             1        1        1      2h
cassandra-operator-cassandra-operator 1        1        1      2h
```

5. Check that the StatefulSets are fulfilled. For each subsystem run `kubect1 get sts -n <name-space>` and ensure that for each entry, the numerical values match for `DESIRED`, `CURRENT`, and `READY`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Obtaining simple health check data of Developer Portal sites by using a REST API call

Call a simple health check API from your external load balancer to dynamically determine whether a specific Developer Portal site in a cluster is working. This API call can be used by a load balancer to help determine where to route traffic.

### About this task

You can use the site health REST API to determine whether a particular Developer Portal site is running. The site health API returns the current system time of the site if both the database and web server are running. This API is fast and puts no load on the system, so it is ideal for use with load balancers to help them determine where to route traffic.

### Procedure

To call the site health REST API, append `/health` to the end of your Developer Portal site URL in your web browser, as follows:

`site_url/health`

Where `site_url` is the URL of the Developer Portal site that you want to check.

If both the database and web server of the site are running, the web browser returns the current system time. For example:

```
1511367695
```

If either the database or the web server of the site is not running, the web browser returns an error that the site can't be reached.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Gathering logs for a Kubernetes environment

The `generate_postmortem.sh` script gathers all logs for troubleshooting and diagnostics.

### Before you begin

---

Review the log rotation recommendations in [Requirements for deploying IBM API Connect into a Kubernetes environment](#).

### About this task

---

When contacting IBM Support, several logs are required to assist in troubleshooting or diagnostics. The `generate_postmortem.sh` script gathers all the required logs using a single shell script. The `generate_postmortem.sh` script must be executed from the `apicup` project directory. You can download the script directly to the `apicup` project directory or set the environment variable `APICUP_PROJECT_PATH` by entering the following command: `export APICUP_PROJECT_PATH="/path/to/directory"`

Note: This article refers to third-party software and documentation that IBM does not control. As such, the software may change and this information may become outdated.

### Procedure

---

1. Install the `kubectl` command line interface for Kubernetes, if it is not already installed.
2. Configure `kubectl` to the Kubernetes cluster by entering the following commands:

```
rm -fr $HOME/.kube
mkdir -p $HOME/.kube
scp root@{kubernetes_master_host}:/etc/kubernetes/admin.conf $HOME/.kube/config
```

3. Install the correct version of `helm` and connect it to the Kubernetes cluster.

```
rm -fr $HOME/.helm
curl -s https://raw.githubusercontent.com/helm/helm/master/scripts/get | bash -s -- --version v2.8.2
helm init --client-only
```

4. Download the `generate_postmortem.sh` script. It will be downloaded to the `apicup` project directory. Enter the following command:

```
curl -o generate_postmortem.sh https://raw.githubusercontent.com/ibm-apiconnect/v2018-postmortem/master/generate_postmortem.sh
```

5. Add execute permissions to the `generate_postmortem.sh` script. Enter the following command:

```
chmod +x generate_postmortem.sh
```

6. Run the `postmortem` tool from the `apicup` project directory. Change directories to the `apicup` project directory, or set the environment variable `APICUP_PROJECT_PATH`. Enter the following command:

```
./generate_postmortem.sh
```

7. (Optional) Currently, log generation is supported for Gateway and Portal subsystems. Specify either `gateway` or `portal`, or all subsystems. Enter the following arguments:

```
All (generates logs for both Portal and Gateway subsystems): ./generate_postmortem.sh --diagnostic-all
Gateway: ./generate_postmortem.sh --diagnostic-gateway
Portal: ./generate_postmortem.sh --diagnostic-portal
```

8. (Optional) If there are errors when running the `generate_postmortem.sh` script, enter the following command:

```
./generate_postmortem.sh --debug &>debug.log
```

Note:

- Offboarding the logs to an external server is the recommended best practice for long term storage of log data.
- If the logs do not cover far enough back in time, or conversely if they are too large, adjust the log retention size. For further information, see: <https://docs.docker.com/config/containers/logging/json-file/>.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing logging levels

You can enable logging for entry and exit trace and for large payloads for apim-v2 pods.

### About this task

For apim-v2 pods in Version 2018.4.1.9 or later, the default logging settings do not include entry and exit trace and logging of large payloads.

Logging of large payloads is typically needed if you are logging `apim:webhookPayload` or other processes that log large variables.

You can also set the debug level without needing to restart the pods.

You can enable the settings through either the APIM deployment YAML file, the toolkit, or the REST APIs. Note, you must use the toolkit or REST API to set the debug level without restarting the pods.

### Procedure

- Updating the settings in the APIM deployment YAML file

Note: This action causes a pod restart.

- The DEBUG variable works the same as prior releases.

```
- env:
  - name: DEBUG
    value: audit,bhendi:error,bhendi:probe,bhendi:flags,bhendi:audit,bhendi:webhookAudit,bhendi:cassandra-
transactions,apim:server,apim:error,apim:routes:*,apim:routesc:*,apim:oidc,apim:oidc:*,apim:webhook:audit,apim:taskma
nager:info
```

To enable entry and exit trace, add `trace:*` to the start of the debug string:

```
- env:
  - name: DEBUG
    value: trace:*,audit,bhendi:error,bhendi:probe,bhendi:flags,bhendi:audit,bhendi:webhookAudit,bhendi:cassandra-
transactions,apim:server,apim:error,apim:routes:*,apim:routesc:*,apim:oidc,apim:oidc:*,apim:webhook:audit,apim:taskma
nager:info
```

- To enable large object logging:

```
- env:
  - name: VELOX_LOG_FULL_PAYLOAD
    value: "true"
```

- Updating the settings by using the toolkit

Note: This action does not cause a pod restart. Also, this action is for a single APIM pod environment only.

```
apic log-spec:get
apic log-spec:update LOG_SPEC_FILE
```

Where the spec file contains:

```
{"specification": "apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error", "large_objects": true}
```

- Updating the settings by using the REST API

Note: This action does not cause a pod restart. Also, this action is for a single APIM pod environment only.

```
GET /cloud/api/log-spec
PUT /cloud/api/log-spec
```

For example, to set debug level and large objects logging:

```
curl -k https://172.16.140.212:3003/api/cloud/log-spec -X PUT -H "Authorization: Bearer $BEARER"
-H "Accept: application/json" -d '{"specification":
"apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error", "large_objects": true}'
-H "Content-Type: application/json"
```

The command responds with:

```
{
  "specification": "apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error",
  "large_objects": true,
  "message": "Successfully changed the log specification on all apim-v2 pods. Success on IP(s): 172.16.140.212,
172.16.140.213"
}
```

Note: The response might include the following warning, which can be ignored:

```
WARNING: Could not change the log specification on all apim-v2 pods. Success on IP(s): 4.5.6.7 Failed on IP(s): 1.2.3.4
```

To view the REST APIs for logging, go to [API Connect REST APIs](#), and select IBM API Connect Platform - Cloud Management API 2.0.0 reference\_2\_Resource: Log Spec.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Disabling the Analytics subsystem on Kubernetes

During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

## About this task

Disabling the Analytics subsystem stops the collection and storage of data as well as making the subsystem unavailable. When you disable Analytics, there will be no data in either the API Manager Dashboards or third-party offloads.

## Procedure

1. Back up your Analytics data as explained in [Backing up and restoring the analytics database](#).

2. Disable shard allocation for storage.

Disabling shard allocation is optional, but is recommended because it prevents new shards from being created for replication when a node is unavailable, and helps avoid corruption issues in full system restarts. Disable shard allocation for storage by running the following command:

```
kubectl -n namespace exec -it storage-master|shared_pod -- curl_es _cluster/settings -XPUT -d '{"persistent": {"cluster.routing.allocation.enable": "none"}}'
```

where:

- **namespace** is the namespace where the Analytics subsystem is installed.
- **storage-master|shared\_pod** is the name of any storage-master or storage-shared pod. You can get the name of your storage pods by running the following command and substituting in the namespace where Analytics is installed:



```
kubectl -n namespace get po
```

When you disable shard allocation, the response looks like the following example:

```
{"acknowledged": true, "persistent": {"cluster": {"routing": {"allocation": {"enable": "none"}}}}, "transient": {}}
```

3. Unassociate the Analytics service from all gateway services.




- a. In Cloud Manager, click  Topology.
- b. In the section for the Availability Zone that contains the Analytics service, locate the Gateway service that the Analytics service is associated with.
- c. Click  and select Unassociate analytics service.

All analytics collection will be disabled and there will be no data in either the API Manager Dashboards or third-party offloads. For more information, see [Associating an analytics service with a gateway service](#).

4. Unregister the Analytics service from Cloud Manager:



- a. In Cloud Manager, click  Topology.
- b. In the section for the Availability Zone that contains the Analytics service, locate the Analytics service and click Delete.

For more information, see [Registering an analytics service](#).

5. Perform a synced flush of storage.

Flushing the storage is optional, but is recommended because it helps to avoid losing the data that was not yet written to disk. This step flushes all current index operations synchronously and attempts to write everything that is in flux to disk. Perform a synced flush of storage by running the following command:

```
kubectl -n namespace exec -it storage-master|shared_pod -- curl_es _flush/synced -XPOST
```

When you flush storage, the response looks like the following example:

```
{"_shards": {"total": 33, "successful": 13, "failed": 0}, ".export-status": {"total": 1, "successful": 1, "failed": 0}, ".apic-config": {"total": 1, "successful": 1, "failed": 0}, ".kibana-6": {"total": 1, "successful": 1, "failed": 0}, "apic-api-2020.06.19-000002": {"total": 15, "successful": 5, "failed": 0}, "apic-api-2020.06.18-1": {"total": 15, "successful": 5, "failed": 0}}
```

The operation often fails, and it is safe to rerun it multiple times until it passes and there are all "failed" counts are zero. If it continues to fail after running multiple attempts over the span of a couple minutes, you can proceed to the next step.

6. Run the following two commands to configure analytics for ingestion-only:

```
$ apicup subsys set <analytics-subsystem-name> ingestion-only=true
```

```
$ apicup subsys install <analytics-subsystem-name>
```

7. Scale the analytics-ingestion deployment down to 0 replicas by completing the following steps.

- a. Get a list of all deployments so that you can find the name of the analytics-ingestion deployment:

```
$ kubectl get deployments -n <namespace>
```

- b. Scale the analytics-ingestion deployment down to 0 replicas:

```
$ kubectl scale deployment/<analytics-ingestion-deployment-name> --replicas=0 -n <namespace>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## VMware

Use the instructions in this section to install, upgrade, and maintain API Connect on VMware.

- [Installing and upgrading on VMware](#)  
Use these instructions to install or upgrade a deployment of API Connect on VMware.
- [Maintaining a VMware deployment](#)  
You can use utilities to complete maintenance tasks such as backup, restore, and certificate management in a VMware environment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Installing and upgrading on VMware

Use these instructions to install or upgrade a deployment of API Connect on VMware.

- [IBM API Connect Version 2018 software product compatibility requirements](#)  
Ensure that you install the minimum API Connect operating system requirements. Use the IBM® Software Product Compatibility Reports site to generate a requirements report appropriate for your API Connect version and environment.
- [Estimating internal storage space for Analytics](#)  
If you plan to store data locally in your IBM API Connect Analytics deployment, estimate disk space requirements.
- [Working with certificates](#)  
Use the `certs` command included in the APICUP installer to set and manage certificates for each subsystem. Default certificates are automatically applied, but defaults can be overridden by user-supplied custom certificates.
- [Tips and tricks for using APICUP](#)  
The APICUP installer in Install Assist contains built-in time saving functions.
- [Deploying to a VMware environment](#)  
The Install Assist tool provides a simplified installation for the OVA files into a VSphere environment.
- [Upgrading in a VMware environment](#)  
You can upgrade an existing deployment of API Connect on VMware to a newer version while retaining your data.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## IBM API Connect Version 2018 software product compatibility requirements

Ensure that you install the minimum API Connect operating system requirements. Use the IBM® Software Product Compatibility Reports site to generate a requirements report appropriate for your API Connect version and environment.

To generate an API Connect requirements report, complete the following steps:

1. Open the [Detailed system requirements for a specific product](#) page on the IBM Software Product Compatibility Reports site.
2. Search for the IBM API Connect product.
3. In the Search results list, select IBM API Connect.
4. From the Version list, select the required version.
5. Use the Filters to refine the contents of the requirements report.
6. Click Submit to generate your requirements report.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Estimating internal storage space for Analytics

If you plan to store data locally in your IBM® API Connect Analytics deployment, estimate disk space requirements.

This information applies only to Analytics deployments where the ingestion-only option is set to false (the default setting). The formula and guidelines are based on known information about how the Analytics service stores data, and cannot be directly applied to any other storage system.

Use the following guidelines to calculate a rough estimate of the amount disk space you need for storing stateful data in the API Connect Analytics microservices.

---

## Data storage: calculate how much data you want to store

---

This section applies if you have chosen a topology that uses the analytics-storage-data or analytics-storage-basic microservices.

The amount of disk space needed for storing analytics data is determined by the following factors:

Number of copies of the analytics data

Each copy of the analytics data must live on a separate node. The number of nodes in your deployment determines the number of copies of analytics data that is stored in your deployment. With the **Production** deployment profile, analytics data is replicated to three nodes (two replicas and one primary copy of the data) for storage. In the **Nonproduction** profile, you only need to store one copy of the data.

Number of copies of data:

```
Copies of data = [1 | 3]
```

Number of days that data is retained

By default, analytics data is retained for 90 days, but you can [modify the retention setting](#) as needed. Make sure you know how long you want to store the data before attempting to calculate required disk space.

Number of days that data is retained:

```
Data retention = [90 days | preferred length]
```

Amount of each type of data stored

The required storage space for each type of logging is highly dependent on your APIs and usage. For each API, you can [configure logging](#) for the API activity, header, and payload. You can also [customize the data](#) to add, redact, or remove fields, which also impacts the amount of data that you store.

To estimate storage needs, calculate the average size of each type of log. When calculating your estimates, remember that the header logging size is the sum of activity logging size and the average size of your headers. The payload logging size is the sum of the header logging size and the average size of your payloads. Typically the average size of an activity logged event is 600-1000 bytes depending on the uniqueness and complexity of your analytics data. This number is highly dependent on your APIs and your implementations. For a rough estimate, you can use an average of 800 bytes per activity logged event.

If you choose to add fields, calculate the average size of the new fields as well, and add that number to all types of log policies. If you choose to remove fields, you should not subtract this size from the log policies unless you are also removing headers and/or payloads.

Amount of each type of data that is stored:

```
Activity log bytes per call = [600-1000 bytes]
Header log bytes per call = Activity log bytes per call + Average size of headers
Payload log bytes per call = Header log bytes per call + Average size of payloads
```

Percentage of each type of data

Estimate the percentage of each type of log (activity, header, payload) for all API calls.

If you follow best practices of using only activity logging for production environments and using only payload logging for test environments, this number is easy to determine. If you use different log policies per API, and they depend on "success" or "error" factors, the percentage is more difficult to determine. Typically, if you do not use an all-or-nothing method for logging, error rates range from 3% to 25% with subsequent payload logging in test and production environments. However, this is entirely dependent on your use case and your APIs.

Percentage of each type of data that is stored:

```
% of Activity log = [0, 100 or other estimate]
% of Header log = [0, 100 or other estimate]
% of Payload log = [0, 100 or other estimate]
```

Estimated number of API calls per month

When planning your API Connect deployment, this number is helpful. When estimating analytics storage, this number is vital because it is directly correlated to how much storage you need for your deployment.

If you do not know the number of calls per month, but you do know the number of calls per second, use the following formula to convert it to calls per month:

```
Calls per month = Calls per second * 86400 seconds per day * 30 days per month
```

Number of API calls per month:

```
Calls per month = [any estimate]
```

## Calculating your disk space requirement

Estimate the disk space requirement for each storage node by completing following calculations.

Formula

```
Bytes per call = (% of Activity Log * Activity Log bytes per call) + (% of Header Log * Header Log bytes per call) + (% of
Payload Log * Payload Log bytes per call)
Calls per retention period = Calls per month * (Number of days retained / 30 days per month)
Storage for API calls = Calls per retention period * Bytes per call * Copies of data
Total storage = Storage for API calls + Overhead
Storage per node = Total storage / Nodes
```

Details

1. Bytes per call:

Estimate the number of bytes that are logged for a single copy of each API call. This can be calculated from the prerequisites of the percentage of each data type and the amount of each data type stored.

```
Bytes per call = (% of Activity Log * Activity Log bytes per call) + (% of Header Log * Header Log bytes per call) +
(% of Payload Log * Payload Log bytes per call)
```

2. Calls per retention period:

Calculate the anticipated number of API calls per retention period. This can be calculated from the prerequisites of the estimated calls per month and your desired data retention.

```
Calls per retention period = Calls per month * (Number of days retained / 30 days per month)
```



### 3. Storage for API Calls:

Calculate the storage needed for all copies of your api calls for your retention period. This can be calculated from steps 1 and 2, as well as the prerequisite of your total copies of your data.

**Storage for API calls = Calls per retention period \* Bytes per call \* Copies of data**

### 4. Total Storage:

Calculate the total storage you need by adding buffer space to the value from step 3. The Analytics service requires the some overhead space to complete its operations; for example, to contain system data and temporary debug header or payload logging. In addition, allowing extra space provides a buffer in case you underestimated the number of calls, or experience an unexpected increase.

There is no specific value for the buffer because it's based on your own situation. One approach is to use a value that brings the Storage for API calls result from step 3 up to the next round number. Make sure the rounding leaves you with a comfortable amount of additional space. For example, if the result from step 3 is 380 GB, then adding 20 GB to reach 400 GB is probably not sufficient and you should consider rounding to the a larger value such as 500 GB.

**Total storage = Storage for API calls + Buffer**

### 5. Storage per node:

Calculate the total storage amount required per storage node. The number of storage nodes is dependent on your deployment profile. For the **development** profile, use 1. For the **production** profile, it defaults to 3. If you manually scaled the analytics-storage-data or analytics-storage-basic microservices to be greater than 3, use your actual values.

**Storage per node = Total storage / Nodes**

Remember: This result is only an estimate. You should monitor the use of space over time and adjust storage as needed.

#### Example

Deployment information:

Copies of data = 3  
Data retention = 90 days  
Activity log bytes per call = 850 bytes  
Header log bytes per call = 15k bytes  
Payload log bytes per call = 30.5k bytes  
% of Activity log = 100%  
% of Header log = 0%  
% of Payload log = 0%  
Calls per month = 64 million

Formula:

Bytes per call = (% of Activity Log \* Activity Log bytes per call) + (% of Header Log \* Header Log bytes per call) + (% of Payload Log \* Payload Log bytes per call)  
Calls per retention period = Calls per month \* (Number of days retained / 30 days per month)  
Storage for API calls = Calls per retention period \* Bytes per call \* Copies of data  
Total storage = Storage for API calls + Overhead  
Storage per node = Total storage / Nodes

Details:

#### 1. Bytes per call = 850 bytes

**(100% of Activity Log \* 850 bytes) + (0% of Header Log \* 15k bytes) + (0% of Payload Log \* 30.5 bytes)**

#### 2. Calls per retention period = 192 million calls

**64 million calls per month \* (90 days retained / 30 days per month)**

#### 3. Storage for API calls = 489.6 GB

**192 million calls per period \* 850 bytes per call \* 3 copies of data**

#### 4. Total storage = 600 GB

**489.6GB storage + Buffer**

Since rounding to 500 GB only provides 10.4 GB of extra space, it's good practice to round to 600 GB instead.

#### 5. Storage per node = 200 GB

**600GB Total storage / 3 nodes**

In this example, the estimated disk needed on each node for analytics-storage-data and analytics-storage-basic microservices is 200GB.

## Master storage: estimate disk space needs

This section applies if you are planning a topology with the analytics-storage-master microservice enabled.

For the analytics-storage-master microservice, estimating the amount of disk space you need is much easier. For a production environment, set this value to 10GB. For a development environment, you can optionally reduce the value to 5GB.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with certificates

Use the `certs` command included in the APICUP installer to set and manage certificates for each subsystem. Default certificates are automatically applied, but defaults can be overridden by user-supplied custom certificates.

## About this task

Note: Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

- [Certificate management: Read This First](#)  
Requirements and best practices for managing API Connect certificates.
- [Setting and managing certificates](#)  
Default certificates are automatically applied, but defaults can be overridden by custom certificates.
- [Reference for certificates, commands, and validations](#)  
This section contains the reference information for certificates, commands for working with certificates, and validations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Certificate management: Read This First

Requirements and best practices for managing API Connect certificates.

Important: Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged. For management purposes, API Connect certificates can be grouped by type (default, custom, and common) and by usage (public, public and user-facing, and internal). Understand each type and usage before you change any certificates.

- A default certificate is a private certificate that is uniquely generated by the installer for the current project directory, and will pass validation. Default certificates are automatically generated for each subsystem by the `apicup subsys install` command, unless the certificates were explicitly set by using the `apicup certs set` command. Note that the default certificates are self-signed, so they might not provide a level of trust suitable for external communication.
- For optimal trust levels, we recommend that you explicitly set all public and user-facing certificates by creating custom certificates.
- Some certificates are common across subsystems. Subsystems require the common certificates to allow them to register with the management subsystem. When installing any one subsystem, the common certificates are set for that subsystem and for all the other subsystems. If you use custom certificates for the common certificates, the custom certificates must be set prior to setting any other custom certificates.
- We do not recommend the explicit setting of internal certificates. The changing of internal certificates creates a risk of incompatibility with other internal certificates.

To review the usage (public, public and user-facing, or internal) of each certificate, see the following [Certificate management best practice](#) table.

Table 1. Certificate management best practice

Certificate usage	Certificate name	Best practice management
Public	<ul style="list-style-type: none"> <li>• apic-gw-service-ingress</li> </ul>	<p>For optimal trust levels, set these explicitly.</p> <p>If certificates are not explicitly set by using the <code>apicup certs set</code> command, then default self-signed certificates are automatically generated by <code>apicup subsys install</code>. Typically you usually want to customize them, to ensure a level of trust suitable for external communication.</p>
Public and User-facing	<ul style="list-style-type: none"> <li>• platform-api, api-manager-ui, cloud-admin-ui</li> <li>• consumer-api</li> <li>• portal-www-ingress</li> </ul>	<p>For optimal trust levels, set these explicitly.</p> <p>Recommended to be explicitly set as custom certificates because they are presented to an end user through a browser or Command Line Interface (CLI).</p>
Internal	<ul style="list-style-type: none"> <li>• root-ca, ingress-ca</li> <li>• mgmt-db-ca, mgmt-ca, service-server, service-client, db-server, db-client, service-plugin</li> <li>• portal-admin-ingress, portal-client, portal-ca, portal-db-ca, service-server, service-client, apim-client</li> <li>• analytics-client-ingress, analytics-ingestion-ingress, analytics-ingestion-client, analytics-client-client</li> <li>• analytics-ca, service-server, service-client</li> <li>• gw-ca, gateway-peering</li> </ul> <p>For VMware appliance deployments only: k8s-ca, appliance-client</p>	<p>Do not change. Accept the default certificates. It is possible to change these certificates (except ingress-ca) but is strongly discouraged because you risk creating incompatibilities that can block internal communications.</p> <p>Each intermediate cert (mgmt-db-ca, mgmt-ca, portal-ca, portal-db-ca, analytics-ca, gw-ca), is used to generate other internal certs. If, for example, you change an intermediate cert, the certs generated from it might not work with internal certs generated from other intermediate certs.</p> <p>Note that ingress-ca is auto-generated and cannot be set using the <code>apicup certs set</code> command.</p>

See also:

- [Setting and managing certificates](#)
- [Reference for certificates, commands, and validations](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting and managing certificates

Default certificates are automatically applied, but defaults can be overridden by custom certificates.

- [Setting default certificates](#)  
Default certificates are automatically generated by APICUP when the subsystem is installed.
- [Setting custom certificates](#)  
Use the APICUP installer `certs` commands to set custom certificates.
- [Replacing custom certificates](#)  
Use the APICUP installer `certs` commands to replace existing certificates.
- [Setting common certificates](#)  
Common certificates are set for one subsystem, but are applied to all subsystems. Use the APICUP installer `certs` commands to set the common certificates.
- [Setting the encryption-secret for the management database](#)  
Use the APICUP installer `certs` commands to set the encryption-secret for the management database.
- [Clearing certificates](#)  
Certificates can be cleared in order to set new certificates.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Setting default certificates

Default certificates are automatically generated by APICUP when the subsystem is installed.

---

### About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

Default certificates are generated for each subsystem by the `apicup subsys install` command. If certificates are not explicitly set by using the `apicup certs set` command, then default certificates are automatically generated by APICUP. The default certificates are self-signed, so they might not provide a level of trust suitable for external communication.

---

## Procedure

1. Enter the settings for the subsystem by using `apicup subsys set <SUBSYS>` and validate the subsystem settings by using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Install the subsystem by using the `apicup subsys install` command.
3. The default certificates are created for the subsystem. A default certificate is a private certificate that is uniquely generated by the installer for this project directory, they are self-signed and always pass validation.
4. List all certificates that are set for a subsystem by using the `apicup certs list` command.

```
apicup certs list -help

List all configured certificates

Usage:
  apicup certs list SUBSYS [flags]

Flags:
  -h, --help  help for list

Global Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging
```

Following is example output from the `apicup certs list` command:

```
Common certificates
=====
Name                Summary                Validation errors
-----
analytics-client-client
CN: analytics-client-client
SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2
```

```

analytics-ingestion-client AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
                           CN: analytics-ingestion-client
                           SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B
ingress-ca AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
           CN: ingress-ca
           SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
mgmt-db-ca AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7
           CN: mgmt-db-ca
           SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66
portal-client AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7
             CN: portal-client
             SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D
root-ca AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
        CN: root-ca
        SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7
        AuthorityKeyId:

```

#### Subsystem mgmt\_subsys certificates

Name	Summary	Validation errors
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
consumer-api	CN: consumer-api SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
db-client	CN: db-client SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88 AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
db-server	CN: db-server SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
encryption-secret mgmt-ca	A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30 CN: mgmt-ca SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
migration-client	CN: migration-client SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
platform-api	CN: platform-api SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:95:7D:3E:C5:43:F9:48 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
service-client	CN: service-client SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-plugin	CN: service-plugin SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-server	CN: service-server SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting custom certificates

Use the APICUP installer `certs` commands to set custom certificates.

### About this task

#### Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

The APICUP installer can be used to set certificates for each subsystem during installation. If certificates are not explicitly set by using the `apicup certs set` command, then default certificates are generated by APICUP. The default certificates are self-signed, so they might not provide a level of trust suitable for external communication.

#### Requirements for custom certificates:

- Extended Key Usage (EKU), either `serverAuth` or `clientAuth` depending upon the type of certificate. Certificates of type `Server` must have an Extended Key Usage with `serverAuth` purpose. Certificates of type `Client` must have an Extended Key Usage with `clientAuth` purpose.
- Subject Alternative Name (SAN) for the required hosts
- Any custom common certificates that are being used must be set prior to setting any custom certificates for a subsystem.

See [Certificate reference](#) to view the list of common certificates and to determine whether an EKU is needed for a certificate and which type of EKU (serverAuth or clientAuth).

Precedence order for TLS certificates for Management endpoints:

The Management subsystem has four public endpoints: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, and *consumer-api*. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, *consumer-api*.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the *api-manager-ui* TLS certificate will be effective for all endpoints, as it is first in the precedence order.
- If the *api-manager-ui*, *cloud-admin-ui*, and *platform-api* endpoints are the same, and *consumer-api* is a different endpoint, then the *api-manager-ui* TLS certificate will be effective for the *api-manager-ui/cloud-admin-ui/platform-api* endpoints, while the *consumer-api* TLS certificate will be effective for the *consumer-api* endpoint

Note: Once API Connect has been installed (meaning that the `apicup subsys install SUBSYS` command has been executed) with a given set of certificates, only the certificates for the public ingress endpoints (*portal-www*, *api-manager-ui*, *cloud-admin-ui*, *platform-api*, *consumer-api*) can be modified. The TLS certificates involved in mutual authentication (*portal-admin-ingress*, *portal-client*, *analytics-ingestion-ingress*, *analytics-ingestion-client*, *analytics-client-ingress*, and *analytics-client-client*) cannot be modified after the `install` command has been executed.

## Procedure

1. Set up and validate the subsystem. Enter the settings for the subsystem using `apicup subsys set <SUBSYS>` and validate the subsystem settings using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Generate the custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (*platform-api-example*) with an EKU *serverAuth* and SAN using `openssl`:

```
openssl x509 -req -days 360 -in platform-api-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:fqdn.myserver.com\nextendedKeyUsage=serverAuth")) -extensions SAN
```

where

- `DNS:fqdn.myserver.com` is the fully qualified domain name of the endpoint the certificate applies to. This matches the endpoints entered in the APICUP installer. See [Deploying the Management subsystem in a VMware environment](#)
- `platform-api-example.csr` is the file name for the certificate signing request

Following is an example for how to generate a certificate (*portal-client*) with an EKU *clientAuth* and SAN using `openssl`:

```
openssl x509 -req -days 360 -in portal-client-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical,CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

3. Once the certificate has been created, set the certificate by entering the following command:  
`apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE]`

If the certificate is signed by an intermediate CA, the `CA_File` argument must point to a file that concatenates the intermediate CA, followed by the root CA, in that order.

You can find definitions for commands at the following location: [Command reference](#)

If the certificate was generated with an EKU *serverAuth*, it must be assigned to a server certificate. If the certificate was generated with an EKU *clientAuth*, it must be assigned to a client certificate.

4. Repeat for additional custom certificates.
5. After setting the custom certificates, you can optionally generate the remaining default certificates prior to installation by entering the `apicup certs generate` command. The `generate` command generates any certificates that have not been set, so it will create default certificates for all remaining certificates. It will not overwrite any custom certificates you have set. You can review the certificates prior to installation.
6. List all certificates with `apicup certs list SUBSYS`. The results will include the generated default certificates and the custom certificates that you set.

Following is example output from the `apicup certs list` command:

```
list command:

Common certificates
=====
```

Name	Summary	Validation errors
----	-----	-----
analytics-client-client	CN: analytics-client-client SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
analytics-ingestion-client	CN: analytics-ingestion-client SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
ingress-ca	CN: ingress-ca SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
mgmt-db-ca	CN: mgmt-db-ca SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
portal-client	CN: portal-client SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
root-ca	CN: root-ca SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	

**AuthorityKeyId:**

**Subsystem mgmt\_subsys certificates**

Name	Summary	Validation errors
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
consumer-api	CN: consumer-api SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
db-client	CN: db-client SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88 AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
db-server	CN: db-server SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
encryption-secret	A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30	
mgmt-ca	CN: mgmt-ca SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
migration-client	CN: migration-client SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
platform-api	CN: platform-api SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:7D:3E:C5:43:F9:48 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
service-client	CN: service-client SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-plugin	CN: service-plugin SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	
service-server	CN: service-server SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7 AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75	

7. Install the subsystem with the certificates using `apicup subsys install SUBSYS`. Any missing certificates will be generated. The installation will not proceed if there are any validation issues with the certificates. See [Validation reference](#).
8. Repeat for other subsystems.
9. If necessary, you can replace custom certificates after installation is complete. See [Replacing custom certificates](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Replacing custom certificates

Use the APICUP installer `certs` commands to replace existing certificates.

### About this task

**Important:**

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

The APICUP installer can be used to update certificates for each subsystem after installation.

Requirements for custom certificates:

- Extended Key Usage (EKU), either `serverAuth` or `clientAuth` depending upon the type of certificate. Certificates of type *Server* must have an Extended Key Usage with `serverAuth` purpose. Certificates of type *Client* must have an Extended Key Usage with `clientAuth` purpose.
- Subject Alternative Name (SAN) for the required hosts
- Any custom common certificates that are being used must be set prior to setting any custom certificates for a subsystem.

See [Certificate Reference](#) to view the list of common certificates and to determine whether an EKU is needed for a certificate and which type of EKU (`serverAuth` or `clientAuth`).

Precedence order for TLS certificates for Management endpoints:

The Management subsystem has four public endpoints: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, and `consumer-api`. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: `api-manager-ui`, `cloud-admin-ui`, `platform-api`, `consumer-api`.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the api-manager-ui TLS certificate will be effective for all endpoints, as it is first in the precedence order.
- If the api-manager-ui, cloud-admin-ui, and platform-api endpoints are the same, and consumer-api is a different endpoint, then the api-manager-ui TLS certificate will be effective for the api-manager-ui/cloud-admin-ui/platform-api endpoints, while the consumer-api TLS certificate will be effective for the consumer-api endpoint

## Procedure

1. Generate the custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (platform-api-example) with an EKU serverAuth and SAN using openssl:

```
openssl x509 -req -days 360 -in platform-api-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:fqdn.myserver.com\nextendedKeyUsage=serverAuth")) -extensions SAN
```

where

- `DNS:fqdn.myserver.com` is the fully qualified domain name of the endpoint the certificate applies to. This matches the endpoints entered in the APICUP installer. See [Deploying the Management subsystem in a VMware environment](#).
- `platform-api-example.csr` is the file name for the certificate signing request

Following is an example for how to generate a certificate (portal-client) with an EKU clientAuth and SAN using openssl:

```
openssl x509 -req -days 360 -in portal-client-example.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical,CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

2. Once the certificate has been created, set the certificate by entering the following command:

```
apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE]
```

You can find definitions for commands at the following location: [Command reference](#)

If the certificate was generated with an EKU serverAuth, it must be assigned to a server certificate. If the certificate was generated with an EKU clientAuth, it must be assigned to a client certificate.

3. Install the subsystem with the new certificate using `apicup subsys install SUBSYS`. Any missing certificates will be generated. The installation will not proceed if there are any validation issues with the certificates. See [Validation reference](#).
4. Repeat for other subsystems requiring new certificates.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting common certificates

Common certificates are set for one subsystem, but are applied to all subsystems. Use the APICUP installer `certs` commands to set the common certificates.

## About this task

Important:

- Customization of public certificates and public user-facing certificates is recommended. Customization of internal certificates is strongly discouraged.
- To view a list of public, public user-facing, and internal certificates, see [Certificate management: Read This First](#). For details on each certificate, see [Certificate reference](#).
- Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

The common certificates are identical across subsystems. Subsystems require the common certificates to allow them to register with the management subsystem. When installing any one subsystem, the common certificates will be set for that subsystem and for all the other subsystems. If you are using custom certificates for the common certificates, they must be set prior to setting any custom certificates. See [Certificate reference](#) for a description of the common certificates.

Common certificates cannot be changed between subsystem installs. For example, you cannot set a common certificate for the management subsystem, install the management subsystem, then change a common certificate for the analytics subsystem, then install the analytics subsystem. This scenario will result in a failed installation because the common certificates are not identical.

## Procedure

Follow these steps to set custom common certificates. If using default certificates, the common certificates will be set for you. See [Setting default certificates](#).

1. Set up and validate all subsystems. Enter the settings for the subsystem using `apicup subsys set <SUBSYS>` and validate the subsystem settings using `apicup subsys get <SUBSYS> --validate`. The subsystem must pass validation before setting the certificates.
2. Generate a custom certificate with the appropriate EKU and SAN. You will need to obtain the private key, public certificate, and CA certificates in non-password-protected PEM format for the custom certificate. Following is an example for how to generate a certificate (platform-api) with an EKU serverAuth and SAN using openssl:



```
openssl x509 -req -days 360 -in platform-api.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out platform-api-sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nsubjectAltName=DNS:ac-msntest.myserver.com,DNS:ac2-msntest.myserver.com\nextendedKeyUsage=serverAuth")) -extensions SAN
```

Following is an example for how to generate a certificate (portal-client) with an EKU clientAuth and SAN using openssl:

```
openssl x509 -req -days 360 -in portal-client.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -out portal-client-cert -sha256 -extfile <(cat /etc/ssl/openssl.cnf <(printf "\n[SAN]\nkeyUsage=critical,digitalSignature,keyEncipherment\nextendedKeyUsage = clientAuth\nbasicConstraints=critical,CA:FALSE\nsubjectKeyIdentifier=hash\n")) -extensions SAN
```

3. Once the certificate has been created and you have a `.pem` file, set the custom common certificates in one of your subsystems. After setting the custom certificates for one subsystem, they will take effect for all subsystems. The command is targeted at a specific subsystem, but the common certificates are copied to all subsystems regardless of which subsystem they are originally set in. Following is an example for setting the `portal-client` certificate:  
`apicup certs set mgmt portal-client myCertFile.pem myKeyFile.key myCAFile.crt`
4. Set the remaining certificates for each subsystem, default or custom. See [Setting default certificates](#) and [Setting common certificates](#).
5. List and validate the certificates for each subsystem. See [Validation reference](#).
6. Install each subsystem using `apicup subsys install SUBSYS`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting the encryption-secret for the management database

Use the APICUP installer `certs` commands to set the encryption-secret for the management database.

### About this task

Note: Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

The encryption-secret is a secure random bytes password used for field level encryption in the management database. You can generate 128 random bytes using the following command in openssl:

```
openssl rand -out /path/to/secret/encryption-secret.bin 128
```

### Procedure

1. Enter the `apicup certs set SUBSYS CERT_NAME [KEY_FILE]` command and complete the following values:
  - SUBSYS - The subsystem for the `encryption-secret` is the name of your management subsystem, because it is used for field-level encryption for the management database.
  - CERT\_NAME - The certificate name is `encryption-secret`.
  - KEY\_FILE - Enter the file name for a secure random bytes string that is 128 bytes in length, for example `encryption-secret.bin`.
2. Set the remaining certificates if using custom certificates and install the management subsystem.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Clearing certificates

Certificates can be cleared in order to set new certificates.

### About this task

Note: Certificates are automatically copied during an upgrade (if the upgrade is initiated from the original project directory). For more information, see [Requirements for upgrading on VMware](#).

Existing certificates must be cleared in order to set new certificates. When using default certificates, new certificates are created only for those certificates that are not set. Some of the use cases for clearing certificates are: expired certificates and configuration changes. For example, if endpoints are changed, the existing certificates must be cleared and new certificates created.

### Procedure

1. Enter the `apicup certs set SUBSYS CERT_NAME --clear` command and complete the following values:
  - SUBSYS - The name of the subsystem
  - CERT\_NAME - The name of the certificate that you want to clear.



2. The certificate will be cleared and can be reset, either as a default or custom certificate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Reference for certificates, commands, and validations

This section contains the reference information for certificates, commands for working with certificates, and validations.

- [Certificate reference](#)  
The Certificate reference provides a description of all the certificates required in API Connect.
- [Command reference](#)  
The APICUP installer includes the `certs` commands to set and manage certificates.
- [Validation reference](#)  
Certificates are validated using several parameters.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Certificate reference

The Certificate reference provides a description of all the certificates required in API Connect.

---

### Before you begin

APICUP sets default certificates for each subsystem during installation. The default certificates are self-signed so may not provide a level of trust suitable for external communication. Custom certificates can be set and managed following the steps described in [Setting custom certificates](#).

The Certificates reference topic lists all certificates that are set by the `apicup certs` commands. The default certificates can be used for the majority of these certificates. The certificates that are marked as *public and user-facing* are recommended to be explicitly set as custom certificates because they are presented to an end user through a browser or Command Line Interface (CLI).

The certificates that are described as *TLS certificate used by ingress* are also considered *public* in the sense that they interact with a client that sits outside of an API Connect cluster.

The remaining certificates are considered *internal* because they interact with internal components.

Certificates that are listed as *auto-generated* cannot be set using the `apicup certs set` command. For example, the common certificate `ingress-ca` is auto-generated and used as an intermediate CA for all default ingress certificates. If you set an ingress certificate as a custom certificate, you will need to configure an intermediate CA if desired.

Important:

When setting custom certificates, additional steps must be taken to provide an Extended Key Usage (EKU) serverAuth and ClientAuth and a Subject Alternative Name (SAN) for the required hosts. These certificates are generated automatically by the `apicup certs` command when using the default certificates.

- For custom certificates of type *server*, an additional extended key usage client authentication (EKU serverAuth) certificate is required.
- For custom certificates of type *client*, an additional extended key usage server authentication (EKU clientAuth) certificate is required.

---

## Procedure

1. **Common certificates** - The following certificates are common to all subsystems in a deployment. Subsystems require the common certificates to allow them to register with the management subsystem. The common certificates are identical across subsystems. When installing any one subsystem, the common certificates will be set for that subsystem and for all the other subsystems. Custom common certificates must be set prior to setting any custom subsystem certificates.

Common certificates cannot be changed between subsystem installs. For example, you cannot set a common certificate for the management subsystem, install the management subsystem, then change a common certificate for the analytics subsystem, then install the analytics subsystem. This scenario will result in a failed installation because the common certificates are not identical.

Table 1. Common certificates

Certificate name (value used in <code>apicup certs</code> )	Type	Usage	Requirements	Description
root-ca	CA	internal		CA certificate which forms the root of the certificate chain

Certificate name (value used in apicup certs)	Type	Usage	Requirements	Description
ingress-ca	CA	internal	signed by: root-ca	Auto-generated intermediate certificate used to generate certificates for subsystem ingress endpoints if not provided by user. The ingress-ca intermediate certificate cannot be set explicitly, it is always only generated. TLS certificates for the ingresses are not required to use ingress-ca as an intermediate certificate, but if a given ingress TLS certificate is left to be auto-generated then it will be signed by this ingress-ca.
mgmt-db-ca	CA	internal	signed by: root-ca	Intermediate CA certificate used to sign certificates used by Cassandra.
portal-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <b>portal-admin-ingress</b> .	Client certificate used by management subsystem to authenticate with the Portal admin endpoint. Requires EKU clientAuth.
analytics-client-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <b>analytics-client-ingress</b> .	Client certificate used by management subsystem to authenticate with analytics client endpoint. Requires EKU clientAuth.
analytics-ingestion-client	Client	internal	Must be signed by the same authority as the one used for the matching ingress TLS certificate <b>analytics-ingestion-ingress</b> .	Client certificate used by gateway subsystem to authenticate with analytics ingestion endpoint. Requires EKU clientAuth.

## 2. Management certificates

These certificates apply to a single Management subsystem.

The Management subsystem has four endpoints: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, and *consumer-api*. Distinct TLS certificates can be set for each endpoint. However, if any two endpoints identical, only one TLS certificate will be effective. The order of precedence is: *api-manager-ui*, *cloud-admin-ui*, *platform-api*, *consumer-api*.

Following are examples for how precedence is determined for TLS certificates for endpoints:

- If all four endpoints are distinct, then all four TLS certificates will be effective for their respective endpoints.
- If all four endpoints are identical, then only the *api-manager-ui* TLS certificate will be effective for all endpoints, as it is first in the precedence order.
- If the *api-manager-ui*, *cloud-admin-ui*, and *platform-api* endpoints are the same, and *consumer-api* is a different endpoint, then the *api-manager-ui* TLS certificate will be effective for the *api-manager-ui/cloud-admin-ui/platform-api* endpoints, while the *consumer-api* TLS certificate will be effective for the *consumer-api* endpoint

The encryption-secret certificate is unique in that it is a secure random number. It is used to provide field level encryption in management (Cassandra) database. To set the encryption secret for the management database, use the following command: **apicup certs set SUBSYS CERT\_NAME**

[KEY\_FILE] For example: **apicup certs set mgmt1 encryption-secret /path/to/keyfile**. See [Setting the encryption-secret for the management database](#)

Table 2. Management certificates

Certificate name	Type	Usage	Requirements	Description
encryption-secret	secure random bytes		length: 128 bytes	Encryption secret used to do field level encryption in management (Cassandra) database
platform-api	Server	public and user-facing	The host names for which the certificate is valid must include the platform-api endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <i>store.acme.com</i> , the certificate can be one that is valid for host names matching <i>*.acme.com</i> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
consumer-api	Server	public and user-facing	The host names for which the certificate is valid must include the consumer-api endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <i>store.acme.com</i> , the certificate can be one that is valid for host names matching <i>*.acme.com</i> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
api-manager-ui	Server	public and user-facing	The host names for which the certificate is valid must include the api-manager-ui endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <i>store.acme.com</i> , the certificate can be one that is valid for host names matching <i>*.acme.com</i> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
cloud-admin-ui	Server	public and user-facing	The host names for which the certificate is valid must include the cloud-admin-ui endpoint. A wildcard certificate can be used as the first element in a host name, for example, if the endpoint is: <i>store.acme.com</i> , the certificate can be one that is valid for host names matching <i>*.acme.com</i> .	TLS certificate used by ingress. Requires EKU serverAuth. Public and user-facing.
mgmt-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign management subsystem certificates

Certificate name	Type	Usage	Requirements	Description
service-server	Server	internal	signed by: mgmt-ca Required hosts: <ul style="list-style-type: none"> <li>*.&lt;namespace&gt;</li> <li>*.&lt;namespace&gt;.svc</li> <li>*.&lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificates used by management services. Requires EKU serverAuth.
service-client	Client	internal	signed by: mgmt-ca	Client certificate used by management services. Requires EKU clientAuth.
db-server	Server	internal	signed by: mgmt-db-ca Required hosts: <ul style="list-style-type: none"> <li>*.&lt;namespace&gt;</li> <li>*.&lt;namespace&gt;.svc</li> <li>*.&lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificate used by management (Cassandra) database to communicate with other nodes. Requires EKU serverAuth.
db-client	Client	internal	signed by: mgmt-db-ca	Client certificate used by management services. Requires EKU clientAuth.
service-plugin	Client	internal	signed by: mgmt-ca	Client certificate used by plugin services to talk to management subsystem. Requires EKU clientAuth.
migration-client	Client		signed by: mgmt-ca	This certificate is deprecated and no longer used, and can be ignored.

### 3. Portal certificates

These certificates apply to a single portal subsystem.

Table 3. Portal certificates

Certificate name	Type	Usage	Requirements	Description
portal-admin-ingress	Server	internal	host must match admin endpoint	TLS certificate used by ingress. The <b>portal-client</b> common certificate must be set prior to setting the <b>portal-admin-ingress</b> certificate. Requires EKU serverAuth. The <b>portal-admin-ingress</b> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <b>ingress-ca</b> .
portal-www-ingress	Server	public and user-facing	host must match www endpoint	TLS certificate used by ingress. Requires EKU serverAuth.
portal-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign portal subsystem certificates
portal-db-ca	CA	internal	signed by: portal-ca	Intermediate CA certificate used to sign certificates used by portal DB
service-server	Server	internal	signed by: portal-ca Required hosts: <ul style="list-style-type: none"> <li>*.&lt;namespace&gt;</li> <li>*.&lt;namespace&gt;.svc</li> <li>*.&lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificates used by portal services. Requires EKU serverAuth.
service-client	Client	internal	signed by: portal-ca	Client certificate used by portal services. Requires EKU clientAuth.
apim-client	Client	internal	signed by: portal-ca	Client certificate used by portal services to talk to management subsystem. Requires EKU clientAuth.

### 4. Analytics certificates

These certificates apply to a single analytics subsystem.

Table 4. Analytics certificates

Certificate name	Type	Usage	Requirements	Description
------------------	------	-------	--------------	-------------

Certificate name	Type	Usage	Requirements	Description
analytics-client-ingress	Server	internal	Required hosts: <ul style="list-style-type: none"> <li>client ingress endpoint</li> <li>*, &lt;namespace&gt;</li> <li>*, &lt;namespace&gt;.svc</li> <li>*, &lt;namespace&gt;.svc.cluster.local</li> </ul>	TLS certificate used by ingress. The <b>analytics-client-client</b> common certificate must be set prior to setting the <b>analytics-client-ingress</b> certificate. Requires EKU serverAuth. The <b>analytics-client-ingress</b> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <b>ingress-ca</b> .
analytics-ingestion-ingress	Server	internal	Required hosts: <ul style="list-style-type: none"> <li>ingestion ingress endpoint</li> <li>*, &lt;namespace&gt;</li> <li>*, &lt;namespace&gt;.svc</li> <li>*, &lt;namespace&gt;.svc.cluster.local</li> </ul>	TLS certificate used by ingress. The <b>analytics-ingestion-client</b> common certificate must be set prior to setting the <b>analytics-ingestion-ingress</b> certificate. Requires EKU serverAuth. The <b>analytics-ingestion-ingress</b> TLS certificate does not have any specific requirement on the authority signing it. If the certificate is auto-generated, it is signed by the <b>ingress-ca</b> .
analytics-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign analytics subsystem certificates
service-server	Server	internal	signed by: analytics-ca Required hosts: <ul style="list-style-type: none"> <li>*, &lt;namespace&gt;</li> <li>*, &lt;namespace&gt;.svc</li> <li>*, &lt;namespace&gt;.svc.cluster.local</li> </ul>	Server certificates used by analytics services. Requires EKU serverAuth.
service-client	Client	internal	signed by: analytics-ca	Client certificate used by analytics services; requires EKU clientAuth.

#### 5. Gateway certificates

These certificates apply to a single gateway subsystem.

Table 5. Gateway certificates

Certificate name	Type	Usage	Requirements	Description
api-gateway-ingress	Server	public and user-facing	host must match api-gateway endpoint	TLS certificate used by ingress. Requires EKU serverAuth. (deprecated, see <b>Note</b> .)
apic-gw-service-ingress	Server	public	host must match apic-gw-service endpoint	TLS certificate used by ingress. Requires EKU serverAuth.
gw-ca	CA	internal	signed by: root-ca	Intermediate CA used to sign gateway subsystem certificates
gateway-peering	Server	internal	signed by: gw-ca	Server certificates used by gateway services. Requires EKU serverAuth.

Note: The **api-gateway-ingress** certificate is a legacy certificate which has been deprecated. It is no longer used to terminate TLS at the api-gateway endpoint. You configure a TLS profile for the termination of the api-gateway endpoint using the Cloud Manager API Invocation Endpoint and SNI settings when registering the gateway. The profiles used during configuration can be updated as needed. See [Registering a gateway service](#)

#### 6. OVA/Appliance certificates

These certificates apply only to an appliance in a VMware environment. The **k8s-ca** and **appliance-client** certificates are auto-generated and cannot be set as custom certificates. These are internal certificates used to set up the Kubernetes cluster in an OVA/Appliance.

Table 6. OVA/Appliance certificates

Certificate name	Type	Requirements	Description
k8s-ca	CA		Auto-generated CA certificate that forms the root of the certificate chain for the Appliance/Kubernetes cluster components
appliance-client	Client	signed by: k8s-ca	Auto-generated client certificate used to communicate with the Appliance API server. Requires EKU clientAuth.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Command reference

The APICUP installer includes the `certs` commands to set and manage certificates.

### About this task

The APICUP installer can be used to set certificates for each subsystem during installation. If certificates are not explicitly set using the `apicup certs set` command, then default certificates are generated by APICUP. We recommend that certificates be set at installation time only (or carried over from an upgrade). The default certificates are self-signed, so they may not be optimal for external communication.

For a description of the certificates that can be set, see [Certificate reference](#). We recommend that all public and user-facing certificates be explicitly set, including `portal-www-ingress` and `api-gateway-ingress`, and the four management endpoints (`platform-api`, `consumer-api`, `api-manager-ui`, and `cloud-admin-ui`). Following is the help reference for the `apicup certs set` command:

```
apicup certs set --help
Set or clear certificates and keys

Usage:
apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE] [KEY_FILE] [flags]

Flags:
--clear    Clear out a certificate or key entry
-h, --help help for set

Global Flags:
--accept-license Accept the license for API Connect
--debug         Enable debug logging
```

### Procedure

To set and clear certificates, complete the following steps:

1. Enter the `apicup certs set` command and complete the following values:

Table 1. `apicup certs set`

Command	Values	Result
<code>apicup certs set SUBSYS CERT_NAME [CERT_FILE KEY_FILE CA_FILE] [KEY_FILE] [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>• SUBSYS - name of the subsystem to which the certificate applies</li> <li>• CERT_NAME - name of the certificate; see <a href="#">Certificate reference</a> for a list of certificates that can be set for each subsystem.</li> <li>• CERT_FILE - Path to the certificate file in PEM format.</li> <li>• KEY_FILE - Path to the private key file in PEM format.</li> <li>• CA_FILE - Path to the Certificate Authority (CA) file. The contents of the file may be the concatenation of an intermediate CA and the root CA (in that order). <b>Note:</b> When setting the <code>root-ca</code> certificate, omit the <code>CA_FILE</code> parameter.</li> </ul>	Applies the certificate when the subsystem is installed.
<code>apicup certs set SUBSYS CERT_NAME [KEY_FILE] [flags]</code>	<code>KEY_FILE</code> - The file containing the encryption-secret for field level encryption in the management database. Applies only to the management subsystem. The certificate name is <code>encryption-secret</code> . The type is secure random bytes with a length of 128 bytes. For example, <code>apicup certs set mgmt1 encryption-secret /path/to/encryption-secret.bin</code> . <b>Note:</b> Do not specify any of the <code>[CERT_FILE KEY_FILE CA_FILE]</code> parameters when setting the encryption-secret.	Applies the <code>encryption-secret</code> when the management subsystem is installed.
<code>flags</code> <ul style="list-style-type: none"> <li>• <code>--clear</code></li> <li>• <code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li>• <code>--clear</code> - Clears the specified certificate. For example, <code>apicup certs set mgmt1 encryption-secret --clear</code></li> <li>• <code>--help</code> - Displays help for the command.</li> </ul>	The specified certificate will be cleared. When making configuration changes such as changing endpoints, the corresponding certificate must be cleared so that a new certificate can be set.

2. The `apicup certs get` command retrieves a specific certificate for the specified subsystem.

```
apicup certs get --help
Get a certificate

Usage:
apicup certs get SUBSYS CERT_NAME [flags]

Flags:
-h, --help help for get
-o, --output string output to file or - (stdout) (default "-")
-t, --type string type of object to return: cert, key, ca (default "cert")

Global Flags:
--accept-license Accept the license for API Connect
--debug         Enable debug logging
```

Table 2. `apicup certs get`

Command	Values	Result
---------	--------	--------

Command	Values	Result
<code>apicup certs get SUBSYS CERT_NAME [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem to which the certificate applies</li> <li>CERT_NAME - name of the certificate to retrieve; see <a href="#">Certificate reference</a> for a list of certificates</li> </ul>	Returns the specified certificate for the specified subsystem.
<code>flags</code> <ul style="list-style-type: none"> <li><code>--output string</code></li> <li><code>--type string</code></li> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--output string</code> - Specify a file for the retrieved values, or specify "-" to send to stdout. Default is "-" to send to stdout. For example, <code>apicup certs get mgmt1 --output myCertsFile</code></li> <li><code>--type string</code> - Returns only the specified type. If not specified, the type is cert. For example, <code>apicup certs get mgmt1 --type ca</code></li> <li><code>--help</code> - Displays help for the command.</li> </ul>	<ul style="list-style-type: none"> <li>For <code>--output</code>: The specified certificate will be retrieved and sent to stdout or saved to the specified file</li> <li>For <code>--type</code>: Certificates will be retrieved that match the type specified.</li> </ul>

3. List all certificates that have been set for a subsystem using the `apicup certs list` command. You can list the certificates at any time to summarize the certificates that have been set.

```
apicup certs list -help
```

```
List all configured certificates
```

```
Usage:
```

```
apicup certs list SUBSYS [flags]
```

```
Flags:
```

```
-h, --help help for list
```

```
Global Flags:
```

```
--accept-license Accept the license for API Connect
--debug Enable debug logging
```

Table 3. apicup certs list

Command	Values	Result
<code>apicup certs list SUBSYS [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem for which you want to list certificates</li> </ul>	Returns a list of certificates that are configured for the subsystem.
<code>flags</code> <ul style="list-style-type: none"> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--help</code> - Displays help for the command.</li> </ul>	Help text is displayed.

Following is example output from the `apicup certs list` command:

```
Common certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
analytics-client-client	CN: analytics-client-client SubjectKeyId: A2:0F:4E:19:FF:72:EE:AE:02:79:4F:7F:A5:DB:A5:D2 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
analytics-ingestion-client	CN: analytics-ingestion-client SubjectKeyId: DE:63:AC:EC:13:E6:33:02:EA:47:92:BF:0D:DA:4F:9B AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
ingress-ca	CN: ingress-ca SubjectKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
mgmt-db-ca	CN: mgmt-db-ca SubjectKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
portal-client	CN: portal-client SubjectKeyId: FC:8A:06:09:DE:48:13:5B:07:75:C3:DC:3A:2C:95:9D AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
root-ca	CN: root-ca SubjectKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7 AuthorityKeyId:	

```
Subsystem mgmt_subsys certificates
```

```
=====
```

Name	Summary	Validation errors
----	-----	-----
api-manager-ui	CN: api-manager-ui SubjectKeyId: F7:96:78:02:BE:01:83:C4:DF:42:A9:87:94:AB:1D:35 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
cloud-admin-ui	CN: cloud-admin-ui SubjectKeyId: AC:13:32:C2:6D:7B:46:6D:67:B5:41:6E:92:42:D3:4C AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
consumer-api	CN: consumer-api SubjectKeyId: DE:84:86:40:4F:1E:30:A6:16:0E:CD:5B:8E:0C:1A:46 AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F	
db-client	CN: db-client SubjectKeyId: 8B:77:9B:F9:DD:26:0E:49:7E:E0:7A:81:76:CD:7F:88 AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
db-server	CN: db-server SubjectKeyId: 13:E0:7E:D5:D4:03:6F:9C:10:02:9D:09:59:37:9D:AC AuthorityKeyId: D2:22:29:08:09:D4:12:FC:BF:28:30:E8:12:84:3D:66	
encryption-secret	A9:F3:28:0E:1E:AA:D9:5E:86:02:A4:95:69:83:94:30	
mgmt-ca	CN: mgmt-ca SubjectKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75 AuthorityKeyId: D8:CE:4E:79:35:44:EB:1D:55:1B:36:E7:C6:47:F8:F7	
migration-client	CN: migration-client SubjectKeyId: 51:21:E0:E1:A8:1E:F7:C6:F2:1E:EC:6C:F5:45:A8:66	

```

platform-api      AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
                  CN: platform-api
                  SubjectKeyId: AC:EF:88:78:01:A1:4D:8E:95:95:7D:3E:C5:43:F9:48
service-client    AuthorityKeyId: 6E:5A:6C:82:BA:F2:62:CF:B3:85:54:23:B6:26:9A:3F
                  CN: service-client
                  SubjectKeyId: AA:8E:08:FC:8B:84:76:D2:B3:88:15:54:0D:F2:54:76
service-plugin    AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
                  CN: service-plugin
                  SubjectKeyId: D3:89:FE:A0:8C:8B:AF:08:4F:18:F2:A6:39:CF:F3:73
service-server    AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75
                  CN: service-server
                  SubjectKeyId: 6B:CD:BC:99:34:AF:50:D2:95:BF:0C:FD:82:94:E4:D7
                  AuthorityKeyId: 29:9C:F7:69:68:E4:1C:FC:D6:CE:83:26:3E:20:12:75

```

4. The `apicup certs generate` command generates and sets default certificates. The `generate` command only generates and sets a certificate if it is not already set; it only sets the missing default certificates that have not been explicitly set using the `set` command. Execute the `generate` command before running the `apicup subsys install <SUBSYS>` command to confirm the certificates are correct before installing. It allows you to validate all certificates before performing the installation. The `generate` command is used as a tool to assist you when entering a combination of default and custom certificates. If you need to set specific certificates you can set them upfront (using `set`) and then generate the missing ones with default certificates. Or you can generate all certificates upfront and then override specific certificates to set custom certificates. Using `generate` helps to avoid validation errors during the installation procedure. Note that you must configure the subsystems and pass the `--validate` option before generating the default certificates.

```

apicup certs generate -help
Generate all unset certificates

```

```

Usage:
  apicup certs generate SUBSYS [flags]

```

```

Flags:
  -h, --help      help for generate
Global Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging

```

Table 4. apicup certs generate

Command	Values	Result
<code>apicup certs generate SUBSYS [flags]</code>	Parameters are: <ul style="list-style-type: none"> <li>SUBSYS - name of the subsystem for which you want to generate certificates</li> </ul>	Generates certificates that have not been set for the subsystem. Generates self-signed certificates.
<code>flags</code> <ul style="list-style-type: none"> <li><code>--help</code></li> </ul>	Flags are: <ul style="list-style-type: none"> <li><code>--help</code> - Displays help for the command.</li> </ul>	Help text is displayed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Validation reference

Certificates are validated using several parameters.

The validations described in the following table are applied for all certificates, but are most helpful for custom certificates. For default certificates, the certificate validations will always pass, as the required elements are generated by APICUP. However, with custom certificates, some of the required elements may be missing or incorrect.

Validation	Messages	Error	See also/Action
Verify the certificate is set properly.	<code>certificate &lt;cert&gt; not set</code>	The certificate is not set.	
	<code>unable to load cert &lt;cert&gt;</code>	The certificate is set but cannot be read.	
Verify certificate key usage (Extended Key Usage).	<code>unable to verify cert &lt;cert&gt;: missing key usage &lt;n&gt;</code>	The certificate is missing the required key usage.	See <a href="#">Certificate reference</a> to see more information, including the type, for all certificates. See <a href="#">Setting custom certificates</a> for tips on how to generate the EKUs for custom certificates.
Verify the certificate signing CA. If available, the CA file is loaded. Then the certificate is verified against the provided CA file, including enforcement of Extended Key Usage.	<code>unable to parse CA to verify cert &lt;cert&gt;</code>	The CA file could not be parsed and loaded.	
	<code>unable to verify cert &lt;cert&gt;</code>	The certificate failed verification against the provided CA file.	One possible reason for receiving this error is that the correct EKU is missing. For a custom certificate, see <a href="#">Setting custom certificates</a> for information on generating EKUs.
Verify certificate hosts. The certificate must be valid for the hosts listed for the certificate in the Requirements column in the Certificates Reference.	<code>unable to verify cert &lt;cert&gt;: missing &lt;host&gt;</code>	The certificate is not valid for the required host.	See <a href="#">Certificate reference</a> for the required hosts.

Validation	Messages	Error	See also/Action
Verify that a certificate that is being used as a CA is actually a CA.	unable to verify cert <cert>: certificate is not a CA	The certificate is not a valid CA.	
Verify client certificate match. The portal-client, analytics-client-client, and analytics-ingestion-client certificates are verified against the CA of, respectively, portal-admin-ingress, analytics-client-ingress, and analytics-ingestion-ingress.	a CA certificate must be provided for this certificate	The CA certificate is missing for one of the portal-admin-ingress, analytics-client-ingress, and analytics-ingestion-ingress.	The common certificates portal-client, analytics-client-client, and analytics-ingestion-client must be set prior to setting any custom certificates.
	client cert cannot be verified against provided CA certificate	The verification failed.	

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tips and tricks for using APICUP

The APICUP installer in Install Assist contains built-in time saving functions.

### Introduction

This section describes tips and techniques for working with the APICUP installation commands. The APICUP installer creates charts and secrets that are then managed by Helm.

- [Running Tiller with APICUP](#)
- [Entering multiple settings per command](#)
- [Entering multiple values](#)
- [Viewing the settings for a subsystem](#)
- [Validating the settings for a subsystem](#)
- [Output an installation plan](#)
- [Getting help for commands](#)
- [Getting help on a specific command](#)

### Running Tiller with APICUP

To run Tiller in a namespace:

```
export TILLER_NAMESPACE=apic
```

### Entering multiple settings per command

To configure multiple settings per command, enter a space between each setting and enter an equals sign (=) to configure the setting.

In the following examples, be sure to replace [spc] with an actual space.

**You can set multiple database parameters on one line:**

```
apicup subsys set mgmt cassandra-max-memory-gb=9[spc]cassandra-cluster-size=3[spc]cassandra-volume-size-gb=16[spc]cassandra-backup-protocol=sftp
```

**You can set all endpoints on one line:**

```
apicup subsys set mgmt platform-api=my.platform.com[spc]api-manager-ui=my.apim.com[spc]cloud-admin-ui=my.cloud.com[spc]consumer-api=my.consumer.com
```

### Entering multiple values

Separate multiple values for a key/value pair with commas.

```
apicup subsys set mgmt dns-servers=8.8.8.8,4.4.4.4 search-domains=a.com,b.com
```

### Viewing the settings for a subsystem

To view the current values that are set for a subsystem, enter the following command: `apicup subsys get <subsys_name>`. For example: `apicup subsys get mgmt` outputs the current values for the subsystem named `mgmt` and provides a description of each value. The output from the `get` command is organized into Kubernetes settings and Subsystem settings. Following is an example:

Figure 1. Output for the `get` command

```
Kubernetes settings
=====
```



```

Name          Value          Description
----          -
extra-values-file (Optional) Path to additional configuration yml file
ingress-type   ingress        Ingress type (use `route` for OpenShift)
mode           standard      mode
namespace      default       K8S namespace to install into
registry       Docker image registry to use
registry-secret Docker registry credentials secret
storage-class  -            K8S storage class for persistent storage

Subsystem settings
=====

Name          Value          Description
----          -
cassandra-backup-auth-pass (Optional) Server password for DB backups
cassandra-backup-auth-user (Optional) Server username for DB backups
cassandra-backup-host (Optional) FQDN for DB backups server
cassandra-backup-path (Optional) path for DB backups server
cassandra-backup-port (Optional) Server port for DB backups
cassandra-backup-protocol (Optional) Protocol for DB backups (sftp/ftp)
cassandra-backup-schedule (Optional) Cron schedule for DB backups
cassandra-cluster-size Size of DB cluster (min 3 for HA)
cassandra-max-memory-gb Memory limit for DB
cassandra-volume-size-gb Size of DB storage volume (not resizable)
create-crd     true          Create DB cluster CRDS (Required cluster admin privilege)
external-cassandra-host (Optional) Hostname of externally hosted DB

Endpoints
=====

Name          Value          Description
----          -
api-manager-ui FQDN of API manager UI endpoint
cloud-admin-ui FQDN of Cloud admin endpoint
consumer-api   FQDN of consumer API endpoint
platform-api   FQDN of platform API endpoint

Error: Subsystem validation failure. Run with --validate to see details

```

## Validating the settings for a subsystem

The values set for a subsystem can be validated for syntax by entering the `--validate` option for the `get` command: `apicup subsys get <subsys_name> --validate`. For example: `apicup subsys get mgmt --validate` validates the current values for the subsystem named `mgmt`. In the following example of the output for the `--validate` option, the check mark indicates a valid setting. The `x` indicates an invalid setting with an error message provided.

Figure 2. Output for the `--validate` option

```

Subsystem settings
=====

Name          Value          Description
----          -
cassandra-backup-auth-pass ✓
cassandra-backup-auth-user ✓
cassandra-backup-host ✓
cassandra-backup-path /backups ✓
cassandra-backup-port 22 ✓
cassandra-backup-protocol sftp ✓
cassandra-backup-schedule 0 0 * * * ✓
cassandra-cluster-size 1 ✓
cassandra-max-memory-gb 9 ✓
cassandra-volume-size-gb 50 ✓
create-crd true ✓
external-cassandra-host ✓

Endpoints
=====

Name          Value          Description
----          -
api-manager-ui x api-manager-ui is not a valid hostname
cloud-admin-ui x cloud-admin-ui is not a valid hostname
consumer-api x consumer-api is not a valid hostname
platform-api x platform-api is not a valid hostname

```

## Output an installation plan

Using APICUP, you can generate an installation plan and confirm it is correct prior to running the installation. You can then install the subsystem from the plan.

To output an installation plan, enter the following command:

```
apicup subsys install SUBSYS --out=install-plan
```

where `<install-plan>` is the name of the directory where the installation plan will be stored.

In the example, a directory named install-plan is created in the project directory. The myProject/install-plan directory contains the configuration parameters for the subsystem.

To install the subsystem from the install-plan, enter the following command from the project directory:

```
apicup subsys install SUBSYS --plan-dir=<full-path-to-plan-directory>
```

where <full-path-to-plan-directory> is the full qualified path to the plan directory.

Following are general rules for installing from the installation plan:

- Never edit the files in the output directory directly. Instead, if changes are needed, update the parameters using APICUP and generate a new plan.
- Always run APICUP commands from the original project directory created during the initial product installation. The project directory contains the apiconnect-up.yml file.
- You can generate the plan in any location, but the install command must be run from the project directory. Enter the full path to the plan as the argument to `--plan-dir` to perform an installation.

The plan must be current with the project and the certificates. If the plan is older than the last modification date of the project or certificates, you will receive an error message such as:

```
the project was modified since the plan was generated, regenerate plan or skip this check
with a --no-verify flag in the command
```

or

```
the certs were changed since the plan was generated, regenerate plan or skip this check with
a --no-verify flag in the command
```

## Getting help for commands

---

You can get help for all commands by entering: `apicup --help`. Following is an example of the output:

Figure 3. Help for Install Assist apicup commands

```
APIConnect Install Assist
Usage:
  apicup [command]

Available Commands:
  certs           Subsystem certificates
  completion      Generates bash or zsh completion scripts
  help            Help about any command
  hosts           Commands to configure subsystem hosts
  iface           Commands to configure hosts interfaces
  init            Create a new APIConnect UP project
  registry-upload Retag and upload images to custom registry
  server          Starts the APIConnect cluster operator
  subsys          Subsystem commands
  version         Get the APIConnect UP version

Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging
  -h, --help        help for apicup

Use "apicup [command] --help" for more information about a command.
```

## Getting help on a specific command

---

For help on a specific command, enter `--help` after the command. For example, `apicup subsys get mgmt --help` prints out the usage and flags for the `get` command. For example:

Figure 4. Help for get command

```
Get subsystem properties

Usage:
  apicup subsys get SUBSYS [flags]

Flags:
  --endpoints  List endpoints (default true)
  -h, --help   help for get
  --platform   List platform settings (default true)
  --subsystem  List subsystem settings (default true)
  --validate   Validate settings

Global Flags:
  --accept-license  Accept the license for API Connect
  --debug           Enable debug logging
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deploying to a VMware environment

The Install Assist tool provides a simplified installation for the OVA files into a VSphere environment.

### About this task

---

The Install Assist tool contains the **APICUP** installation utility program, which provides an automated installation process for API Connect. This section contains the download and installation procedure.

- [Requirements for deploying on VMware](#)  
Ensure that your deployment uses supported versions of all software, and meets the firewall requirements.
- [Load balancer configuration in a VMware deployment](#)  
When deploying API Connect for High Availability, it is recommended that you configure a cluster with at least three nodes and a load balancer. A sample configuration is provided for placing a load balancer in front of your API Connect OVA deployment.
- [Configuring remote logging for a VMware deployment](#)  
Appliance-based (OVA) deployments use Syslog for collecting logs. Logs must be collected for both running and terminated containers and processes. Logging collection is required for IBM Support to assist with troubleshooting.
- [First steps for deploying in a VMware environment](#)  
The first steps in deploying in a VMware environment are to obtain the API Connect distribution files and to create a project directory.
- [Deploying the Management subsystem in a VMware environment](#)  
You can create a virtual server by deploying the relevant IBM® API Connect OVA file on a VMware virtual server. Create all of the virtual servers that you want to use in your cloud.
- [Deploying the Analytics subsystem in a VMware environment](#)  
You can add analytics data collection by deploying the IBM API Connect Analytics OVA file on a VMware virtual server.
- [Deploying the Developer Portal in a VMware environment](#)  
You create a Developer Portal node by deploying the Developer Portal OVA template. After you deploy the Developer Portal OVA template, you can install the Developer Portal.
- [Access the Cloud Manager and begin API Connect Cloud Configuration](#)  
When all subsystems are deployed, use the admin account to access the Cloud Manager administrative console and begin configuration.
- [Configuring API Connect subsystems in a cluster on VMware](#)  
This topic describes how to configure a cluster of API Connect subsystems (management server, analytics, and Developer Portal) with three VMs for each subsystem, for use with a load balancer, to support a high availability (HA) environment.
- [Installing the IBM License Metric Tool for VMware](#)  
The IBM License Metric Tool (ILMT) helps you assess if you are compliant with licensing requirements.
- [Adding a static route on a virtual machine](#)  
You can add a static route to the routing table when deploying API Connect on a VMware virtual machine.
- [Configuring use of an external NTP server](#)  
You must configure an external NTP server for use by API Connect when deploying on a VMware virtual machine.
- [Deploying DataPower Gateway](#)  
API Connect uses IBM DataPower® Gateway to provide the gateway service.
- [Installing the toolkit](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Requirements for deploying on VMware

Ensure that your deployment uses supported versions of all software, and meets the firewall requirements.

- [Deployment overview for endpoints and certificates](#)  
Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.
- [Firewall requirements on VMware](#)  
Diagram for port configuration, and list of active ports, for an IBM® API Connect deployment on VMware.
- [Requirements for initial deployment on VMware](#)  
Review the requirements and considerations for deploying in a VMware environment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deployment overview for endpoints and certificates

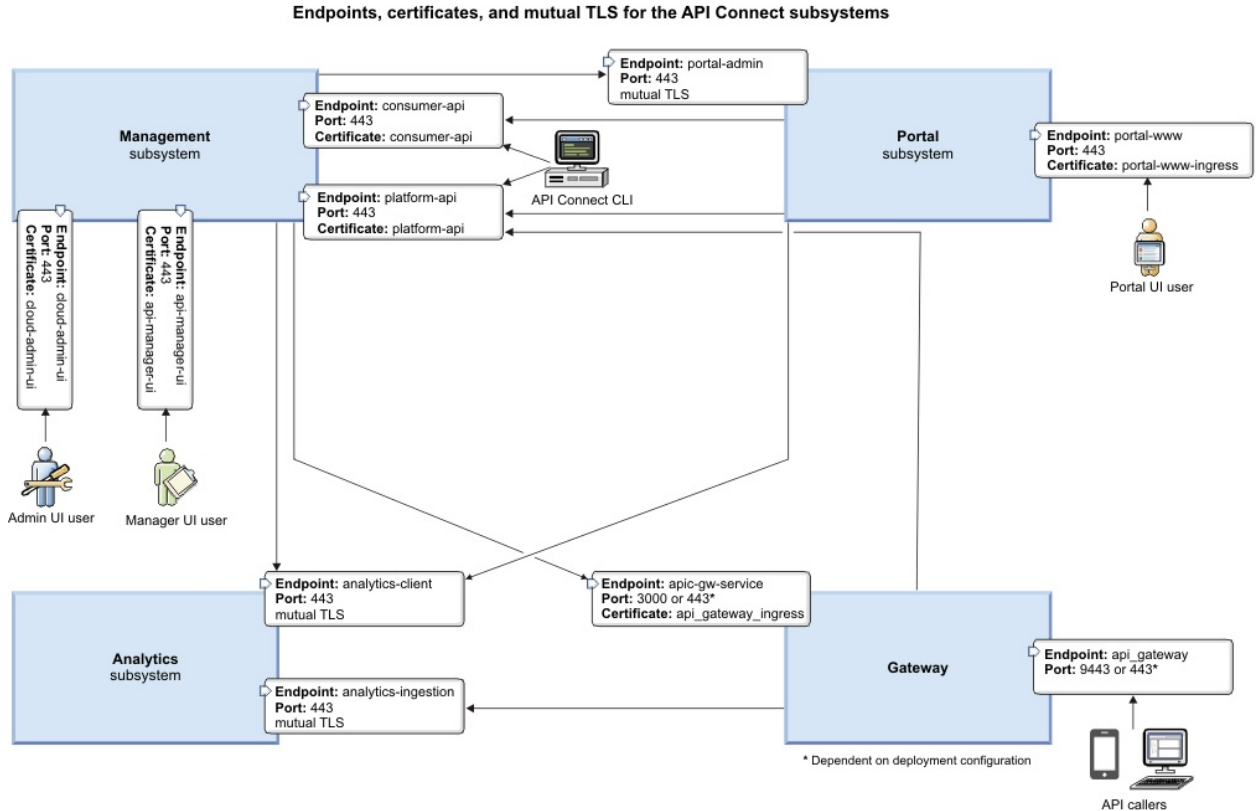
Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.

### Introduction

---

When deploying API Connect, you will create one or more endpoints for the subsystems and then configure certificates or mutual TLS for most endpoints. [Figure 1](#) shows the endpoints for each subsystem by name, the name of the certificate that secures the endpoint, and whether mutual TLS is required. It also shows the ports consumed by the endpoints, which are standard for HTTP and HTTPS.

Figure 1. Deployment Overview diagram



## Configuring endpoints

The endpoints are configured by the Install Assist program using the APICUP installer. They are set for each subsystem. Endpoints are also entered when configuring the Topology for the Gateway, Portal, and Analytics subsystems in Cloud Manager.

For instructions on configuring endpoints and installing into an OVA environment, see [Deploying to a VMware environment](#).

Subsystem	Endpoints	Description	Certificates
Management	cloud-admin-ui	Configured using APICUP installer. Endpoint on the management server for communication with the Cloud Manager user interface.	cloud-admin-ui
	api-manager-ui	Configured using APICUP installer. API Manager URL endpoint on the management server for communication with the API Manager user interface.	api-manager-ui
	consumer-api	Configured using APICUP installer. Platform REST API endpoint for running consumer APIs on the management server.	consumer-api
	platform-api	Configured using APICUP installer. Platform REST API endpoint for running admin and provider APIs on the management server.	platform-api
Portal	portal-admin	Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS.	mutual TLS
	portal-www	Configured using APICUP installer. Portal Web site URL entered in Cloud Manager. Used publicly to access Portal.	portal-www-ingress
Analytics	analytics-client	Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS.	mutual TLS
	analytics-ingestion	Configured using APICUP installer. The analytics-ingestion endpoint is used by the Gateway service to push data to the Analytics service. Requires a TLS profile configured with mutual TLS.	mutual TLS
Gateway	apic-gw-service	Configured using APICUP installer. This is the endpoint the gateway uses for network communication. Enter this endpoint as the Management Endpoint entered in Cloud Manager.	apic-gw-service-ingress
	api-gateway	Configured using APICUP installer. This is the endpoint the gateway uses for API traffic. Enter this endpoint as the API Invocation Endpoint in Cloud Manager.	api-gateway-ingress

## Configuring certificates

The certificates are configured by the Install Assist program using the APICUP installer. The certificates for the endpoints are usually configured as custom certificates as described in [Setting custom certificates](#).

## Configuring mutual TLS

Mutual TLS is configured for TLS profiles in Cloud Manager. See [Creating a TLS Server Profile](#).

## Configuring a proxy

If a Developer Portal is deployed externally to the management server zone, it does not have access to the consumer and product APIs. You need to configure a proxy to enable communication. For more information, see [Configuring a proxy](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Firewall requirements on VMware

Diagram for port configuration, and list of active ports, for an IBM® API Connect deployment on VMware.

### Required Ports between zones

The following network diagram example helps to explain which ports must be configured in an API Connect network. Specific ports must be configured to enable the communication between the various zones, both public and private, in a network.

The ports specified in the diagram are default ports. Check your deployment to understand which communication, if any, is configured to use non-default ports.

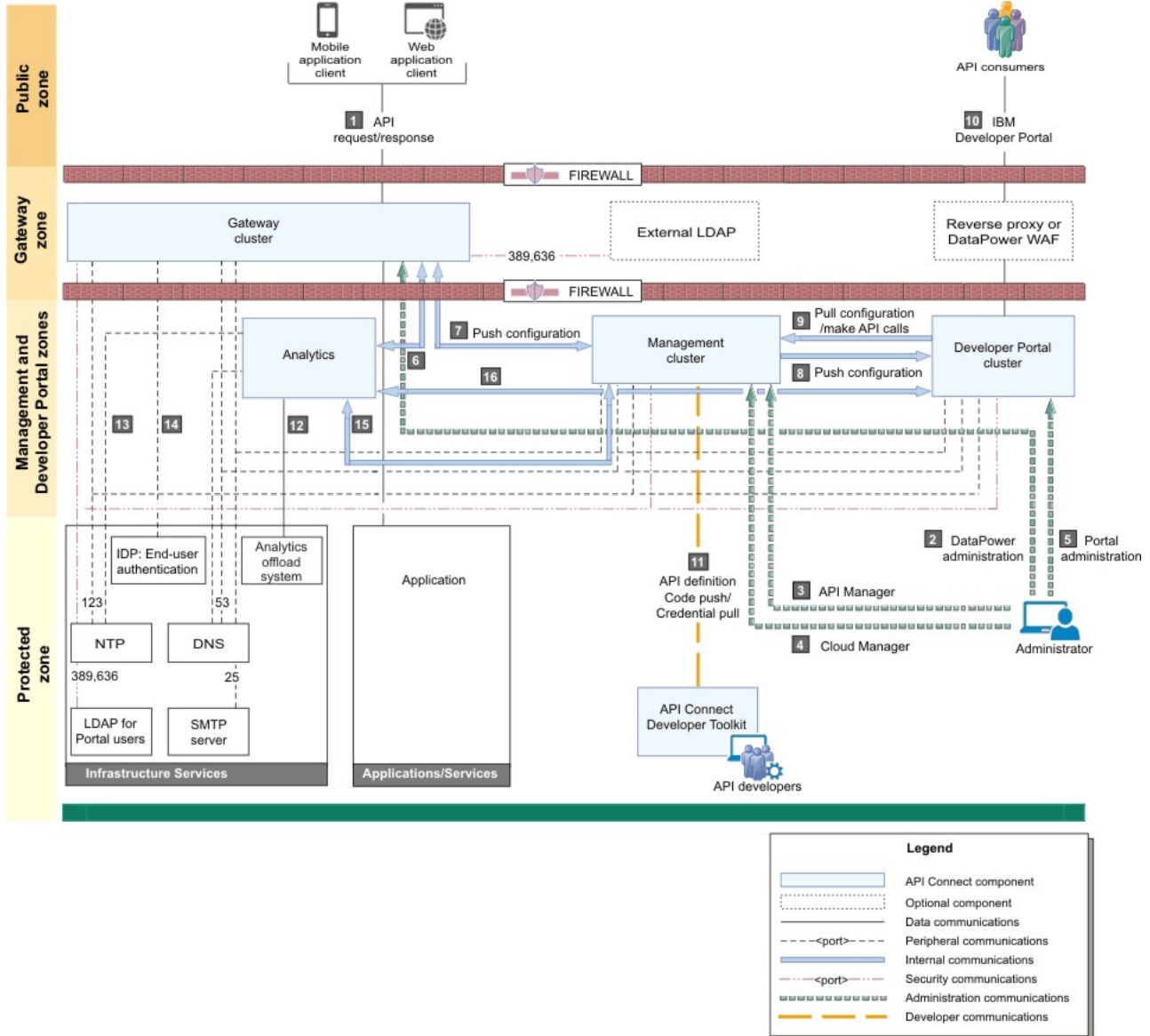


Table 1. Key for the network diagram example. The following table lists the port numbers with a usage description.

	Usage description	Default port number
1	API request/response – Users invoking the provided APIs.	443 HTTPS from Public zone to Gateway zone
2	DataPower® administration – Internal operators who are managing the Gateway servers.	22 SSH, 9090 HTTPS from Protected zone to Gateway zone
3	API Manager – Internal business users who are defining and monitoring APIs.	443 HTTPS from Protected zone to Management zone
4	Cloud Manager – Internal operators who are administering the Cloud.	22 SSH, 443 HTTPS from Protected zone to Management zone
5	Developer Portal administration – Internal operators who are managing the Portal servers.	22 SSH, 443 HTTPS from Protected zone to Management zone
6	Gateway servers post traffic to Analytics service.	443 HTTPS from Gateway servers to Analytics service
7	Push configuration – Management servers communicate bi-directionally with Gateway servers.	3000 or 443 (dependent on configuration) HTTPS Management servers to and from Gateway servers for webhook delivery
8	Push configuration/webhooks – Management servers push configuration and webhooks to the Developer Portal.	443 HTTPS Management servers to Developer Portal servers for webhook delivery
9	Pull configuration/make API calls – Developer Portal servers pull configuration and call REST APIs.	443 HTTPS from Developer Portal servers to Management servers within Management zone
10	Developer Portal – External developers who are accessing the Developer Portal.	443 HTTPS from Public zone to Developer Portal management zone. The reverse proxy/DataPower WAF for incoming web traffic to the Developer Portal cluster must be a transparent proxy - no modification of the portal URL, port, host name or path is allowed. For more information, see <a href="#">Configuring a proxy</a> .
11	Push API definition to Management server. Pick up credential for microservice code push.	443 HTTPS from Protected zone to Management zone
12	Analytics offload	Port will depend on type of plugin and protocol used for the offload. Some possible protocols are: HTTP, HTTPS, TCP, UDP, KAFKA
13	Analytics accesses NTP	Standard NTP
14	Analytics access DNS	Standard DNS
15	Management service queries Analytics service	443 HTTPS within Management zone
16	The Portal service invokes an API (GET) on the Analytics service to retrieve data.	443 HTTPS within Management zone

## Firewall port requirements on VMware

The following tables lists ports that must be open in both cluster and non-clustered deployments.

Note that clustered VMware environment deployments require additional ports. See [Firewall enabled ports for clustered OVA deployments](#)

Table 2. Firewall port requirements common to all subsystems

Subsystem	Ports and description
Ports that must be open on all API Connect subsystems	<p>The following ports must be open on the Management Server, Analytics, and Developer Portal subsystems, whether in a cluster or not.</p> <ul style="list-style-type: none"> <li>• 53 (outbound) DNS</li> <li>• 123 (outbound) NTP</li> <li>• 179 (inbound and outbound) BIRD routing daemon - VMware environments (OVA) deployments. Used for communication over TCP between servers either within the same subsystem or across different subsystems.</li> <li>• 443 (inbound and outbound) Used by all subsystems for communication with other subsystems</li> <li>• 2020 (inbound) status request. Used for communication over TCP between servers within the same subsystem.</li> <li>• 9177 (inbound and outbound) Member list (group membership) for API Connect <b>apic</b> daemon communication. Used for communication over both TCP and UDP, between servers within the same subsystem.</li> <li>• 9178 (inbound and outbound) API server, for API Connect <b>apic</b> daemon communication. Used for communication over TCP from between servers, both within the same subsystem and across different subsystems. Also used for communication between the <b>apicup</b> configuration utility and servers.</li> </ul>

Each subsystem uses ports in addition to the ports in [Table 2](#). See the following table.

Table 3. Additional firewall port requirements for each subsystem

Subsystem	Ports and description
-----------	-----------------------

Subsystem	Ports and description
Management Service	<p>Management Service uses the ports in <a href="#">Table 2</a> plus:</p> <ul style="list-style-type: none"> <li>22 remote backup server port (inbound, configurable) If backups are configured to use another port, ensure that the port is open. See <a href="#">Backing up the management subsystem in VMware environments</a>.</li> </ul> <p>Note: this port does not have to be open between the management server and the portal server.</p> <ul style="list-style-type: none"> <li>SMTP server port (outbound), typically 25</li> <li>161 (inbound) SNMP. Over UDP.</li> <li>LDAP server port (if using LDAP user registry), typically 389 (outbound)</li> <li>In a clustered deployment, the Management Services uses additional ports. See <a href="#">Firewall enabled ports for clustered OVA deployments</a>.</li> </ul>
Developer Portal	<p>The Developer Portal uses the ports listed in <a href="#">Table 2</a>, plus:</p> <ul style="list-style-type: none"> <li>22 - A remote backup server port (inbound, configurable) If backups are configured to use another port, ensure that the port is open. See <a href="#">Backing up and restoring the Developer Portal in a Kubernetes environment</a>.</li> </ul> <p>Note: this port does not have to be open between the management server and the portal server.</p> <ul style="list-style-type: none"> <li>In a clustered deployment, the Developer Portal uses additional ports. See <a href="#">Firewall enabled ports for clustered OVA deployments</a>.</li> </ul>
Analytics	<p>The Analytics subsystem uses the ports listed in <a href="#">Table 2</a>, plus:</p> <ul style="list-style-type: none"> <li>161 for SNMP is required for both non-clustered and clustered deployments. UDP, inbound only.</li> <li>In a non-clustered deployment, no additional ports are required.</li> <li>Analytics supports an optional configuration for offload of data. This configuration might require additional outbound ports to be open.</li> <li>In a clustered deployment, the Analytics subsystem uses additional ports. See <a href="#">Firewall enabled ports for clustered OVA deployments</a>.</li> </ul>
Gateway Server	<p>The Gateway Server uses these ports in both non-clustered and clustered deployments:</p> <ul style="list-style-type: none"> <li>161 (inbound) SNMP. UDP protocol. Not needed if SNMP not enabled.</li> <li>162 (outbound) SNMP traps</li> <li>3000 (inbound) Gateway Service local port (configurable)</li> <li>5550 (inbound) XML management port (configurable)</li> <li>5554 (inbound) REST management port (if enabled; configurable)</li> <li>9022 (inbound) Gateway SSH (if enabled; configurable)</li> <li>9090 (inbound) Web GUI console (if enabled; configurable)</li> <li>9443 (inbound) Gateway local port (configurable)</li> <li>In a clustered deployment, the Gateway Server uses additional ports. See <a href="#">Firewall enabled ports for clustered OVA deployments</a>.</li> </ul>

## Communications inside the Gateway cluster

There are a number of important points to note regarding the communications within the Gateway cluster.

- We advise that you use the same port for all Gateway servers within a cluster.
- Gateway servers communicate with each other to synchronize invocation counts.
- All Gateway servers in a Gateway cluster must be able to reach all of the other Gateway servers in the same Gateway cluster.
- Gateway servers in a Gateway cluster do not directly communicate with Gateway servers in a different Gateway cluster.
- All Gateway servers must be able to reach the management subsystem platform API endpoint, which was configured during the installation of your API Connect environment.

## Ethernet interface usage

To separate network traffic, you can use two or more Ethernet interfaces on the DataPower appliance on which a Gateway server is installed. For example, you can use one interface for internal IBM API Connect communications, and another for processing incoming API calls.

- [Firewall enabled ports for clustered OVA deployments](#)  
In a clustered OVA deployment of API Connect, specific ports must be configured for communication between members of each API Connect subsystem.
- [Load balancer ports for clustered OVA deployments](#)  
When you deploy a load balancer with a clustered deployment, in a VMware environment, you must ensure that the necessary ports are open.

## Related concepts

- [Firewall enabled ports for clustered OVA deployments](#)
- [Load balancer ports for clustered OVA deployments](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Firewall enabled ports for clustered OVA deployments



In a clustered OVA deployment of API Connect, specific ports must be configured for communication between members of each API Connect subsystem.

OVA deployments require the common ports listed in [Firewall requirements on Kubernetes](#). When the VMs are clustered, additional ports are used for communication between the members of the subsystems in the cluster.

All ports must be enabled inbound and outbound.

Table 1. Firewall enabled ports for clustered VMware environment deployments

Subsystem	Ports
Ports that must be open between all subsystem VMs	442, 2379, 2380, 6443, 6444, 9099, 10248, 10249, 10250, 10251, 10252, 10254, 10256, 10257, 10259 These are ports that must be open between all servers within a given subsystem. For example, from management server to management server, or from portal server to portal server, or from analytics server to analytics server. These ports are not used for communication between subsystems.  You might need additional TCP ports for Kubernetes-proxied services. The default range is 30000 – 32767. Since the ports in use can change dynamically, ensure that the default range is open.
Additional ports that must be open between Management Service VMs	7001, 7199, 8778, 9042
Additional ports that must be open between Developer Portal VMs	3009, 3010, 3306, 3307, 4443, 4444, 4567, 4568, 30865
Additional ports that must be open between Gateway Service VMs	16380, 16381, 26380, 26381
Additional ports that must be open between Analytics VMs	No additional ports are needed.

These internal ports are not used for communication between VMs. Ensure that they are open on the VM server locally.

Table 2. Internal ports reserved by API Connect

Subsystem	Ports
Reserved local ports on all subsystem VMs	8080, 30000:59999
Management Service VMs	2000, 2001, 3003, 3004, 3006, 3007, 8084, 8404, 8443
Portal Service VMs	3058, 3059, 3060, 3061
Analytics VMs	4443, 9200, 9300

Note: The ports should support IP-in-IP (protocol 4), per <https://docs.projectcalico.org/getting-started/kubernetes/requirements>.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Load balancer ports for clustered OVA deployments

When you deploy a load balancer with a clustered deployment, in a VMware environment, you must ensure that the necessary ports are open.

The load balancer needs port 443 for all API Connect subsystems (Management, Analytics, Developer Portal, DataPower Gateway).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for initial deployment on VMware

Review the requirements and considerations for deploying in a VMware environment.

### Deployment requirements on VMware

- Ensure you have supported software requirement versions. See [IBM API Connect Version 2018 software product compatibility requirements](#)
- Do not change the hardware version of the OVA during installation. Do not attempt to use an unsupported version, even if VMWare indicates compatibility with other versions. For example, when deploying IBM API Connect, the VMWare UI for the APIC OVAs might show information like:

Table 1.

Property	Value
Guest OS	Ubuntu Linux (64-bit)
Compatibility	ESXi 5.5 and later (VM version 10)
VMware Tools	Yes
CPUs	4
Memory	16 GB



Although the **Compatibility** field shows **ESXi 5.5 and later (VM version 10)**, API Connect supports only the versions listed in [IBM API Connect Version 2018 software product compatibility requirements](#). Do not change the VM version of the OVA. Ensure that the **Compatibility** field values are not changed, and remain **ESXi 5.5 and later (VM version 10)**.

Attempts to modify the VMware compatibility may typically result in failure to boot the OVA, as per <https://kb.vmware.com/s/article/52683>.

- Ensure that your operating system has one of the supported utilities for creating ISOs. The apicup installer uses **mkisofs** on Linux, and **hdiutil** on macOS. For Windows, you need software that creates ISO files using **mkisofs**, such as CDRTools. Verify that the utility you will use is located in a directory that is referenced by the PATH environment setting for your operating system. When creating the ISO, if you encounter the message **Error: unable to create config ISO for host**, verify that you have sufficient permissions to run the command.
- Verify you have access to [Passport Advantage®](#) to download the Install Assist package and the latest IBM API Connect packages for your operating system.

Package	IBM API Connect subsystem file
IBM API Connect® Management for VMWare	apiconnect-management.ova
IBM API Connect Analytics for VMWare	apiconnect-analytics.ova
IBM API Connect Developer Portal for VMWare	apiconnect-portal.ova

Important:

Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

The original project directory created with APICUP during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. Note that the endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored or upgraded system. A good practice is to back up the original project directory to a location from where it can always be retrieved.

- For each subsystem, gather the following networking settings, which you will need to supply during configuration:

Table 2.

Required information	Value for your system
IP address of the server	
IP address of the server	
Domain of the server	
IP addresses of the name servers	
IP address of the network gateway (not DataPower® gateway) for the server	
Name of the Ethernet interface	
VLAN	

Some virtualization environments require additional information when you create and configure virtual machines. For example, it might be necessary to assign a specific VLAN ID, Resource Pool, or Datastore. Please refer to information provided by your virtualization environment administrators.

## Configuration on VMware

- API Connect cannot be deployed on NFS.
- The timezone for API Connect pods is set to UTC. In API Connect deployments on VMware, the operating system timezone is also set to UTC. Do not change the timezone for the pods or the operating system.
- Ensure that the time settings match (within a few seconds) between the machine running **apicup** and the VMware **Host** clock. To verify the VMware host clock setting, see <https://kb.vmware.com/s/article/1003736>.

If the clocks are too different, the installation can fail because of invalid certificates due to time discrepancies.

- API Connect requires a dedicated IP range for deployment of the Kubernetes pod and Kubernetes service networks. These IP addresses cannot conflict with IP addresses used by other resources in your deployment, such as SMTP servers or user registries. The default values are 172.16.0.0/16 and 172.17.0.0/16, respectively. If a /16 subnet overlaps with existing IPs on the network, a CIDR as small as /22 is acceptable. If the default ranges conflict with other programs, you can modify the API Connect ranges during initial installation. Note that you cannot modify them once an appliance has been deployed. Follow the instructions for your subsystem if you want to modify either of the IP ranges.

- Only static IP addresses that are specified during the **apicup** project configuration before the installation of the OVAs are supported.
- Designated host names must have wildcard aliases or host aliases, which ensures that the different endpoints work together. For example, *\*.hostname.mycompany.com*.
- Kubernetes ingress limits the character set for DNS names to not support the underscore character "**\_**". This means you cannot specify underscores in domain names that are used as endpoints. For example, *my\_domain.bar.com* and *my.domain\_abc.com* are not supported for **<xxx>.<hostname>.<domainname>**, and will cause an error. For example:

```
Invalid value: "my_domain.bar.com": a DNS-1123 subdomain must consist of lowercase alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character (e.g. 'example.com', regex used for validation is '[a-z0-9]([-a-z0-9]*[a-z0-9])?(\.[a-z0-9]([-a-z0-9]*[a-z0-9])?)*)')
```

- Note that you cannot change host names or DNS names on a running cluster.
- **Version 2018.4.1.0 only:** Ensure that Reverse DNS lookup is configured for the host names and endpoints for each subsystem. Be sure that a ping of the subsystem host names and endpoints resolves to the corresponding IP address.

```
nslookup <ip_address>
ping <*.hostname.example.com>
```

Note: Reverse DNS lookup is not required for Version 2018.4.1.1 or later.

- For the management subsystem, the installer sets a default value of 9GB for **cassandra-max-memory-gb**. The minimum value for this setting is 9GB. The following table compares values for memory allocation for the management database:

Memory allocated to OVA node	Recommended cassandra-max-memory-gb	MAX_HEAP_SIZE (47% of cassandra-max-memory-gb value)
16GB	9GB (default)	4331m
32GB	16GB	7700m
64GB	24GB	11550m

- The allocation of memory by the JVM is more than just heap size, so where the previous table shows the maximum heap size can grow by up to 4331m, there is also more memory that can be requested by the JVM from the operating system. The ratio of 47% that is used in the table provides sufficient padding so that under

normal conditions the JVM should not request more than 9GB in total from the operating system. If the JVM did request more memory than the ratio padding allows, the limit in place on the Kubernetes pod would lead Kubernetes to kill the process/container so that it can be restarted, as per the design of setting memory limits on pods in Kubernetes.

- For Analytics, the installer specifies a default value of 16 GB for `es-max-memory-gb`. The minimum value for this setting is 12 GB.
- Decide the *mode* to use for the installation.

Table 3. Deployment modes

Mode	Description
<b>dev</b>	Development mode ( <b>dev</b> ) deploys a subsystem with the scale of one. Development mode is recommended for development, testing, and demonstration purposes.  <code>apicup subsys set mgmt mode=dev</code>  Do not use <b>dev</b> mode for production environments. Development mode does not provide high availability.
<b>standard</b>	Standard mode ( <b>standard</b> ) deploys in high availability mode for a production environment.  <code>apicup subsys set mgmt mode=standard</code>

Usage:

- Mode is set for each subsystem type: Management, Analytics, Developer Portal, and Gateway.
- If you do not specify the mode, a default mode is applied, as follows:
  - **Version 2018.4.1.4 and later:** **dev** mode
  - **Version 2018.4.1.3 and earlier:** **standard** mode
- Use of **standard** mode is supported only on installations with three or more nodes. Installations with less than three nodes must use **dev** mode. If you install in **standard** mode with only a single node, some pods can remain in a pending state. To avoid having pods remain pending, either install in **dev** mode or add additional nodes.
- For additional considerations for configuring clusters, see [Configuring API Connect subsystems in a cluster on VMware](#). Important: For best performance it is recommended that the network latency between any 2 nodes be as low as possible. Do not configure nodes from the same subsystem cluster across multiple data centers with a high latency network. A high latency network is one that experiences more than 30ms latency between nodes. For more information, see the [API Connect V2018 Whitepaper](#)
- Configure a remote server for logging. Follow these instructions: [Configuring remote logging for a VMware deployment](#). Pods are terminated during an upgrade, and the logs will be lost if not stored remotely.

## Passwords and certificates

- When creating configuration files for use in generating an ISO image for VMware, ensure that your working directories are secure. The VMware configuration requires an ISO image that contains a plain text password to be used to unlock the VMware data disk. This means that the API Connect project configuration file `apiconnect-up.yml`, and the `/user-data` directory for each host, contain the passwords you specified. This configuration information is used to create the ISO image that you combine with the `.ova` distribution files when deploying (unlocking) the VMware data disk.
- You can use `apicup` to specify an `ssh` keyfile that contains a public certificate for using `ssh` to log in to a virtual machine. Logging in through `ssh` is preferred because it is more secure than password-based login.
- Default certificates are generated for each subsystem by the `apicup subsys install` command. If certificates are not explicitly set using the `apicup certs set` command, then default certificates are automatically generated by `apicup`. The default certificates are self-signed, so they may not provide a level of trust suitable for external communication. See [Working with certificates](#).

## Setting and using a hashed default password

During configuration of the Management, Analytics, and Developer Portal subsystems, you create a password to use to log in to the management console for the first time. You must use a password hashing utility to hash the password. You then use `apicup` to assign this hashed password to the subsystem. These configuration steps ensure that the password is not stored in plain text on the data disk.

The syntax of the `apicup` command is:

```
apicup subsys set mgmt default-password='hashed_password'
```

Usage notes:

- The `default-password` is for the `apicadm` user account on the Appliance.
- The password for `apicadm` can be used **only** to log in through the VMware console. You cannot use it to `ssh` into the Appliance as an alternative to using the `ssh-keyfiles`. Interactive login for `ssh` is disabled.
- The `default-password` value configured is only used during initial installation (first boot) of each virtual appliance. Changing the value, then regenerating the ISO, and attaching the new ISO to the virtual appliance does not change the `apicadm` password.
- The `default-password` must be hashed. If it is plain text, you will not be able to log into the Appliance through the VMware console. When you use `apicup` to set or `get default-password`, `apicup` ensures that the hash type of the password is one of the following:
  - MD5
  - SHA1
  - SHA256
  - SHA512
  - BCRYPT
  - MD5-Crypt
- When using `apicup` to set a default password for a subsystem, be aware of syntax differences between operating systems. Windows requires double-quotes. Linux and OSX require single quotes.

Operating system	Command syntax
Linux or OSX	<code>apicup subsys set mgmt default-password='hashed_password'</code>
Windows	<code>apicup subsys set mgmt default-password="hashed_password"</code>

- You can use the `passwd` command (on the appliance) to change the `apicadm` password.
- When using the VMware Remote console to login to the appliance, be aware that the keyboard layout is English. This can cause a problems with hashed passwords, if you created the ISO on a system with a different keyboard layout and you used special characters or symbols. Passwords are hashed when you set the password that you enter to log into your Management appliance for the first time, and when you create hosts.

## DataPower Gateway for API Connect on VMware

- The Management server and Gateway firmware versions must match, for example, API Connect 2018.4.1.3 and DataPower 2018.4.1.3.
- Installation and configuration of DataPower Gateway on an appliance (physical or virtual) is completed after you install the API Connect subsystems. For the gateway service in a VMware environment, use the instructions in [Deploying DataPower Gateway](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Load balancer configuration in a VMware deployment

When deploying API Connect for High Availability, it is recommended that you configure a cluster with at least three nodes and a load balancer. A sample configuration is provided for placing a load balancer in front of your API Connect OVA deployment.

### About this task

API Connect can be deployed on a single node cluster. In this case the ingress endpoints are host names for which the DNS resolution points to the single IP address of the corresponding node hosting a particular subsystem, and no load balancer is required. For high availability, it is recommended to have at least a three node cluster. With three nodes, the ingress endpoints cannot resolve to a single IP address. A load balancer should be placed in front of an API Connect subsystem to route traffic.

Because it is difficult to add nodes once endpoints are configured, a good practice is to configure a load balancer even for single node deployments. With the load balancer in place, you can easily add nodes when needed. Add the node to the list of servers pointed to by the load balancer to avoid changing the ingress endpoints defined during the installation of API Connect.

To support Mutual TLS communication between the API Connect subsystems, configure the load balancer with **SSL Passthrough** and **Layer 4** load balancing. In order for Mutual TLS to be performed directly by the API Connect subsystems, the load balancer should leave the packets unmodified, as is accomplished by Layer 4. Following is a description of the communication between the endpoints that are configured with Mutual TLS:

- API Manager (with the client certificate portal-client) communicates with the Portal Admin endpoint portal-admin (with the server certificate portal-admin-ingress)
- API Manager (with the client certificate analytics-client-client) communicates with the Analytics Client endpoint analytics-client (with the server certificate analytics-client-ingress)
- API Manager (with the client certificate analytics-ingestion-client) communicates with the Analytics Ingestion endpoint analytics-ingestion (with the server certificate analytics-ingestion-ingress)

Set endpoints to resolve to the load balancer

When configuring a load balancer in front of the API Connect subsystems, the ingress endpoints are set to host names that resolve to a load balancer, rather than to the host name of any specific node. For an overview of endpoints, see [Deployment overview for endpoints and certificates](#).

Use this example configuration as a guideline to determine the best way to configure the load balancer for your deployment.

## Procedure

### • Appliance deployment

In this example configuration, the API Connect Management, Portal, Analytics and Gateway subsystems are deployed as three node clusters in Standard mode. An HAProxy load balancer is used. The load balancer is configured with `ssl-passthrough` and with upstream selection based on SNI. DNS resolution is configured to resolve the endpoints to the IP address of the load balancer. If a single HAProxy node is used, then all endpoints must resolve to the IP address of the single HAProxy. The following example endpoints require DNS resolution for this example:

- `api-manager-ui.sample.example.com`
- `cloud-admin-ui.sample.example.com`
- `consumer-api.sample.example.com`
- `platform-api.sample.example.com`
- `admin.portal.sample.example.com`
- `web.portal.sample.example.com`
- `analytics.client.sample.example.com`
- `analytics.ingestion.sample.example.com`
- `api-gateway.sample.example.com`
- `apic-gw-service.sample.example.com`

Following is an example HAProxy configuration for one HAProxy node distributing traffic to Management and Portal clusters:

Note:

When you configure a load balancer in front of a Management subsystem, specify timeouts of at least 240 seconds. Note that large deployments might need larger values.

The default timeout is typically 50 or 60 seconds, which is not long enough to avoid **409**

**Conflict** or **504 Gateway Timeout** errors. The **409 Conflict** error can occur when the time needed to complete an operation is sufficiently long that a second request gets issued.

For example, to specify 240 seconds when using HAProxy as a load balancer, set `timeout client` and `timeout server` to `240000`.

```
# This sample HAProxy configuration file configures one HAProxy node to distribute traffic to
# Management, Portal, Analytics, and Gateway clusters. Another option is to configure one HAProxy
# node per cluster.
```

```
global
    log /dev/log      local0
```

```

log /dev/log      local1 notice
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin
stats timeout 30s
user haproxy
group haproxy
daemon

# Default SSL material locations
ca-base /etc/ssl/certs
crt-base /etc/ssl/private

# Default ciphers to use on SSL-enabled listening sockets.
# For more information, see ciphers(1SSL). This list is from:
# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:
RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS
ssl-default-bind-options no-sslv3

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout connect 5000
    timeout client 240000
    timeout server 240000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

##### MANAGEMENT CONFIGURATION #####
frontend fe_management
    mode tcp
    option tcplog
    #
    # Map to the hostname and TCP port for the Management load balancer.
    # In this example, the hostname for the load balancer is ubuntu.sample.example.com.
    #
    bind ubuntu.sample.example.com:443
    tcp-request inspect-delay 5s
    tcp-request content accept if { req_ssl_hello_type 1 }

    #
    # The value for the Management endpoints as defined in the apiconnect-up.yml
    # file using the apicup installer. In this example, the endpoints are api-manager-ui.sample.example.com,
    # cloud-admin-ui.sample.example.com, consumer-api.sample.example.com, and
    # platform-api.sample.example.com. Standard SNI structure specifies
    # whether the INCOMING request is for api-manager or cloud-admin or for consumer-api or platform-api
    # then use "be_management".
    #
    use backend be_management if
    { req_ssl_sni -i api-manager-ui.sample.example.com OR req_ssl_sni -i cloud-admin-ui.sample.example.com }
    use backend be_management if
    { req_ssl_sni -i consumer-api.sample.example.com OR req_ssl_sni -i platform-api.sample.example.com }

    #
    # be_management is defined to point management traffic to the cluster
    # containing three management nodes
    #

backend be_management
    mode tcp
    option tcplog
    balance roundrobin
    option ssl-hello-chk

    #
    # One entry per Management node in the cluster.
    # Hostname and TCP Port for each Management node.
    #
    server management0 manager1.sample.example.com:443 check
    server management1 manager2.sample.example.com:443 check
    server management2 manager3.sample.example.com:443 check

##### PORTAL CONFIGURATION #####
frontend fe_portal
    mode tcp
    option tcplog
    #
    # The Hostname and TCP Port for the Portal Load balancer
    #
    bind ubuntu.sample.example.com:443
    tcp-request inspect-delay 5s
    tcp-request content accept if { req_ssl_hello_type 1 }

    #
    # The value for both of the Portal subsystem endpoints as defined in the apiconnect-up.yml file
    #
    use backend be_portal if
    { req_ssl_sni -i admin.portal.sample.example.com OR req_ssl_sni -i web.portal.sample.example.com }

```

```

backend be_portal
  mode tcp
  option tcplog
  balance roundrobin
  option ssl-hello-chk

#
# One entry per Portal node.
# Hostname and TCP Port for the Portal node.
#
server portal0 portal1.sample.example.com:443 check
server portal1 portal2.sample.example.com:443 check
server portal2 portal3.sample.example.com:443 check

##### ANALYTICS CONFIGURATION #####
frontend fe_analytics
  mode tcp
  option tcplog
  #
  # The Hostname and TCP Port for the Analytics Load balancer
  #
  bind ubuntu.sample.example.com:443
  tcp-request inspect-delay 5s
  tcp-request content accept if { req_ssl_hello_type 1 }

#
# The value for both of the Analytics subsystem endpoints as defined in the apiconnect-up.yml file
#
use_backend be_analytics if
{ req_ssl_sni -i analytics.client.sample.example.com OR req_ssl_sni -i analytics.ingestion.sample.example.com }

backend be_analytics
  mode tcp
  option tcplog
  balance roundrobin
  option ssl-hello-chk

#
# One entry per Analytics node.
# Hostname and TCP Port for the Analytics node.
#
server analytics0 analytics1.sample.example.com:443 check
server analytics1 analytics2.sample.example.com:443 check
server analytics2 analytics3.sample.example.com:443 check

##### GATEWAY CONFIGURATION #####
frontend fe_gateway
  mode tcp
  option tcplog
  #
  # The Hostname and TCP Port for the Gateway Load balancer
  #
  bind ubuntu.sample.example.com:443
  tcp-request inspect-delay 5s
  tcp-request content accept if { req_ssl_hello_type 1 }

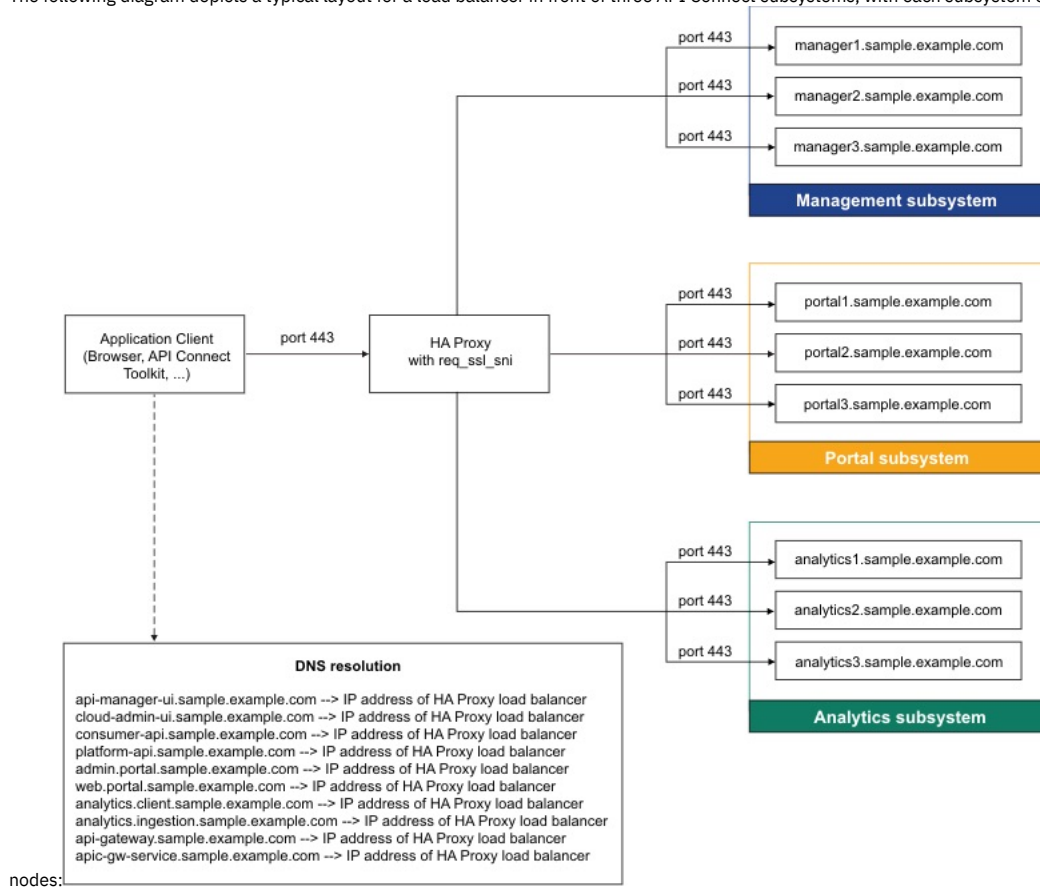
#
# The values for the Gateway subsystem endpoints as defined in the apiconnect-up.yml file.
#
use_backend be_gateway if
{ req_ssl_sni -i api-gateway.sample.example.com OR req_ssl_sni -i apic-gw-service.sample.example.com }

backend be_gateway
  mode tcp
  option tcplog
  balance roundrobin
  option ssl-hello-chk

#
# One entry per Gateway node.
# Hostname and TCP Port for the Gateway node.
#
server gateway0 gateway1.sample.example.com:443 check
server gateway1 gateway2.sample.example.com:443 check
server gateway2 gateway3.sample.example.com:443 check

```

- The following diagram depicts a typical layout for a load balancer in front of three API Connect subsystems, with each subsystem containing three Appliance/OVA



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring remote logging for a VMware deployment

Appliance-based (OVA) deployments use Syslog for collecting logs. Logs must be collected for both running and terminated containers and processes. Logging collection is required for IBM Support to assist with troubleshooting.

### About this task

For appliance-based deployments using OVA, rsyslog is used to collect log files.

There are three options for collecting logs:

- Set the rsyslog configuration at boot up (this is the best practice in order to preserve settings when rebooting)
- Set the rsyslog configuration post-boot up
- Gather logs from a running system

For more information about rsyslog, see: <https://www.rsyslog.com/doc/v8-stable/configuration/index.html>

### Procedure

Log collection from the Syslog collectors is accomplished using the rsyslog module included in the cloudinit application. Cloud-init is an open source package used for injecting configurations during boot up. It is included with the appliance-based (OVA) installation. Cloud-init uses a configuration file, for example, config\_file.yaml, to configure logging and other customizations. The complete list of configuration customizations that can be made with cloud-init is documented in <https://cloudinit.readthedocs.io/en/latest/topics/modules.html>. Follow these steps to configure logging at boot up:

- Enter the rsyslog configuration in the config\_file.yaml file.
  - Following is an example of the options for rsyslog in config\_file.yaml:

```
rsyslog:
  remotes:
    syslog_server1: "192.168.1.1:514"
    syslog_server2: "10.0.4.1:601"
```

where:

- **remotes** are key pairs specifying the remote Syslog collectors from which you want to collect logs. Each key is the name for an rsyslog remote entry. Each value holds the remote IP address and port for the Syslog collector.
- **syslog\_serverX** are Syslog collectors (514 = TCP, 601=UDP)

Note:

A number of Appliance containers such as the **kube-apiserver** log by default to stderr, which is interpreted by rsyslog as a high severity message. To avoid having non-error log entries be presented as errors in rsyslog, it is recommend to configure a format that defaults the severity of the message to info level, and rely on further parsing of the log messages to highlight higher severity messages. Example configuration:

```
rsyslog:
  configs:
  - "*" * @192.168.1.187;myformat"
  - filename: 00-myformat.conf
    content: |
      template(name="myformat" type="string"
        string="<${myprio}> %TIMESTAMP::date-rfc3339% %HOSTNAME% %syslogtag% %msg::sp-if-no-1st-sp%%msg%")

      set $.myseverity = 6;
      set $.myprio = 20*8+$.myseverity;
```

where the logs would be forwarded to the server **192.168.1.187** using the format **myformat**. Further scanning of the log lines could be defined in the logging infrastructure to highlight the higher severity log entries by, for example, scanning for occurrences of **Error/Err/ERROR**.

- Enter the following command to configure log collection at boot up before starting installation:

```
apicup subsys set mgmt additional-cloud-init-file config_file.yaml
```

Refer to the rsyslog documentation for more information. See <https://www.rsyslog.com/doc/v8-stable/configuration/index.html>

- To collect logs post boot up, follow these steps:
  - ssh** into each VM and change directories to `etc/rsyslog.d`.
  - Create a new `.conf` file, for example, `<new>-cloud-config.conf`.
  - Restart rsyslog with the following command: **systemctl restart rsyslog**
- To gather logs from a running system, enter the following command:

```
sudo apic logs
```

The **apic logs** command collects the following information for the current VM:

- Logs for all containers (including terminated ones)
- Everything from **journalctl**
- All system service logs

The information is stored locally on the current VM. Entries are culled based upon age for a maximum size of 2 GB (compressed).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## First steps for deploying in a VMware environment

The first steps in deploying in a VMware environment are to obtain the API Connect distribution files and to create a project directory.

### Before you begin

- Review [Requirements for initial deployment on VMware](#).
- See also [Configuring API Connect subsystems in a cluster on VMware](#).

### About this task

In these First Steps, you will use Install Assist (**apicup**) to create a project directory that contains the configuration file **apiconnect-up.yaml**. When you configure the first subsystem, such as the Management Service, the subsystem configuration settings are placed in the file.

When you want to configure a second subsystem, it is important that you reuse the same project directory. This means that you only need to download Install Assist once, and create a project directory once. The configuration settings for the second subsystem are added to the existing **apiconnect-up.yaml**. In this way, the second subsystem, such as Developer Portal, can access the configuration information needed to interact with the first subsystem.

After completing these First Steps, follow the links to instructions for configuring your subsystem. You will specify configuration settings, create an ISO of the subsystem configuration, and then use VMware to deploy the distribution file (.ova file) for the subsystem with your ISO. Note that you create a separate ISO for each subsystem.

Installation and configuration of DataPower® Gateway on an appliance (physical or virtual) is completed after you install the API Connect subsystems. For the gateway service in a VMware environment, use the instructions in [Deploying DataPower Gateway](#).

Note: When maintaining API Connect, do not use **kubectl exec** commands to access API Connect pods unless advised by IBM.

Do not make any changes on the deployed VMs unless documented here or otherwise advised by IBM. Attempting to manually update packages, adding new users, or installing new software will likely cause problems. Operating system updates are handled by API Connect fix packs.

### Procedure

1. Download the latest IBM API Connect® package from [IBM Fix Central](#).

Package	IBM API Connect subsystem file
IBM API Connect Management for VMWare	apiconnect-management.ova and the Developer Toolkit
IBM API Connect Analytics for VMWare	apiconnect-analytics.ova
IBM API Connect Developer Portal for VMWare	apiconnect-portal.ova

Note that the IBM [Passport Advantage](#) site contains the API Connect LTS release, 2018.4.1.0, but not any packages that contain later Fix Packs. The Fix Pack distributions, such as 2018.4.1.5, contain the entire product. For a new installation, you do not need to first install the LTS version, 2018.4.1.0, or any of the fix packs (2018.4.1.x) prior to the latest one. For example, if you want to install 2018.4.1.5 as a new installation, you do not need to first install 2018.4.1.0, 2018.4.1.1, 2018.4.1.2, 2018.4.1.3 or 2018.4.1.4.

Note also that the Fix Pack page contains two copies of the files for each subsystem. One copy is for new installations and one is for upgrades of existing installations. For new installations, do not use files with names with the prefix `upgrade-`. Use the file names specified in the previous table. In addition, be aware that during upgrades, there are prerequisites to meet when moving between specific versions. For upgrades, do not use the procedures on this page. See [Upgrading in a VMware environment](#).

- Select one of the following actions:
  - If you are configuring your first subsystem, go to step [3](#).
  - If you have already configured a subsystem, go to step [7](#).
- Download the IBM API Connect Install Assist package for your operating system from [IBM Fix Central](#). It contains the `apicup` file, which simplifies installation of the API Connect components.
- OSX and Linux® only: Make the `apicup` file an executable file by entering the following command:

```
chmod +x apicup
```

- Set your path to the location of your `apicup` file.
  - For the OSX and Linux operating systems:

```
export PATH=$PATH:/Users/your_path/
```

- For the Windows operating system:

```
set PATH=c:\your_path;%PATH%
```

- Create a project called `myProject` and optionally copy the `apicup` executable into the project directory.
  - Create a project:

```
apicup init myProject
```

- Read and accept the license agreement.

- Optionally, copy the `apicup` executable into the `myProject` directory or folder created by the `apicup init` command.

Important:

Use a single APICUP project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

The original project directory created with APICUP during the initial product installation (for example, `myProject`) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. Note that the endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored or upgraded system. A good practice is to back up the original project directory to a location from where it can always be retrieved.

The `apiconnect-up.yml` file is created in that directory.

Important: The `apiconnect-up.yml` file must be in a secure and permanent location. The file contains password information and other information that is exposed in text format. Ensure that the project directory is secure.

- Continue with the configuration steps for your subsystem:
  - [Deploying the Management subsystem in a VMware environment](#)
  - [Deploying the Analytics subsystem in a VMware environment](#)
  - [Deploying the Developer Portal in a VMware environment](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying the Management subsystem in a VMware environment

You can create a virtual server by deploying the relevant IBM® API Connect OVA file on a VMWare virtual server. Create all of the virtual servers that you want to use in your cloud.

### Before you begin

Before you deploy:

- Review the [Deployment requirements on VMware](#).
- Review the [Configuration on VMware](#).
- For information on deploying a cluster, see [Configuring API Connect subsystems in a cluster on VMware](#)
- If you are upgrading from a previous version, see [Upgrading in a VMware environment](#).

### About this task

You must deploy the API Connect OVA template to create each Management virtual server that you want in your cloud.



## Procedure

1. Ensure that you obtained the distribution file and have a project directory, as described in [First steps for deploying in a VMware environment](#).
2. Change to the project directory.

```
cd myProject
```

3. Create a management subsystem.

```
apicup subsys create mgmt management
```

Where:

- `mgmt` is the name of your management server that you are creating. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character.
- `management` indicates that you are creating a management microservice.

The API Connect Helm charts are deployed into the default namespace. You do not need to specify a namespace.

Tip: At any time, you can view the current management subsystem values in the `apiconnect-up.yml` by running the `apicup subsys get` command:

```
apicup subsys get mgmt
```

If you have not yet configured the subsystem, the command might return errors. Also, if you have not updated the value, a default value is listed, if there is one that is available.

After configuration is complete, you can view output similar to the following sample:

```
apicup subsys get mgmt
```

```
Appliance settings
```

```
=====
```

Name	Value	Description
-----	-----	-----
additional-cloud-init-file		(Optional) Path to additional cloud-init yml file
data-device	sdb	VM disk device (usually `sdb` for SCSI or `vdb` for VirtIO)
default-password	\$6\$rounds=4096\$imCJ9cfhFJ8X\$pbm19ClWzcYzHZFoQ6n7OnYCF/owQZiICpAtWazs/FUn/uE8uLD.9jwHE0AX4upFSqx/jf0ZmDbHPZ9bU1CY1	(Optional) Console login password for `apicadm` user
dns-servers	[1.2.136.11]	List of DNS servers
k8s-pod-network	172.16.0.0/16	(Optional) CIDR for pods within the appliance
k8s-service-network	172.17.0.0/16	(Optional) CIDR for services within the appliance
mode	standard	
public-iface	eth0	Device for API/UI traffic (Eg: eth0)
search-domain	[subnet1.example.com]	List for DNS search domains
ssh-keyfiles	[/home/vsphere/.ssh/id_rsa.pub]	List of SSH public keys files
traffic-iface	eth0	Device for cluster traffic (Eg: eth0)
license-version	Production	

```
Subsystem settings
```

```
=====
```

Name	Value	Description
-----	-----	-----
az-name	default-az	Availability Zone name
cassandra-backup-auth-pass		(Optional) Server password for DB backups
cassandra-backup-auth-user		(Optional) Server username for DB backups
cassandra-backup-host		(Optional) FQDN for DB backups server
cassandra-backup-path	/backups	(Optional) path for DB backups server
cassandra-backup-port	22	(Optional) Server port for DB backups
cassandra-backup-protocol	sftp	(Optional) Protocol for DB backups (sftp/ftp/objstore)
cassandra-backup-schedule	0 0 * * *	(Optional) Cron schedule for DB backups
cassandra-max-memory-gb	9	Memory limit for DB
cross-az-peers	[ ]	(Optional) IP addresses of nodes in other AZs

```
Endpoints
```

```
=====
```

Name	Value	Description
-----	-----	-----
api-manager-ui	api-manager-ui.testsrv0231.subnet1.example.com	FQDN of API manager UI endpoint
cloud-admin-ui	cloud-admin-ui.testsrv0231.subnet1.example.com	FQDN of Cloud admin endpoint
consumer-api	consumer-api.testsrv0231.subnet1.example.com	FQDN of consumer API endpoint
platform-api	platform-api.testsrv0231.subnet1.example.com	FQDN of platform API endpoint

4. For production environments, specify `mode=standard`.

```
apicup subsys set mgmt mode=standard
```

The `mode=standard` parameter indicates that you are deploying in high availability (HA) mode for a production environment. If the mode parameter is omitted, the subsystem deploys by default in `dev` mode, for use in development and testing. For more information, see [Requirements for initial deployment on VMware](#).

5. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set mgmt  
license-version=<license_type>
```

The `license_type` must be either **Production** or **Nonproduction**. If not specified, the default value is **Nonproduction**.

6. Optional: Configure scheduled backups of the subsystem. This step is optional but is recommended. Note that once you set up scheduled backups, you can also run backups on-demand. Refer to the instructions for scheduled backups in [Backing up the management subsystem in VMware environments](#).
7. Optional: Configure your logging.
  - Logging can be configured at a later time, but you must enable it before installation to capture the log events from the installation.
  - a. Complete the procedure at [Configuring remote logging for a VMware deployment](#).
  - b. Enter the following command to create the log file:

```
apicup subsys set mgmt additional-cloud-init-file=config_file.yml
```

8. Enter the following commands to update the `apiconnect-up.yml` with the information for your environment:
  - a. Set your search domain. Multiple search domains should be separated by commas.

```
apicup subsys set mgmt search-domain=your_search_domain
```

Where `your_search_domain` is the domain of your servers, entered in all lowercase. Setting this value ensures that your searches also append these values, which are based on your company's DNS resolution, at the end of the search value. A sample search domain is `mycompany.example.com`.

Ensure that the value for `your_search_domain` is resolved in the system's `/etc/resolv.conf` file to avoid "502" errors when accessing the Cloud Manager web site. For example:

```
# Generated by resolvconf
search your_search_domain ibm.com other.domain.com
```

- b. Set your domain name servers (DNS).
    - Supply the IP addresses of the DNS servers for your network. Use a comma to separate multiple server addresses.

```
apicup subsys set mgmt dns-servers=ip_address_of_dns_server[,ip_address_of_another_dns_server_if_necessary]
```

DNS entries may not be changed on a cluster after the initial installation.

- c. Use `apicup` to set your endpoints.
    - You can use wildcard aliases or host aliases with your endpoints.
    - Optionally, you can specify all endpoints with one `apicup` command. See [Tips and tricks for using APICUP](#).

Note: You cannot specify the underscore character "\_" in domain names that are used in endpoints. See [Configuration on VMware](#).

Table 1. Management subsystem endpoints

Setting	Endpoint host description
<code>platform-api</code>	Platform API endpoint. The host where your platform API calls are routed. <code>apicup subsys set mgmt platform-api=platform-api.hostname.domain</code>
<code>consumer-api</code>	Consumer API endpoint. The host where your consumer API calls are routed. <code>apicup subsys set mgmt consumer-api=consumer-api.hostname.domain</code>
<code>cloud-admin-ui</code>	Cloud admin user interface API endpoint. The host where your cloud administrator user-interface API calls are routed. <code>apicup subsys set mgmt cloud-admin-ui=cloud-admin-ui.hostname.domain</code>
<code>api-manager-ui</code>	API Manager user interface endpoint. The host where your API Manager API calls are routed. <code>apicup subsys set mgmt api-manager-ui=api-manager-ui.hostname.domain</code>

9. Set a Public key.

```
apicup subsys set mgmt ssh-keyfiles=path_to_public_ssh_keyfile
```

Setting this key enables you to use `ssh` with this key to log in to the virtual machine to check the status of the installation. You will perform this check in step [29](#) of these instructions.

10. Set the password that you enter to log into your Management appliance for the first time.
  - a. **Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).
  - b. If you do not have a password hashing utility, install one.

Operating system	Command
Ubuntu, Debian, OSX	If the <code>mkpasswd</code> command utility is not available, download and install it. (You can also use a different password hashing utility.) On OSX, use the command: <code>gem install mkpasswd</code> .
Windows, Red Hat	If necessary, a password hashing utility like OpenSSL.

- c. Create a hashed password

Operating system	Command
Ubuntu, Debian, OSX	<code>mkpasswd --method=sha-512 --rounds=4096 password</code>
Windows, Red Hat	For example, using OpenSSL: <code>openssl passwd -1 password</code> . Note that you might need to add your password hashing utility to your path; for example, in Windows: <code>set PATH=c:\cygwin64\bin;%PATH%</code>

- d. Set the hashed password for your subsystem:

```
apicup subsys set mgmt default-password='hashed_password'
```

11. Optional: If the default IP ranges for the API Connect Kubernetes pod and the service networks conflict with IP addresses that must be used by other processes in your deployment, modify the API Connect values.
 

You can change the IP ranges of the Kubernetes pod and the service networks from the default values of 172.16.0.0/16 and 172.17.0.0/16, respectively. In the case that a /16 subnet overlaps with existing IPs on the network, a Classless Inter-Domain Routing (CIDR) as small as /22 is acceptable. You can modify these ranges during initial installation and configuration only. You cannot modify them once an appliance has been deployed. See [Configuration on VMware](#).

- a. Update the IP range for the Kubernetes pod

```
apicup subsys set mgmt k8s-pod-network='new_pod_range'
```

Where *new\_pod\_range* is the new value for the range.

- b. Update the IP range for Service networks.

```
apicup subsys set mgmt k8s-service-network='new_service_range'
```

Where *new\_service\_range* is the new value for the range.

12. Add your hosts.

```
apicup hosts create mgmt hostname.domainname hd_password
```

Where the following are true:

- *hostname.domainname* is the fully qualified name of the server where you are hosting your Management service, including the domain information.
- *hd\_password* is the password that the Linux Unified Key Setup uses to encrypt the storage for your Management service. This password is hashed when it is stored on the server or in the ISO. Note that the password is base64 encoded when stored in `apiconnect-up.yml`.

Repeat this command for each host that you want to add.

Note:

- Host names and DNS entries may not be changed on a cluster after the initial installation.
- **Version 2018.4.1.0:** Ensure that Reverse DNS lookup configuration is configured for the host names.

```
nslookup <ip_address>
```

For **Version 2018.4.1.1 or later**, Reverse DNS lookup is not required.

13. Create your interfaces.

```
apicup iface create mgmt hostname.domainname physical_network_id host_ip_address/subnet_mask gateway_ip_address
```

Where *public\_iface\_id* is the network interface ID of your physical server. The value is most often `eth0`. The value can also be `ethx`, where *x* is a number identifier.

The format is similar to this example: `apicup iface create mgmt myHostname.domain eth0`

```
192.0.2.10/255.255.255.0 192.0.2.1
```

14. Optional: Use `apicup` to view the configured hosts:

```
apicup hosts list mgmt
testsrv0231.subnet1.example.com
  Device  IP/Mask  Gateway
  eth0    1.2.152.231/255.255.254.0  1.2.152.1
```

Note: This command might return the following messages, which you can ignore:

```
* host is missing traffic interface
* host is missing public interface
```

15. Optional: Verify that the configuration settings are valid.

```
apicup subsys get mgmt --validate
```

The output lists each setting and adds a check mark after the value once the value is validated. If the setting lacks a check mark and indicates an invalid value, reconfigure the setting. See the following sample output.

```
apicup subsys get mgmt --validate
Appliance settings
=====
```

Name	Value	
additional-cloud-init-file		✓
data-device	sdb	✓
default-password	\$6\$rounds=4096\$iMCJ9cfhFJ8X\$pbm A19C1WzcYzHZFoQ6n7OnYCF/owQZiIcPAtWazs/ FUu/uE8uLD.9jwHE0AX4upFSqx/jf0ZmDbHPZ9bU1CY1	✓
dns-servers	[1.2.136.11]	✓
k8s-pod-network	172.16.0.0/16	✓
k8s-service-network	172.17.0.0/16	✓
mode	standard	✓
public-iface	eth0	✓
search-domain	[subnet1.example.com]	✓
ssh-keyfiles	[/home/vsphere/.ssh/id_rsa.pub]	✓
traffic-iface	eth0	✓
license-version	Production	✓

```
Subsystem settings
=====
```

Name	Value	
az-name	default-az	✓
cassandra-backup-auth-pass		✓
cassandra-backup-auth-user		✓
cassandra-backup-host		✓
cassandra-backup-path	/backups	✓
cassandra-backup-port	22	✓
cassandra-backup-protocol	sftp	✓
cassandra-backup-schedule	0 0 * * *	✓
cassandra-max-memory-gb	9	✓
cross-az-peers	[ ]	✓

## Endpoints

=====

Name	Value
api-manager-ui	api-manager-ui.testsrv0231.subnet1.example.com ✓
cloud-admin-ui	cloud-admin-ui.testsrv0231.subnet1.example.com ✓
consumer-api	consumer-api.testsrv0231.subnet1.example.com ✓
platform-api	platform-api.testsrv0231.subnet1.example.com ✓

16. Create your ISO file.

```
apicup subsys install mgmt --out mgmtplan-out
```

The `--out` parameter and value are required.

In this example, the ISO file is created in the `myProject/mgmtplan-out` directory.

If the system cannot find the path to your software that creates ISO files, create a path setting to that software by running a command similar to the following command:

Operating system	Command
OSX and Linux	<code>export PATH=\$PATH:/Users/your_path/</code>
Windows	<code>set PATH="c:\Program Files (x86)\cdrtools";%PATH%</code>

17. Log into the VMware vSphere Web Client.
18. Using the VSphere Navigator, navigate to the directory where you are deploying the OVA file.
19. Right-click the directory and select Deploy OVF Template.
20. Complete the Deploy OVF Template wizard.
  - a. Select the `apiconnect-management.ova` template by navigating to the location where you downloaded the file from Passport Advantage®.
  - b. Enter a name and location for your file.
  - c. Select a resource for your template.
  - d. Review the details for your template.
  - e. Select the size of your configuration.
  - f. Select the storage settings.
  - g. Select the networks.
  - h. Customize the Template, if necessary.
  - i. Review the details to ensure that they are correct.
  - j. Select Finish to deploy the virtual machine.

Note: Do not change the OVA hardware version, even if the VMware UI shows a Compatibility range that includes other versions. See [Requirements for initial deployment on VMware](#).

The template creation appears in your Recent Tasks list.

21. Select the Storage tab in the Navigator.
22. Navigate to your datastore.
23. Upload your ISO file.
  - a. Select the Navigate to the datastore file browser icon in the icon menu.
  - b. Select the Upload a file to the Datastore icon in the icon menu.
  - c. Navigate to the ISO file that you created in your project.

It is the `myProject/mgmtplan-out`
  - d. Upload the ISO file to the datastore.
24. Leave the datastore by selecting the VMs and Templates icon in the Navigator.
25. Locate and select your virtual machine.
26. Select the Configure tab in the main window.
27. Select Edit....
  - a. On the Virtual Hardware tab, select CD/DVD Drive 1.
  - b. For the Client Device, select Datastore ISO File.
  - c. Find and select your datastore in the Datastores category.
  - d. Find and select your ISO file in the Contents category.
  - e. Select OK to commit your selection and exit the Select File window.
  - f. Ensure that the Connect At Power On check box is selected.

Tip:

- Expand the CD/DVD drive 1 entry to view the details and the complete Connect At Power On label.
- Note that VMware related issues with ISO mounting at boot may occur if Connect At Power On

- g. Select OK to commit your selection and close the window.

28. Start the virtual machine by selecting the play button on the icon bar.

The installation might take several minutes to complete, depending on the availability of the system and the download speed.
29. Log in to the virtual machine by using an SSH tool to check the status of the installation:
  - a. Enter the following command to connect to `mgmt` using SSH:

```
ssh ip_address -l apicadm
```

You are logging in with the default ID of `apicadm`, which is the API Connect ID that has administrator privileges.

- b. Select Yes to continue connecting.

Your host names are automatically added to your list of hosts.

- c. Run the `apic status` command to verify that the installation completed and the system is running correctly.

Note that after installation completes, it can take several minutes for all servers to start. If you see the error message `Subsystems not running`, wait a few minutes, try the command again, and review the output in the `Status` column.

The command output for a correctly running Management system is similar to the following lines:

```
apicadm@testsys0181:~$ sudo apic status
```

```
INFO[0001] Log level: info
Cluster members:
```

```

- testsys0164.subnet1.example.com (1.1.1.1)
Type: BOOTSTRAP MASTER
Install stage: DONE
Upgrade stage: NONE
Docker status:
  Systemd unit: running
Kubernetes status:
  Systemd unit: running
  Kubelet version: testsys0164 (4.4.0-137-generic) [Kubelet v1.10.6, Proxy v1.10.6]
  Etcd status: pod etcd-testsys0164 in namespace kube-system has status Running
  Addons: calico, dns, helm, kube-proxy, metrics-server, nginx-ingress,
- testsys0165.subnet1.example.com (1.1.1.2)
Type: MASTER
Install stage: DONE
Upgrade stage: NONE
Docker status:
  Systemd unit: running
Kubernetes status:
  Systemd unit: running
  Kubelet version: testsys0165 (4.4.0-137-generic) [Kubelet v1.10.6, Proxy v1.10.6]
  Etcd status: pod etcd-testsys0165 in namespace kube-system has status Running
  Addons: calico, kube-proxy, nginx-ingress,
- testsys0181.subnet1.exmample.com (1.1.1.3)
Type: MASTER
Install stage: DONE
Upgrade stage: NONE
Docker status:
  Systemd unit: running
Kubernetes status:
  Systemd unit: running
  Kubelet version: testsys0181 (4.4.0-137-generic) [Kubelet v1.10.6, Proxy v1.10.6]
  Etcd status: pod etcd-testsys0181 in namespace kube-system has status Running
  Addons: calico, kube-proxy, nginx-ingress,
Etcd cluster state:
- etcd member name: testsys0164.subnet1.example.com, member id: 11019072309842691371,
  cluster id: 5154498743703662183, leader id: 11019072309842691371, revision: 21848, version: 3.1.17
- etcd member name: testsys0165.subnet1.example.com, member id: 541472388445093633,
  cluster id: 5154498743703662183, leader id: 11019072309842691371, revision: 21848, version: 3.1.17
- etcd member name: testsys0181.subnet1.example.com, member id: 3261849123413063575,
  cluster id: 5154498743703662183, leader id: 11019072309842691371, revision: 21848, version: 3.1.17

```

Pods Summary:

NODE STATUS	NAMESPACE	NAME	READY
testsys0165 Running	kube-system	calico-node-jp8zv	2/2
testsys0164 Running	kube-system	calico-node-pjjgh	2/2
testsys0181 Running	kube-system	calico-node-ssb9w	2/2
testsys0164 Running	kube-system	coredns-87cb95869-9nvdr	1/1
testsys0164 Running	kube-system	coredns-87cb95869-r9q8w	1/1
testsys0164 Running	kube-system	etcd-testsys0164	1/1
testsys0165 Running	kube-system	etcd-testsys0165	1/1
testsys0181 Running	kube-system	etcd-testsys0181	1/1
testsys0165 Running	kube-system	ingress-nginx-ingress-controller-92mkz	1/1
testsys0181 Running	kube-system	ingress-nginx-ingress-controller-kt9sr	1/1
testsys0164 Running	kube-system	ingress-nginx-ingress-controller-p7x55	1/1
testsys0164 Running	kube-system	ingress-nginx-ingress-default-backend-6f58fb5f56-t27gx	1/1
testsys0164 Running	kube-system	kube-apiserver-testsys0164	1/1
testsys0165 Running	kube-system	kube-apiserver-testsys0165	1/1
testsys0181 Running	kube-system	kube-apiserver-testsys0181	1/1
testsys0164 Running	kube-system	kube-apiserver-proxy-testsys0164	1/1
testsys0165 Running	kube-system	kube-apiserver-proxy-testsys0165	1/1
testsys0181 Running	kube-system	kube-apiserver-proxy-testsys0181	1/1
testsys0164 Running	kube-system	kube-controller-manager-testsys0164	1/1
testsys0165 Running	kube-system	kube-controller-manager-testsys0165	1/1
testsys0181 Running	kube-system	kube-controller-manager-testsys0181	1/1
testsys0165 Running	kube-system	kube-proxy-7gqpw	1/1
testsys0181 Running	kube-system	kube-proxy-8hc8t	1/1
testsys0164 Running	kube-system	kube-proxy-bhgqj	1/1
testsys0164 Running	kube-system	kube-scheduler-testsys0164	1/1
testsys0165 Running	kube-system	kube-scheduler-testsys0165	1/1

Running			
testsys0181	kube-system	kube-scheduler-testsys0181	1/1
Running			
testsys0164	kube-system	metrics-server-6fbfb84cdd-1ffxc	1/1
Running			
testsys0164	kube-system	tiller-deploy-84f4c8bb78-xxfds	1/1
Running			

30. Verify you can access the API Connect Cloud Manager. Enter the URL in your browser. The syntax is `https://<hostname.domain>/admin`. For example:

```
https://cloud-admin-ui.testsrv0231.subnet1.example.com/admin
```

The first time that you access the Cloud Manager user interface, you enter `admin` for the user name and `7iron-hide` for the password. You will be prompted to change the Cloud Administrator password and email address. See [Accessing the Cloud Manager user interface](#).

## What to do next

If you want to deploy an API Connect Analytics OVA file, continue with [Deploying the Analytics subsystem in a VMware environment](#).

If you want to deploy an API Connect Developer Portal OVA file, continue with [Deploying the Developer Portal in a VMware environment](#).

Identify the DataPower® appliances to be used as gateway servers in the API Connect cloud and obtain the IP addresses.

Define your API Connect configuration by using the API Connect cloud console. For more information, see [Defining the cloud](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying the Analytics subsystem in a VMware environment

You can add analytics data collection by deploying the IBM® API Connect Analytics OVA file on a VMware virtual server.

### Before you begin

Before you deploy:

- Review the [Deployment requirements on VMware](#).
- Review the [Configuration on VMware](#).
- For information on deploying a cluster, see [Configuring API Connect subsystems in a cluster on VMware](#)
- If you are upgrading from a previous version, see [Upgrading in a VMware environment](#).

### About this task

You must deploy the API Connect OVA template to create each Analytics virtual server that you want in your cloud.

By default, the Analytics subsystem is configured to store data so that users can review it in the Analytics user interface. After deploying the subsystem, you can optionally configure it to offload some data to a third-party service for review and storage. Any data that is not offloaded remains accessible from the Analytics user interface.

If you want to offload all Analytics data, you can optionally install the subsystem with the ingestion-only configuration. In this scenario, unused Analytics components (such as `analytics-storage` and `analytics-client`) are omitted from the topology. Only the components that are required for offloading data are deployed. The reduced topology requires less CPU, memory, and storage.

If you do not configure ingestion-only during installation, you can enable it later as explained in [Enabling Analytics ingestion-only on VMware](#).

Settings that are required for the ingestion-only configuration are noted in the steps that follow.

Attention: You must deploy the API Connect OVA template to create each Analytics virtual server that you want in your cloud.

### Procedure

1. Ensure that you obtained the distribution file and have a project directory, as described in [First steps for deploying in a VMware environment](#).
2. Change to the project directory.

```
cd myProject
```

3. Create an analytics subsystem.

```
apicup subsys create analyt analytics
```

Where:

- `analyt` is the name of the analytics server that you are creating. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character.
- `analytics` indicates that you want it to create an Analytics microservice.

The `apiconnect-up.yml` file that is in that directory is updated to add the analytics-related entries.

Tip: At any time, you can view the current analytics subsystem values in the `apiconnect-up.yml` by running the `apicup get` command:

**apicup subsys get analyt**

If you have not updated the value, a default value is listed, if there is one that is available.  
Sample output from `apicup subsys get`, after configuration is completed:

```

apicup subsys get analyt
Appliance settings
=====

Name          Value          Description
----          -
additional-cloud-init-file yml file      (Optional) Path to additional cloud-init
data-device    sdb            VM disk device (usually `sdb` for SCSI or
`vdb` for VirtIO)
default-password $6$rounds=4096$imCJ9cfhFJ8X$pbmAl
9ClWzcYzHZFoQ6n7OnYCF/owQZIIcPAtWazs/FUN/
uE8uLD.9jwHE0AX4upFSqx/jf0ZmDbHPZ9bU1CY1 (Optional) Console login password for
`apicadm` user
dns-servers    [1.2.136.11]  List of DNS servers
k8s-pod-network 172.16.0.0/16 (Optional) CIDR for pods within the
appliance
k8s-service-network 172.17.0.0/16 (Optional) CIDR for services within the
appliance
mode           standard
public-iface   eth0           Device for API/UI traffic (Eg: eth0)
search-domain  [subnet1.example.com] List for DNS search domains
ssh-keyfiles   [/home/vsphere/.ssh/id_rsa.pub] List of SSH public keys files
traffic-iface  eth0           Device for cluster traffic (Eg: eth0)
license-version Production

Subsystem settings
=====

Name          Value          Description
----          -
es-max-memory-gb 16             Memory limit for elastic search

Endpoints
=====

Name          Value          Description
----          -
analytics-client a7s-client.testsrv0233.subnet1.example.com FQDN of Analytics client/UI endpoint
analytics-ingestion a7s-in.testsrv0233.subnet1.example.com FQDN of Analytics ingestion endpoint

```

4. For production environments, specify `mode=standard`.

```
apicup subsys set analyt mode=standard
```

The `mode=standard` parameter indicates that you are deploying in high availability (HA) mode for a production environment. If the mode parameter is omitted, the subsystem deploys by default in `dev` mode, for use in development and testing. For more information, see [Requirements for initial deployment on VMware](#).

5. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set analyt
license-version=<license_type>
```

The `license_type` must be either `Production` or `Nonproduction`. If not specified, the default value is `Nonproduction`.

6. Optional: Configure your logging.

Logging can be configured at a later time, but you must enable it before installation to capture the log events from the installation.

- Complete the procedure at [Configuring remote logging for a VMware deployment](#).
- Enter the following command to create the log file:

```
apicup subsys set analyt additional-cloud-init-file=config_file.yml
```

7. Enter the following commands to update the `apiconnect-up.yml` with the information for your environment:

- Use `apicup` to set your endpoints.

You can use wildcard aliases or host aliases with your endpoints.

Optionally, you can specify all endpoints with one `apicup` command. See [Tips and tricks for using APICUP](#).

Note: You cannot specify the underscore character "\_" in domain names that are used in endpoints. See [Configuration on VMware](#).

The endpoints must be unique hostnames which both point to the IP address of the OVA (single node deployment), or to the IP of a load balancer configured in front of the OVA nodes. See examples in sample output in step [3](#).

Setting	Endpoint host description
analytics-ingestion	Your <i>unique_hostname</i> identifies the endpoint that enables the Gateway to push your analytics data. The values for the analytics-ingestion and analytics-client must be different. <code>apicup subsys set analyt analytics-ingestion=unique_hostname.domain</code>
analytics-client	Your <i>unique_hostname</i> identifies the endpoint that enables the Cloud Manager, API Manager, and Developer Portal to communicate with the Analytics subsystem. <code>apicup subsys set analyt analytics-client=unique_hostname.domain</code>
	This setting is not needed for the ingestion-only configuration.

- To configure the Analytics subsystem for ingestion-only, include the following command:

```
apicup subsys set analyt ingestion-only=true
```

- Set your search domain. Multiple search domains should be separated by commas.

```
apicup subsys set analyt search-domain=your_search_domain
```

Where *your\_search\_domain* is the domain of your servers, entered in all lowercase. Setting this value ensures that your searches also append these values, which are based on your company's DNS resolution, at the end of the search value. A sample search domain is *mycompany.example.com*.

Ensure that the value for *your\_search\_domain* is resolved in the system's */etc/resolv.conf* file to avoid "502" errors when accessing the Cloud Manager web site. For example:

```
# Generated by resolvconf
search your_search_domain ibm.com other.domain.com
```

d. Set your domain name servers (DNS).

Supply the IP addresses of the DNS servers for your network. Use a comma to separate multiple server addresses.

```
apicup subsys set analyt dns-servers=ip_address_of_dns_server
```

DNS entries may not be changed on a cluster after the initial installation.

8. Set a Public key.

```
apicup subsys set analyt ssh-keyfiles=path_to_public_ssh_keyfile
```

Setting this key enables you to use **ssh** with this key to log in to the virtual machine to check the status of the installation. You will perform this check in step 29 of these instructions.

9. You can set the password that you enter to log into your Analytics server for the first time.

a. **Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).

b. If you do not have a password hashing utility, install one.

Operating system	Command
Ubuntu, Debian, OSX	If the <b>mkpasswd</b> command utility is not available, download and install it. (You can also use a different password hashing utility.) On OSX, use the command: <b>gem install mkpasswd</b> .
Windows, Red Hat	If necessary, a password hashing utility like OpenSSL.

c. Create a hashed password

Operating system	Command
Ubuntu, Debian, OSX	<b>mkpasswd --method=sha-512 --rounds=4096 password</b>
Windows, Red Hat	For example, using OpenSSL: <b>openssl passwd -1 password</b> . Note that you might need to add your password hashing utility to your path; for example, in Windows: <b>set PATH=c:\cygwin64\bin;%PATH%</b>

d. Set the hashed password for your subsystem:

```
apicup subsys set analyt default-password='hashed_password'
```

Notes:

- The password is hashed. If it is in plain text, you cannot log into the VMWare console.
- Note that the password can only be used to login through the VMware console. You cannot use it to SSH into the Appliance as an alternative to using the **ssh-keyfiles**.
- On Linux or OSX, use single quotes around *hashed\_password*. For Windows, use double quotes.
- If you are using a non-English keyboard, understand the limitations with using the remote VMware console. See [Requirements for initial deployment on VMware](#).

10. Optional: If the default IP ranges for the API Connect Kubernetes pod and the service networks conflict with IP addresses that must be used by other processes in your deployment, modify the API Connect values.

You can change the IP ranges of the Kubernetes pod and the service networks from the default values of 172.16.0.0/16 and 172.17.0.0/16, respectively. In the case that a /16 subnet overlaps with existing IPs on the network, a Classless Inter-Domain Routing (CIDR) as small as /22 is acceptable. You can modify these ranges during initial installation and configuration only. You cannot modify them once an appliance has been deployed. See [Configuration on VMware](#).

a. Update the IP range for the Kubernetes pod

```
apicup subsys set analyt k8s-pod-network='new_pod_range'
```

Where *new\_pod\_range* is the new value for the range.

b. Update the IP range for Service networks.

```
apicup subsys set analyt k8s-service-network='new_service_range'
```

Where *new\_service\_range* is the new value for the range.

11. Add your hosts.

```
apicup hosts create analyt hostname.domainname hd_password
```

Where the following are true:

- *hostname.domainname* is the fully qualified name of the server where you are hosting your Analytics service, including the domain information.
- *hd\_password* is the password of the Linux Unified Key Setup uses to encrypt the storage for your Analytics service. This password is hashed when it is stored on the server or in the ISO. Note that the password is base64 encoded when stored in **apiconnect-up.yml**.

Repeat this command for each host that you want to add.

Note:

- Host names and DNS entries may not be changed on a cluster after the initial installation.
- **Version 2018.4.1.0:** Ensure that Reverse DNS lookup configuration is configured for the host names.



```
nslookup <ip_address>
```

For **Version 2018.4.1.1 or later**, Reverse DNS lookup is not required.

12. Create your interfaces.

```
apicup iface create analyt hostname.domainname physical_network_id host_ip_address/subnet_mask gateway_ip_address
```

Where *public\_iface\_id* is the network interface ID of your physical server. The value is most often `eth0`. The value can also be `ethx`, where `x` is a number identifier. The format is similar to this example: `apicup iface create analyt myHostname.domain eth0 192.0.2.1/255.255.1.1 192.0.2.1`

13. Optional: Use `apicup` to view the configured hosts:

```
apicup hosts list analyt
testsrv0233.subnet1.example.com
  Device IP/Mask Gateway
  eth0   1.2.152.233/255.255.254.0 1.2.152.1
```

14. Optional: Enable the message queue for analytics.

```
apicup subsys set analyt enable-message-queue=true
```

Options are `true` or `false`. The default is `false`. When set to true, the message queue will be activated and the analytics pipeline will be configured to use it. See [Configuring the analytics message queue](#).

You can enable the message queue later if you do not want to enable it during installation.

15. Verify that the configuration settings are valid.

```
apicup subsys get analyt --validate
```

The output lists each setting and adds a check mark after the value once the value is validated. If the setting lacks a check mark and indicates an invalid value, reconfigure the setting. See the following sample output.

```
apicup subsys get analyt --validate
Appliance settings
=====
Name          Value
----          -
additional-cloud-init-file  ✓
data-device   sdb ✓
default-password  $6$rounds=4096$iMCJ9cfhFJ8X$pbmAl9C1WzcYzH
                ZFoQ6n7OnYcf/owQZiIcPAtWazs/FUn/uE8uLD.9jwHE0AX4upFSqx/jf0ZmDbHPZ9bUlCY1 ✓
dns-servers    [1.2.3.1] ✓
k8s-pod-network 172.16.0.0/16 ✓
k8s-service-network 172.17.0.0/16 ✓
mode           standard ✓
public-iface   eth0 ✓
search-domain  [subnet1.example.com] ✓
ssh-keyfiles   [/home/vsphere/.ssh/id_rsa.pub] ✓
traffic-iface  eth0 ✓
license-version Production ✓

Subsystem settings
=====
Name          Value
----          -
es-max-memory-gb 16 ✓

Endpoints
=====
Name          Value
----          -
analytics-client a7s-client.testsrv0233.subnet1.example.co ✓
analytics-ingestion a7s-in.testsrv0233.subnet1.example.com ✓
```

Note: If you are installing the Analytics subsystem with the ingestion-only configuration, the following settings are not actually validated (because they are not used) but are marked as correct in the validation results:

- `analytics-client`
- `es-max-memory`

16. Create your ISO file.

```
apicup subsys install analyt --out analytplan-out
```

The `--out` parameter and value are required. In this example, the ISO file is created in the `myProject/analytplan-out/node-config` directory.

If the system cannot find the path to your software that creates ISO files, create a path setting to that software by running a command similar to the following command:

Operating system	Command
OSX and Linux	<code>export PATH=\$PATH:/Users/your_path/</code>
Windows	<code>set PATH="c:\Program Files (x86)\cdrtools";%PATH%</code>

17. Log into the VMware vSphere Web Client.

18. Using the vSphere Navigator, navigate to the directory where you are deploying the OVA file.

19. Right-click the directory and select Deploy OVF Template.

20. Complete the Deploy OVF Template wizard.

- a. Select the apiconnect-analytics.ova template by navigating to the location where you downloaded the file from [Passport Advantage®](#).
- b. Enter a name and location for your file.
- c. Select a resource for your template.
- d. Review the details for your template.
- e. Select the size of your configuration.
- f. Select the storage settings.
- g. Select the networks.
- h. Customize the Template, if necessary.
- i. Review the details to ensure that they are correct.
- j. Select Finish to deploy the virtual machine.

Note: Do not change the OVA hardware version, even if the VMware UI shows a Compatibility range that includes other versions. See [Requirements for initial deployment on VMware](#).

The template creation appears in your Recent Tasks list.

21. Select the Storage tab in the Navigator.
22. Navigate to your datastore.
23. Upload your ISO file.
  - a. Select the Navigate to the datastore file browser icon in the icon menu.
  - b. Select the Upload a file to the Datastore icon in the icon menu.
  - c. Navigate to the ISO file that you created in your project.  
It is the myProject/analytplan-out/node-config
  - d. Upload the ISO file to the datastore.
24. Leave the datastore by selecting the VMs and Templates icon in the Navigator.
25. Locate and select your virtual machine.
26. Select the Configure tab in the main window.
27. Select Edit....
  - a. On the Virtual Hardware tab, select CD/DVD Drive 1.
  - b. For the Client Device, select Datastore ISO File.
  - c. Find and select your datastore in the Datastores category.
  - d. Find and select your ISO file in the Contents category.
  - e. Select OK to commit your selection and exit the Select File window.
  - f. Ensure that the Connect At Power On check box is selected.

Tip:

- Expand the CD/DVD drive 1 entry to view the details and the complete Connect At Power On label.
- Note that VMware related issues with ISO mounting at boot may occur if Connect At Power On

- g. Select OK to commit your selection and close the window.

28. Start the virtual machine by selecting the play button on the icon bar.

The installation might take several minutes to complete, depending on the availability of the system and the download speed.

29. Log in to the virtual machine by using an SSH tool to check the status of the installation:

- a. Enter the following command to connect to *mgmt* using SSH:

```
ssh ip_address -l apicadm
```

You are logging in with the default ID of *apicadm*, which is the API Connect ID that has administrator privileges.

- b. Select Yes to continue connecting.  
Your host names are automatically added to your list of hosts.
- c. Run the **apic status** command to verify that the installation completed and the system is running correctly.  
The command output for a correctly running Analytics system is similar to the following lines:

```
#sudo apic status
INFO[0000] Log level: info

Cluster members:
- testsrv0233.subnet1.example.com (1.2.152.233)
  Type: BOOTSTRAP_MASTER
  Install stage: DONE
  Upgrade stage: UPGRADE_DONE
  Docker status:
    Systemd unit: running
  Kubernetes status:
    Systemd unit: running
    Kubelet version: testsrv0233 (4.4.0-138-generic) [Kubelet v1.10.6, Proxy v1.10.6]
  EtcD status: pod etcd-testsrv0233 in namespace kube-system has status Running
  Addons: calico, dns, helm, kube-proxy, metrics-server, nginx-ingress,
EtcD cluster state:
- etcd member name: testsrv0233.subnet1.example.com, member id: 12836860275847862867, cluster id:
14018872452420182423,
  leader id: 12836860275847862867, revision: 365042, version: 3.1.17
```

Pods Summary:

NODE STATUS	NAMESPACE	NAME	READY
Pending	default	apic-analytics-analytics-client-76956644b9-cmgx8	0/0
Running	default	apic-analytics-analytics-client-76956644b9-v1qp2	1/1
Succeeded	default	apic-analytics-analytics-cronjobs-retention-1541381400-hp9fc	0/1
Succeeded	default	apic-analytics-analytics-cronjobs-rollover-1541445300-c5n6z	0/1
Pending	default	apic-analytics-analytics-ingestion-547f875467-8mhs1	0/0
Running	default	apic-analytics-analytics-ingestion-547f875467-s7f1j	1/1
Pending	default	apic-analytics-analytics-mtls-gw-85b8676855-jmh8c	0/0
Running	default	apic-analytics-analytics-mtls-gw-85b8676855-sw6ps	1/1

testsrv0233	default	apic-analytics-analytics-storage-basic-8cckh	1/1
Running			
testsrv0233	kube-system	calico-node-8crtp	2/2
Running			
testsrv0233	kube-system	coredns-87cb95869-6flvn	1/1
Running			
testsrv0233	kube-system	coredns-87cb95869-rccvb	1/1
Running			
testsrv0233	kube-system	etcd-testsrv0233	1/1
Running			
testsrv0233	kube-system	ingress-nginx-ingress-controller-f7b9z	1/1
Running			
testsrv0233	kube-system	ingress-nginx-ingress-default-backend-6f58fb5f56-nk1mv	1/1
Running			
testsrv0233	kube-system	kube-apiserver-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-apiserver-proxy-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-controller-manager-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-proxy-2vw9b	1/1
Running			
testsrv0233	kube-system	kube-scheduler-testsrv0233	1/1
Running			
testsrv0233	kube-system	metrics-server-5558db4678-9drz6	1/1
Running			
testsrv0233	kube-system	tiller-deploy-84f4c8bb78-vx65c	1/1
Running			

30. If you have now installed all subsystems, continue to [Access the Cloud Manager and begin API Connect Cloud Configuration](#).

## Results

An analytics installation starts the following Kubernetes pods:

- Full installation in **standard** mode:  
Table 1. Kubernetes pods in a full Analytics installation

Number	Pod
2	client
2	ingestion
2	mtls
1	operator
equal to the number of nodes	analytics-storage-basic

Note that for all pod types except analytics-storage-basic, there is only one of each pod when installed in **dev** mode. For analytics-storage-basic, the number of pods always matches the number of nodes in the subsystem, regardless of installation mode.

- Message queue deployment in **standard** mode: All of the pods shown in Table [Table 1](#), plus:

Table 2. Additional pods used for Message Queue deployments

Number	Pod
equal to the number of nodes	mq-kafka
equal to the number of nodes	mq-zookeeper

Note that the number of pods for mq-kafka and mq-zookeeper always match the number of nodes in the subsystem, regardless of installation mode.

- Ingestion-only installation in **standard** mode:

Table 3. Kubernetes pods in an ingestion-only Analytics installation

Number	Pods
2	ingestion
2	mtls

Note that in **dev** mode there is only one of each pod.

## What to do next

Identify the DataPower® appliances to be used as gateway servers in the API Connect cloud and obtain the IP addresses.

Define your API Connect configuration by using the API Connect cloud console. For more information, see [Defining the cloud](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying the Developer Portal in a VMware environment

You create a Developer Portal node by deploying the Developer Portal OVA template. After you deploy the Developer Portal OVA template, you can install the Developer Portal.

## Before you begin

Before you deploy:

- Review the [Deployment requirements on VMware](#).
- Review the [Configuration on VMware](#).
- For information on deploying a cluster, see [Configuring API Connect subsystems in a cluster on VMware](#)
- If you are upgrading from a previous version, see [Upgrading in a VMware environment](#).

Note:

Ensure that your kernel or Kubernetes node has the value of its `inotify` watches set high enough so that the Developer Portal can monitor and maintain the files for each Developer Portal site. If set too low, the Developer Portal containers might fail to start or go into a `non-ready` state when this limit is reached. If you have many Developer Portal sites, or if your sites contain a lot of content, for example, many custom modules and themes, then a larger number of `inotify` watches are required. You can start with a value of 65,000, but for large deployments, this value might need to go up as high as 1,000,000. The Developer Portal containers take `inotify` watches only when they need them. The full number is not reserved or held, so it is acceptable to set this value high.

## About this task

You must deploy the Developer Portal OVA template to create each Developer Portal node that you want in your cloud. Each node has a separate CLI password account that is required to log in through a Secure Shell (SSH) to complete specific administrative actions for only that node.

Important deployment information for Developer Portal:

- You must deploy the Developer Portal OVA template by using a version of the VMware vSphere Client that supports the SHA-512 Cryptographic Hash Algorithm.
- The Developer Portal node is initially configured with a default password of `7iron-hide`, with the user name of `admin`. For security reasons, change the default password by completing one of the following actions:
  - During deployment, if the feature is available in your VMware instance, enter a new password. If you specify a password during deployment, the password for the Developer Portal command line interface (CLI) is modified. Use the new password when you log into the CLI. You cannot modify the admin user name for the CLI.
  - After deployment, log in to the CLI for each virtual appliance and run the command to change the default password for that specific node. The CLI command is `passwd`.

Note that this console uses a US keyboard configuration, in which the `@` symbol can be in a different place to other keyboard configurations. If you are not using a US keyboard, ensure that you typed the password correctly.

- Only static IP addresses that are specified during the `apicup` project configuration before the installation of the OVAs are supported.
- To enable effective high availability for your Portal service, you need a latency that is less than 50ms between all OVAs to avoid the risk of performance degradation. Servers with uniform specifications are required, as any write actions occur at the speed of the slowest OVA, as the write actions are synchronous across the cluster of OVAs. It is recommended that there are three servers in each cluster of OVAs for the high availability configuration. The three servers can be situated in the same data center, or across three data centers to ensure the best availability. However, you can configure high availability with two data centers.
- The backup secret is a Kubernetes secret that contains your username and password for your backup database (sftp/s3). Only password-based authentication is supported for sftp and s3, not authentication based on public certificates and private keys. Password-based authentication for s3 requires that you generate an access key and secret. For example:
  - IBM (Cloud Object Storage): [Service credentials](#).
  - AWS: [Managing access keys](#).

## Procedure

Note: The following steps apply to VMware only. Depending on the VMware version that you are using, some of the steps might vary. For example, you might not be able to change the user name and password during deployment.

1. Ensure that you obtained the distribution file and have a project directory, as described in [First steps for deploying in a VMware environment](#).
2. Change to the project directory.

```
cd myProject
```

3. Create a portal subsystem.

```
apicup subsys create port portal
```

Where:

- `port` is the name of the Developer Portal server that you are creating. You can assign it any name, as long as the identifier consists of lower case alphanumeric characters or '-', with no spaces, starts with an alphabetic character, and ends with an alphanumeric character.
- `portal` indicates that you want it to create a Developer Portal microservice.

The `apiconnect-up.yml` file that is in that directory is updated to add the portal-related entries.

Tip: At any time, you can view the current developer portal subsystem values in the `apiconnect-up.yml` by running the `apicup get` command:

```
apicup subsys get port
```

If you have not yet configured the subsystem, the command might return errors. Also, if you have not updated the value, a default value is listed, if there is one that is available.

After configuration is complete, you can view output similar to the following sample:

```
Appliance settings
```

```
=====
```

Name	Value	Description
-----	-----	-----
additional-cloud-init-file		(Optional) Path to additional cloud-init yml file

```

data-device          sdb                               VM disk device (usually `sdb` for SCSI
or `vdb` for VirtIO)
default-password    $6$rounds=4096$imCJ9cfhFJ8X$pbmAl9ClWzcYzHZFoQ6
                    n7OnYcf/owQZiCpAtWazs/FUn/uE8uLD.9jwHE0AX4upfSqx/jf0ZmDbHPZ9bUlCY1 (Optional) Console login password for
`apicadm` user
dns-servers         [1.2.136.11]                               List of DNS servers
k8s-pod-network    172.16.0.0/16                               (Optional) CIDR for pods within the
appliance
k8s-service-network 172.17.0.0/16                               (Optional) CIDR for services within the
appliance
mode                standard
public-iface        eth0                                           Device for API/UI traffic (Eg: eth0)
search-domain       [subnet1.example.com]                       List for DNS search domains
ssh-keyfiles        [/home/vsphere/.ssh/id_rsa.pub]             List of SSH public keys files
traffic-iface       eth0                                           Device for cluster traffic (Eg: eth0)
license-version     Production

```

**Subsystem settings**  
=====

Name	Value	Description
site-backup-auth-pass		(optional) Server password for portal backups
site-backup-auth-user		(optional) Server username for portal backups
site-backup-host		(optional) FQDN for portal backups server
site-backup-path	/site-backups	(optional) Path for portal backups
site-backup-port	22	(optional) port for portal backups server
site-backup-protocol	sftp	(Optional) Protocol for portal backups (sftp/objstore)
site-backup-schedule	0 2 * * *	(optional) Cron schedule for portal backups

**Endpoints**  
=====

Name	Value	Description
portal-admin	api.portal.apimdev0232.subnet1.example.com	FQDN of Portal admin endpoint
portal-www	portal.apimdev0232.subnet1.example.com	FQDN of Portal web endpoint

4. For production environments, specify `mode=standard`.

```
apicup subsys set port mode=standard
```

The `mode=standard` parameter indicates that you are deploying in high availability (HA) mode for a production environment. If the mode parameter is omitted, the subsystem deploys by default in `dev` mode, for use in development and testing. For more information, see [Requirements for initial deployment on VMware](#).

5. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set port
license-version=<license_type>
```

The `license_type` must be either `Production` or `Nonproduction`. If not specified, the default value is `Nonproduction`.

6. Optional: Configure scheduled backups of the subsystem. This step is optional but is recommended. Refer to the instructions in [Backing up and restoring the Developer Portal in a Kubernetes environment](#).
7. Optional: Configure your logging.

Logging can be configured at a later time, but you must enable it before installation to capture the log events from the installation.

- Complete the procedure at [Configuring remote logging for a VMware deployment](#).
- Enter the following command to create the log file:

```
apicup subsys set port additional-cloud-init-file=config_file.yml
```

8. Enter the following commands to update the `apiconnect-up.yml` with the information for your environment:

- Use `apicup` to set your endpoints. You can use wildcard aliases or host aliases with your endpoints. Optionally, you can specify all endpoints with one `apicup` command. See [Tips and tricks for using APICUP](#).

Note: You cannot specify the underscore character "\_" in domain names that are used in endpoints. See [Configuration on VMware](#).

The endpoints must be unique hostnames which both point to the IP address of the OVA (single node deployment), or to the IP of a load balancer configured in front of the OVA nodes. See examples in the sample output in step 3.

Setting	Endpoint host description
portal-admin	This is the <i>unique_hostname</i> for communication between your Cloud Manager and API Manager, and your Developer Portal. The values for the portal-admin and portal-www must be different. <code>apicup subsys set port portal-admin=unique_hostname.domain</code>
portal-www	This is the <i>unique_hostname</i> for the Developer Portal Internet site that is created for the Developer Portal. Multiple portal-www endpoints may be configured, as described here: <a href="#">Defining multiple portal endpoints for a VMware environment</a> . <code>apicup subsys set port portal-www=unique_hostname.domain</code>

- Set your search domain. Multiple search domains should be separated by commas.

```
apicup subsys set port search-domain=your_search_domain
```

Where `your_search_domain` is the domain of your servers, entered in all lowercase. Setting this value ensures that your searches also append these values, which are based on your company's DNS resolution, at the end of the search value. A sample search domain is `mycompany.example.com`.

Ensure that the value for `your_search_domain` is resolved in the system's `/etc/resolv.conf` file to avoid "502" errors when accessing the Cloud Manager web site. For example:

```
# Generated by resolvconf
search your_search_domain ibm.com other.domain.com
```

c. Set your domain name servers (DNS).

Supply the IP addresses of the DNS servers for your network. Use a comma to separate multiple server addresses.

```
apicup subsys set port dns-servers=ip_address_of_dns_server[,ip_address_of_another_dns_server_if_necessary]
```

DNS entries may not be changed on a cluster after the initial installation.

9. Set a Public key.

```
apicup subsys set port ssh-keyfiles=path_to_public_ssh_keyfile
```

Setting this key enables you to use `ssh` with this key to log in to the virtual machine to check the status of the installation. You will perform this check in step 29 of these instructions.

10. You can set a hashed password that you enter to log in to your Developer Portal server for the first time.

a. **Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).

b. If you do not have a password hashing utility, install one.

Operating system	Command
Ubuntu, Debian, OSX	If the <code>mkpasswd</code> command utility is not available, download and install it. (You can also use a different password hashing utility.) On OSX, use the command: <code>gem install mkpasswd</code> .
Windows, Red Hat	If necessary, a password hashing utility like OpenSSL.

c. Create a hashed password

Operating system	Command
Ubuntu, Debian, OSX	<code>mkpasswd --method=sha-512 --rounds=4096 password</code>
Windows, Red Hat	For example, using OpenSSL: <code>openssl passwd -1 password</code> . Note that you might need to add your password hashing utility to your path; for example, in Windows: <code>set PATH=c:\cygwin64\bin;%PATH%</code>

d. Set the hashed password for your subsystem:

```
apicup subsys set port default-password="hashed_password"
```

Notes:

- The password is hashed. If it is in plain text, you cannot log into the VMWare console.
- Note that the password can only be used to login through the VMWare console. You cannot use it to SSH into the Appliance as an alternative to using the `ssh-keyfiles`.
- On Linux or OSX, use single quotes around `hashed_password`. For Windows, use double quotes.
- If you are using a non-English keyboard, understand the limitations with using the remote VMWare console. See [Requirements for initial deployment on VMWare](#).

11. Optional: If the default IP ranges for the API Connect Kubernetes pod and the service networks conflict with IP addresses that must be used by other processes in your deployment, modify the API Connect values.

You can change the IP ranges of the Kubernetes pod and the service networks from the default values of 172.16.0.0/16 and 172.17.0.0/16, respectively. In the case that a /16 subnet overlaps with existing IPs on the network, a Classless Inter-Domain Routing (CIDR) as small as /22 is acceptable. You can modify these ranges during initial installation and configuration only. You cannot modify them once an appliance has been deployed. See [Configuration on VMWare](#).

a. Update the IP range for the Kubernetes pod

```
apicup subsys set port k8s-pod-network='new_pod_range'
```

Where `new_pod_range` is the new value for the range.

b. Update the IP range for Service networks.

```
apicup subsys set port k8s-service-network='new_service_range'
```

Where `new_service_range` is the new value for the range.

12. Add your hosts.

```
apicup hosts create port hostname.domainname hd_password
```

Where the following are true:

- `hostname.domainname` is the fully qualified name of the server where you are hosting your Developer Portal, including the domain information.
- `hd_password` is the password of the Linux Unified Key Setup used to encrypt the storage for your Developer Portal. This password is hashed when it is stored on the server or in the ISO. Note that the password is base64 encoded when stored in `apiconnect-up.yml`. Repeat this command for each host that you want to add.

Note:

- Host names and DNS entries may not be changed on a cluster after the initial installation.
- **Version 2018.4.1.0:** Ensure that Reverse DNS lookup configuration is configured for the host names.

```
nslookup <ip_address>
```

For **Version 2018.4.1.1 or later**, Reverse DNS lookup is not required.

13. Create your interfaces.

```
apicup iface create port hostname.domainname physical_network_id host_ip_address/subnet_mask gateway_ip_address
```

Where `public_iface_id` is the network interface ID of your physical server. The value is most often `eth0`. The value can also be `ethx`, where `x` is a number identifier. The format is similar to this example: `apicup iface create port myHostname.domain eth0 192.0.2.1/255.255.1.1 192.0.2.1`

- Optional: Use `apicup` to view the configured hosts:

```
apicup hosts list port
apimdev0232.hursley.ibm.com
Device IP/Mask Gateway
eth0 1.2.152.232/255.255.254.0 1.2.152.1
```

- Optional: Verify that the configuration settings are valid.

```
apicup subsys get port --validate
```

The output lists each setting and adds a check mark after the value once the value is validated. If the setting lacks a check mark and indicates an invalid value, reconfigure the setting. See the following sample output.

```
apicup subsys get port --validate
Appliance settings
=====
Name Value
----
additional-cloud-init-file ✓
data-device sdb ✓
default-password $6$rounds=4096$iMCJ9cfhFJ8X$pbmAl9ClWzcYz
HZFoQ6n7OnYCF/owQZiiCpAtWazs/FUn/uE8uLD.9jwHE0AX4upFSqx/jf0ZmDbHPZ9bU1CY1 ✓
dns-servers [1.2.136.11] ✓
k8s-pod-network 172.16.0.0/16 ✓
k8s-service-network 172.17.0.0/16 ✓
mode standard ✓
public-iface eth0 ✓
search-domain [subnet1.example.com] ✓
ssh-keyfiles [/home/vsphere/.ssh/id_rsa.pub] ✓
traffic-iface eth0 ✓
license-version Production ✓

Subsystem settings
=====
Name Value
----
site-backup-auth-pass ✓
site-backup-auth-user ✓
site-backup-host ✓
site-backup-path /site-backups ✓
site-backup-port 22 ✓
site-backup-protocol sftp ✓
site-backup-schedule 0 2 * * * ✓

Endpoints
=====
Name Value
----
portal-admin api.portal.testsrv0232.subnet1.example.com ✓
portal-www portal.testsrv0232.subnet1.example.com ✓
```

- Create your ISO file.

```
apicup subsys install port --out portplan-out
```

The `--out` parameter and value are required. In this example, the ISO file is created in the `myProject/portplan-out` directory.

If the system cannot find the path to your software that creates ISO files, create a path setting to that software by running a command similar to the following command:

Operating system	Command
OSX and Linux	<code>export PATH=\$PATH:/Users/your_path/</code>
Windows	<code>set PATH="c:\Program Files (x86)\cdrtools";%PATH%</code>

- Log into the VMware vSphere Web Client.
- Using the VSphere Navigator, navigate to the directory where you are deploying the OVA file.
- Right-click the directory and select Deploy OVF Template.
- Complete the Deploy OVF Template wizard.
  - Select the `apiconnect-portal.ova` template by navigating to the location where you downloaded the file from [Passport Advantage®](#).
  - Enter a name and location for your file.
  - Select a resource for your template.
  - Review the details for your template.
  - Select the size of your configuration.
  - Select the storage settings.

Note that the number of Central Processing Units, RAM, and size of disk that you need for the Developer Portal varies depending on the number of sites that are hosted and the number of concurrent users you expect your site to have:

Table 1. Developer Portal hardware requirements

Number of sites	Number of concurrent users	Number of CPUs	Amount of RAM (GB)	Data Disk Size (GB)++
-----------------	----------------------------	----------------	--------------------	-----------------------

Number of sites	Number of concurrent users	Number of CPUs	Amount of RAM (GB)	Data Disk Size (GB)++
1	1	2**	4	50
20	5	4	16	70
100	20	8	32	100
100	100	16	64	500

Important:

- ++The data disk size is extra to the main disk of the OVA. The main disk of the OVA is sized at 100GB, and should not be changed. Therefore, the total disk size of the OVA is 100GB plus the data disk size. The default data disk size is 100GB. If you want to select a different value, then you need to do this by using the OVF Tool, or the VMware GUI, before you power on the VM. Note that certain versions of the VMware GUI may not allow you to resize the data disk, and in this case you should use the OVF Tool.
- \*\*The requirement of 2 CPUs is suitable only for proof-of-concept work, and non-high availability deployments. For example, this configuration is suitable for the demo mode of the apicup installation, which is set by using the command `apicup subsys set port mode=dev`. Note that `standard` mode is set by default.
- It's not recommended to have more than 100 sites per Developer Portal service. Note that it's not necessary to have a Portal site for every Catalog, for example Catalogs that are only for API Developers don't need a Portal site, as the APIs can be tested by using credentials from the API Manager. If more than 100 sites are required, you should configure additional Developer Portal services; see [Registering a Portal service](#).

- Select the networks.
- Customize the Template, if necessary.
  - Review the details to ensure that they are correct.
  - Select Finish to deploy the virtual machine.

Note: Do not change the OVA hardware version, even if the VMware UI shows a Compatibility range that includes other versions. See [Requirements for initial deployment on VMware](#).

The template creation appears in your Recent Tasks list.

- Select the Storage tab in the Navigator.
- Navigate to your datastore.
- Upload your ISO file.
  - Select the Navigate to the datastore file browser icon in the icon menu.
  - Select the Upload a file to the Datastore icon in the icon menu.
  - Navigate to the ISO file that you created in your project. It is the `myProject/portplan-out`
  - Upload the ISO file to the datastore.
- Leave the datastore by selecting the VMs and Templates icon in the Navigator.
- Locate and select your virtual machine.
- Select the Configure tab in the main window.
- Select Edit....
  - On the Virtual Hardware tab, select CD/DVD Drive 1.
  - For the Client Device, select Datastore ISO File.
  - Find and select your datastore in the Datastores category.
  - Find and select your ISO file in the Contents category.
  - Select OK to commit your selection and exit the Select File window.
  - Ensure that the Connect At Power On check box is selected.

Tip:

- Expand the CD/DVD drive 1 entry to view the details and the complete Connect At Power On label.
- Note that VMware related issues with ISO mounting at boot may occur if Connect At Power On

- Select OK to commit your selection and close the window.

- Start the virtual machine by selecting the play button on the icon bar.

The installation might take several minutes to complete, depending on the availability of the system and the download speed.

- Log in to the virtual machine by using an SSH tool to check the status of the installation:

- Enter the following command to connect to `mgmt` using SSH:

```
ssh ip_address -l apicadm
```

You are logging in with the default ID of `apicadm`, which is the API Connect ID that has administrator privileges.

- Select Yes to continue connecting. Your host names are automatically added to your list of hosts.
- Run the `apic status` command to verify that the installation completed and the system is running correctly. The command output for a correctly running Developer Portal system is similar to the following lines:

```
$ sudo apic status
INFO[0000] Log level: info
```

```
Cluster members:
```

```
- testsrv1251.subnet1.example.com (1.2.3.4)
```

```
  Type: BOOTSTRAP MASTER
```

```
  Install stage: DONE
```

```
  Upgrade stage: NONE
```

```
  Docker status:
```

```
    Systemd unit: running
```

```
  Kubernetes status:
```

```
    Systemd unit: running
```

```
    Kubelet version: testsrv1251 (4.4.0-137-generic) [Kubelet v1.10.6, Proxy v1.10.6]
```

```
    Etdc status: pod etcd-testsrv1251 in namespace kube-system has status Running
```

```
    Addons: calico, dns, helm, kube-proxy, metrics-server, nginx-ingress,
```

```
Etdc cluster state:
```

```
- etcd member name: testsrv1251.subnet1.example.com, member id: 10293853252850049269, cluster id: 17044377177359475136, leader id: 10293853252850049269, revision: 1485879, version: 3.1.17
```

```
Pods Summary:
```

NODE	NAMESPACE	NAME	READY
testsrv1251	default	re702738954-apic-portal-db-f89vn	2/2
Running	default	re702738954-apic-portal-nginx-6ffb8676d9-gtfc6	0/0
Pending			



Pending	default	re702738954-apic-portal-nginx-6ffb8676d9-nqdzf	0/0
testsrv1251	default	re702738954-apic-portal-nginx-6ffb8676d9-q85mt	1/1
Running	default	re702738954-apic-portal-www-p9bvix	2/2
Running	kube-system	calico-node-xkpbk	2/2
Running	kube-system	coredns-87cb95869-p4qhf	1/1
Running	kube-system	coredns-87cb95869-z2n5z	1/1
Running	kube-system	etcd-testsrv1251	1/1
Running	kube-system	ingress-nginx-ingress-controller-dsnxw	1/1
Running	kube-system	ingress-nginx-ingress-default-backend-6f58fb5f56-ldx7t	1/1
Running	kube-system	kube-apiserver-testsrv1251	1/1
Running	kube-system	kube-apiserver-proxy-testsrv1251	1/1
Running	kube-system	kube-controller-manager-testsrv1251	1/1
Running	kube-system	kube-proxy-4pp8v	1/1
Running	kube-system	kube-scheduler-testsrv1251	1/1
Running	kube-system	metrics-server-6f58fb84cdd-hkztz	1/1
Running	kube-system	tiller-deploy-84f4c8bb78-v6k95	1/1

30. If you have now installed all subsystems, continue to [Access the Cloud Manager and begin API Connect Cloud Configuration](#).

## What to do next

If you want to deploy an API Connect Analytics OVA file, continue with [Deploying the Analytics subsystem in a VMware environment](#).

If you did not specify a new password during deployment in VMware, then after deployment, log in to the command-line interface (CLI) for each appliance and run the command `passwd` to change the password.

- [Defining multiple portal endpoints for a VMware environment](#)  
Multiple public facing endpoints (portal-www) can be defined for the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Defining multiple portal endpoints for a VMware environment

Multiple public facing endpoints (portal-www) can be defined for the Developer Portal.

### About this task

You can override the single endpoint definition for portal-www (and the associated portal-www-ingress TLS certificate), in order to support multiple portal-www endpoints.

For information about the endpoints for the Portal, see [Deploying the Developer Portal in a VMware environment](#).

Following are the example endpoints for configuring different sites served by the same Portal service, as configured in this task:

- `https://banking.example.com/loans`
- `https://insurance.example.com/vehicle`

These unique endpoints allow portal sites to be defined on the Portal service with different host names and domains. They replace endpoints that distinguish different sites by sub paths, as shown in the following examples:

- `https://www.example.com/banking/loans`
- `https://www.example.com/insurance/vehicle`

## Procedure

1. Create TLS secrets for each portal-www endpoint by generating certificates  
Following is an example for how to generate certificates for each portal-www endpoint using openssl:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout banking-tls.key -out banking-tls.crt -subj
"/CN=banking.example.com"
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout insurance-tls.key -out insurance-tls.crt -subj
"/CN=insurance.example.com"
```

2. Store the SSL certificates in a secret.

Copy the certificates to the Portal virtual machine as follows:

```
scp banking-tls.key banking-tls.crt insurance-tls.key insurance-tls.crt apicadm@<portal-vm-address>
```

Access the virtual machine using SSH and store the SSL certificates in a secret:

```
sudo kubectl --kubeconfig /etc/kubernetes/admin.conf create secret tls banking-tls --key banking-tls.key --cert banking-tls.crt
sudo kubectl --kubeconfig /etc/kubernetes/admin.conf create secret tls insurance-tls --key insurance-tls.key --cert insurance-tls.crt
```

3. Specify the portal-www endpoints in an extra values file.

Create an extra values file or append to your current one. Enter the name and secret for each endpoint as an ingress setting in the extra values file. (One extra values file is allowed.) For instructions on creating an extra-values-file, see [Creating an extra values file in a Kubernetes environment](#).

```
apic-portal-www:
  ingress:
    web:
      hosts:
        - name: banking.example.com
          secret: banking-tls
        - name: insurance.example.com
          secret: insurance-tls
```

4. Configure your Portal subsystem to load the extra values file with the following command:

```
apicup subsys set <portal-subsys> extra-values-file=<full-path-to-extra-values-file>
```

5. Install the portal subsystem with the new extra values file using `apicup subsys install portal-subsys`.

For more information on installing the Portal subsystem, see [Deploying the Developer Portal in a VMware environment](#).

6. If your deployment had existing Portal sites when you configured multiple endpoints, ensure that the Portal site URLs specified in the Manager UI Catalog settings page are consistent with the new endpoint URLs. Access the Catalog setting page, and review the URLs of those existing sites. Modify as appropriate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Access the Cloud Manager and begin API Connect Cloud Configuration

When all subsystems are deployed, use the admin account to access the Cloud Manager administrative console and begin configuration.

### About this task

When you have deployed all required API Connect components, and all subsystems are running, complete the following:

### Procedure

1. If you have not previously accessed the Cloud Manager, verify you can access the API Connect Cloud Manager. Enter the URL in your browser.

The syntax is `https://<hostname.domain>/admin`. For example:

```
https://cloud-admin-ui.testsrv0231.subnet1.example.com/admin
```

The first time that you access the Cloud Manager user interface, you enter `admin` for the user name and `7iron-hide` for the password. You will be prompted to change the Cloud Administrator password and email address. See [Accessing the Cloud Manager user interface](#).

2. Define your API Connect configuration by using the API Connect Cloud Manager. See [Cloud Manager configuration checklist](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring API Connect subsystems in a cluster on VMware

This topic describes how to configure a cluster of API Connect subsystems (management server, analytics, and Developer Portal) with three VMs for each subsystem, for use with a load balancer, to support a high availability (HA) environment.

### Before you begin

To create a cluster of hosts for each API Connect subsystem, use `apicup` to create a subsystem with all the required parameters, and to add as many hosts as needed to the configuration file. The configuration file is a `.yaml` file in the project directory.

For each host of a subsystem added in the `.yaml` file, a separate ISO file is created for the cluster member VM. Note that the ISO files for each VM must stay attached during the whole lifetime of the VM.

In an API Connect cluster in a VMware environment, the first three nodes are master nodes. Additional nodes are worker nodes.

Note: To add a new host to an existing cluster, you can create the host, regenerate the ISO and attach it to the new virtual machine, and it will automatically join the cluster.

To use these instructions, you should first review:

- [Requirements for initial deployment on VMware](#)
- [Load balancer configuration in a VMware deployment](#)
- [Deployment overview for endpoints and certificates](#)
- [Firewall requirements on VMware](#)
- [Firewall enabled ports for clustered OVA deployments](#)
- [API Connect 2018.4.1.x Whitepaper](#)

Important: For best performance it is recommended that the network latency between any 2 nodes be as low as possible. Do not configure nodes from the same subsystem cluster across multiple data centers with a high latency network. A high latency network is one that experiences more than 30ms latency between nodes. For more information, see the [API Connect V2018 Whitepaper](#)

## About this task

This page presents a concise step-by-step flow, with sample commands, for the configuration of each subsystem. As you step through the flow, you might want to refer back to the detailed configuration steps for each subsystem. The detailed instructions provide additional considerations for each step, and include optional configuration tasks for backups, logging, message queues (analytics), and password hashing. Each of the subsystem pages describe how to use the VMware console to deploy the ISOs that you create here.

Detailed configuration steps:

- [Deploying the Management subsystem in a VMware environment](#)
- [Deploying the Analytics subsystem in a VMware environment](#)
- [Deploying the Developer Portal in a VMware environment](#)

The example commands uses the following values for DNS server, internet gateway, host names and IP addresses:

Component	Host names and IP addresses
DNS Name Server	IP: 192.168.1.1
Internet gateway	IP: 192.168.1.2
Manager on VM1	Hostname: manager1.sample.example.com IP: 192.168.1.101
Manager on VM2	Hostname: manager2.sample.example.com IP: 192.168.1.102
Manager on VM3	Hostname: manager3.sample.example.com IP: 192.168.1.103
Analytics on VM4	Hostname: analytics1.sample.example.com IP: 192.168.1.104
Analytics on VM5	Hostname: analytics2.sample.example.com IP: 192.168.1.105
Analytics on VM6	Hostname: analytics3.sample.example.com IP: 192.168.1.106
Developer Portal on VM7	Hostname: portal1.sample.example.com IP: 192.168.1.107
Developer Portal on VM8	Hostname: portal2.sample.example.com IP: 192.168.1.108
Developer Portal on VM9	Hostname: portal3.sample.example.com IP: 192.168.1.109

## Procedure

1. Create the Management subsystems

a. Create the Management subsystem.

```
apicup subsys create mgmt management
```

b. Set install mode to **standard**.

```
apicup subsys set mgmt mode=standard
```

If you omit this step, the install mode defaults to **dev**, which is for non-HA environments, and will not support three instances of the subsystem in one cluster.

c. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set mgmt  
license-version=<license_type>
```

The *license\_type* must be either **Production** or **Nonproduction**. If not specified, the default value is **Nonproduction**.

d. Set management endpoints

Endpoints can point to VM host names, but in cluster deployments typically point to a load balancer. The load balancer distributes requests over the 3 VMs. The following values point to a sample load balancer URL.

Component	Command
Management REST API URL	<code>apicup subsys set mgmt platform-api platform-api.sample.example.com</code>
Consumer (Portal) REST API URL	<code>apicup subsys set mgmt consumer-api consumer-api.sample.example.com</code>
Cloud Manager UI	<code>apicup subsys set mgmt cloud-admin-ui cloud-admin-ui.sample.example.com</code>
API Manager UI	<code>apicup subsys set mgmt api-manager-ui api-manager-ui.sample.example.com</code>

e. Set the search domain for the VM.

```
apicup subsys set mgmt search-domain sample.example.com
```

- f. Set the DNS Name Server for the VM to look up endpoints.

```
apicup subsys set mgmt dns-servers 192.168.1.1
```

- g. Set a Public Keyfile.

This is the public key of the user account that you want to use to **ssh** from, to the appliance.

```
apicup subsys set mgmt ssh-keyfiles "id_rsa.pub"
```

- h. Create the hosts for the subsystem.

You must specify a password to use to encrypt the disks that the appliance uses. Replace the example password in the previous example with a strong password that meets your security requirements.

```
apicup hosts create mgmt manager1.sample.example.com password123
apicup hosts create mgmt manager2.sample.example.com password123
apicup hosts create mgmt manager3.sample.example.com password123
```

- i. Set the network interface. Note that the last parameter is the Internet Gateway.

```
apicup iface create mgmt manager1.sample.example.com eth0 192.168.1.101/255.255.255.0 192.168.1.2
apicup iface create mgmt manager2.sample.example.com eth0 192.168.1.102/255.255.255.0 192.168.1.2
apicup iface create mgmt manager3.sample.example.com eth0 192.168.1.103/255.255.255.0 192.168.1.2
```

- j. Set the network traffic interfaces.

```
apicup subsys set mgmt traffic-iface eth0
apicup subsys set mgmt public-iface eth0
```

- k. Verify the host configuration.

```
apicup hosts list mgmt
```

Note: This command might return the following messages, which you can ignore:

```
* host is missing traffic interface
* host is missing public interface
```

- l. Set a hashed password to access the appliance VM through the VMware Remote Console. Use an operating system utility to create a hashed password, and then use **apicup** to set the hashed password for your subsystem:

```
apicup subsys set mgmt default-password='$1$aTD7uXAO$kNoMAefjGKBwMFiu.8ctr0'
```

**Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).

- m. Validate the installation.

```
apicup subsys get mgmt --validate
```

- n. Create an ISO file in a plan folder. For example, **mgmtplan-out**.

```
apicup subsys install mgmt --out mgmtplan-out
```

If you have multiple nodes listed for the hosts, when you run the **--out** command, it creates an ISO for each node. When deploying the nodes from VMware, each node gets its own ISO file attached.

- o. To deploy the ISOs on VMware, see step [17](#) in [Deploying the Management subsystem in a VMware environment](#).

## 2. Create the analytics subsystems

- a. Create the subsystem:

```
apicup subsys create analyt analytics
```

- b. Specify **mode=standard**.

```
apicup subsys set analyt mode=standard
```

- c. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set analyt
license-version=<license_type>
```

The *license\_type* must be either **Production** or **Nonproduction**. If not specified, the default value is **Nonproduction**.

- d. Set analytics endpoints

Endpoints can point to VM host names, but in cluster deployments typically point to a load balancer. The load balancer distributes requests over the 3 VMs. The following values point to a sample load balancer URL.

Component	Command
analytics-ingestion	<code>apicup subsys set analyt analytics-ingestion=analytics-ingestion.sample.example.com</code>
analytics-client	<code>apicup subsys set analyt analytics-client=analytics-client.sample.example.com</code>

- e. Set the search domain for the VM.

```
apicup subsys set analyt search-domain sample.example.com
```

- f. Set the DNS Name server for the VM to look up endpoints.

```
apicup subsys set analyt dns-servers 192.168.1.1
```

- g. Set a Public Keyfile.

This is the public key of the user account that you want to use to **ssh** from, to the appliance

```
apicup subsys set analyt ssh-keyfiles "id_rsa.pub"
```

- h. Set a hashed password to access the appliance VM through the VMware Remote Console. Use an operating system utility to create a hashed password, and then use `apicup` to set the hashed password for your subsystem:

```
apicup subsys set analyt default-password='$1$aTD7uXAO$kNoMAefjGKBwMFiU.8ctr0'
```

**Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).

- i. Create the hosts for the subsystem. You must specify a password to use to encrypt the disks that the appliance uses. Replace the example password in the following password with a strong password that meets your security requirements.

```
apicup hosts create analyt analytics1.sample.example.com password123
apicup hosts create analyt analytics2.sample.example.com password123
apicup hosts create analyt analytics3.sample.example.com password123
```

- j. Set the network interface.

Note that the last parameter is the Internet Gateway.

```
apicup iface create analyt analytics1.sample.example.com eth0 192.168.1.104/255.255.255.0 192.168.1.2
apicup iface create analyt analytics2.sample.example.com eth0 192.168.1.105/255.255.255.0 192.168.1.2
apicup iface create analyt analytics3.sample.example.com eth0 192.168.1.106/255.255.255.0 192.168.1.2
```

- k. Check the host configuration for problems.

```
apicup hosts list analyt
```

- l. Validate the installation.

```
apicup subsys get analyt --validate
```

- m. Create an ISO file in a plan folder. For example, `analytplan-out`.

```
apicup subsys install analyt --out analytplan-out
```

If you have multiple nodes listed for the hosts, when you run the `--out` command, it creates an ISO for each node. When deploying the nodes from VMware, each node gets its own ISO file attached.

- n. To deploy the ISOs, see step 17 in [Deploying the Analytics subsystem in a VMware environment](#)

### 3. Create the portal subsystems

- a. Create the portal.

```
apicup subsys create port portal
```

- b. For production environments, specify `mode=standard`.

```
apicup subsys set port mode=standard
```

If you omit this step, the install mode defaults to `dev`, which is for development and testing in non-HA environments only, and will not support three instances of the subsystem in one cluster.

- c. **Version 2018.4.1.10 or later:** Specify the license version you purchased.

```
apicup subsys set port
license-version=<license_type>
```

The `license_type` must be either `Production` or `Nonproduction`. If not specified, the default value is `Nonproduction`.

- d. Set the portal endpoints.

Endpoints can point to VM host names, but in cluster deployments typically point to a load balancer. The load balancer distributes requests over the 3 VMs. The following values point to a sample load balancer URL.

Component	Command
portal-admin	<code>apicup subsys set port portal-admin=portal-admin.sample.example.com</code>
portal-www	<code>apicup subsys set port portal-www=portal-www.sample.example.com</code>

- e. Set the search domain for the VM.

```
apicup subsys set port search-domain sample.example.com
```

- f. Set the DNS Name server for the VM to look up endpoints.

```
apicup subsys set port dns-servers 192.168.1.1
```

- g. Set a Public Keyfile.

This is the public key of the user account that you want to use to `ssh` from, to the appliance

```
apicup subsys set port ssh-keyfiles "id_rsa.pub"
```

- h. Set a hashed password to access the appliance VM through the VMware Remote Console. Use an operating system utility to create a hashed password, and then use `apicup` to set the hashed password for your subsystem:

```
apicup subsys set port default-password='$1$aTD7uXAO$kNoMAefjGKBwMFiU.8ctr0'
```

**Important:** Review the requirements for creating and using a hashed password. See [Setting and using a hashed default password](#).

- i. Create the hosts for the subsystem. You must specify a password to use to encrypt the disks that the appliance uses.

```
apicup hosts create port portal1.sample.example.com password123
apicup hosts create port portal2.sample.example.com password123
apicup hosts create port portal3.sample.example.com password123
```

Replace the example password in the previous example with a strong password that meets your security requirements.

- j. Set the network interface.

Note that the last parameter is the Internet Gateway.

```
apicup iface create port portal1.sample.example.com eth0 192.168.1.107/255.255.255.0 192.168.1.2
apicup iface create port portal2.sample.example.com eth0 192.168.1.108/255.255.255.0 192.168.1.2
apicup iface create port portal3.sample.example.com eth0 192.168.1.109/255.255.255.0 192.168.1.2
```

k. Check the host configuration for problems.

```
apicup hosts list port
```

l. Validate the installation.

```
apicup subsys get port --validate
```

m. Create an ISO file in a plan folder. For example, `portplan-out`.

```
apicup subsys install port --out portplan-out
```

If you have multiple nodes listed for the hosts, when you run the `--out` command, it creates an ISO for each node. When deploying the nodes from VMware, each node gets its own ISO file attached.

n. To deploy the ISOs on VMware, see step [17](#) in [Deploying the Developer Portal in a VMware environment](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing the IBM License Metric Tool for VMware

The IBM® License Metric Tool (ILMT) helps you assess if you are compliant with licensing requirements.

### About this task

ILMT provides useful features for managing virtualized environments and measuring license utilization. ILMT discovers the software that is installed in your infrastructure, helps you to analyze the consumption data, and allows you to generate audit reports. Each report provides you with different information about your infrastructure, for example the computer groups, software installations, and the content of your software catalog.

By default, every ILMT audit report presents data from the previous 90 days. You can customize the type and amount of information displayed in a report by using filters, and save your personal settings for future use. You can also export the reports to .csv or .pdf format, and schedule report emails so that specified recipients are notified when important events occur.

For more information, see the [IBM License Metric Tool](#) documentation.

Note: These instructions cover ILMT installation for IBM API Connect deployments in a VMware environment, and apply to the Management, Developer Portal, and Analytics OVAs.

### Procedure

To download the BigFix Agent for the Ubuntu operating system, complete the following steps:

1. For example, starting from here: <http://support.bigfix.com/bes/release/>, follow the link to the latest release, for example: <http://support.bigfix.com/bes/release/9.5/patch11/>
2. In the Agent section of the table, click the Download link corresponding to the following release:
  - Operating System: Ubuntu
  - Version: 16
  - Architecture: x86\_64In this example, the direct download link would be: <http://software.bigfix.com/download/bes/95/BESAgent-9.5.11.191-ubuntu10.amd64.deb>

Upload the BigFix Agent package to each OVA server, as follows:

3. Upload the BigFix Agent .deb package to each running OVA server. Following is an example of the command for uploading the file that was downloaded in Step 2:

```
scp BESAgent-9.5.11.191-ubuntu10.amd64.deb apicadm@<OVA_Hostname>:~
```

The generic command is:

```
scp <BigFix_Agent_Package_File_Path> apicadm@<OVA_Hostname>:~
```

The BigFix Agent .deb package will be uploaded to `/home/apicadm`.

Install the BigFix client on each OVA server, as follows:

4. Log in, with a secure shell connection, to the server as the `apicadm` user.
5. Become root by entering `sudo -s`.
6. Follow the BigFix Agent installation instructions located here: [IBM Bigfix installation instructions Ubuntu](#) to install the .deb package.

### What to do next

Ensure that TCP/IP port 52311 is open on your firewall.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding a static route on a virtual machine

You can add a static route to the routing table when deploying API Connect on a VMware virtual machine.

### About this task

---

Add commands to your additional `cloud-init` file. You can do this for the Management, Analytics, and Portal subsystems.

### Procedure

---

1. Specify an additional cloud-init file:

```
apicup subsys set <subsys> additional-cloud-init-file <path-to-cloud-init-file>
```

2. Add lines to your cloud-init file.

Syntax:

```
bootcmd:  
- ip route add <destination>/<mask> via <gateway> dev eth<n>
```

Example:

```
bootcmd:  
- ip route add 172.27.218.0/24 via 172.27.218.1 dev eth1  
- ip route add 172.26.203.0/24 via 172.27.218.1 dev eth1
```

3. Regenerate the ISO file.

```
apicup subsys install <subsys> --out <plan-directory>
```

4. Restart the ISO, and ensure that the node(s) have this updated ISO attached at startup.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring use of an external NTP server

You must configure an external NTP server for use by API Connect when deploying on a VMware virtual machine.

### About this task

---

Secure communication between API Connect subsystems relies on the system time being in sync on all hosts. For example, time stamps are checked to ensure that certificates are valid. When API Connect is deployed behind a firewall that blocks access to the internet, the API Connect subsystems cannot by default access a Network Time Protocol (NTP) server.

You can use an additional cloud-init file to manually specify an NTP server for use by the subsystems. Complete the following steps.

### Procedure

---

1. Create the cloud-init file extra values file, and enter the configuration details that you want to overwrite. For example:

```
ntp:  
  enabled: true  
  ntp_client: systemd-timesyncd  
  servers:  
    - time.google.com
```

2. Use `apicup` to specify the cloud-init file.

Syntax:

```
apicup subsys set <subsys> additional-cloud-init-file <path-to-cloud-init-file>
```

Example:

```
apicup subsys set mgmt additional-cloud-init-file myCloudInitFile.yaml
```

3. Install the subsystem. Note that the output directory must be empty:

```
apicup subsys install mgmt --out mgmtplan-out
```

4. Deploy the VMware image (.ova) with the ISO file that is generated.

To review the deployment steps, see [Deploying the Management subsystem in a VMware environment](#).

5. Verify that the correct NTP server is being used:

```
journalctl -u systemd-timesyncd
```

Example output for the NTP server that was set in Step 1:

```
Nov 05 21:09:24 h-apicdev-4 systemd[1]: Starting Network Time Synchronization...
Nov 05 21:09:24 h-apicdev-4 systemd[1]: Started Network Time Synchronization.
Nov 05 21:09:24 h-apicdev-4 systemd-timesyncd[1697]: Synchronized to time server 216.239.35.8:123 (time.google.com).
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deploying DataPower Gateway

API Connect uses IBM® DataPower® Gateway to provide the gateway service.

The instructions in this section describe how to deploy IBM DataPower Gateway from files obtained from [Passport Advantage®](#), and apply to the following API Connect scenarios:

- API Connect deployments for the VMware environment
- API Connect deployments for Kubernetes, with a DataPower Gateway in a *non-Kubernetes* environment.

For these scenarios, continue with [Installing DataPower Gateway](#).

Note: Do not use these instructions if you are deploying API Connect deployments for Kubernetes, with DataPower Gateway in a *Kubernetes* environment. In this case, you can use DataPower Gateway files that are packaged with API Connect, on the API Connect section of Fix Central, and configured by using the API Connect Install Assist (`apicup`) commands. Follow the instructions in [First steps for installing API Connect: Upload files to registry](#).

- [Installing DataPower Gateway](#)  
Install and configure IBM DataPower Gateway in a non-Kubernetes environment for use with API Connect.
- [Configuring DataPower Gateway for API Connect](#)  
You can configure the IBM DataPower Gateway to prepare for a registration with the API Connect Management server.
- [Sample configuration for multiple peering objects on gateway services external to Kubernetes](#)  
Sample for reconfiguring the API Connect domain configuration for your gateway services to include multiple peering objects.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing DataPower Gateway

Install and configure IBM® DataPower® Gateway in a non-Kubernetes environment for use with API Connect.

### Before you begin

Do not use these instructions for either of the following deployment scenarios:

- Both API Connect and DataPower Gateway in a Kubernetes environment.  
In this scenario, use the instructions in [First steps for installing API Connect: Upload files to registry](#).
- An existing DataPower Gateway 2018.4.1.0 in a non-Kubernetes environment, and you want to install a newer Fix Pack, such as 2018.4.1.3.  
In this scenario, complete an *upgrade* rather than a new installation. See [Upgrading DataPower Gateway Service](#).

To review the installation and configuration scenarios, see [Deploying DataPower Gateway](#).

### About this task

These instructions apply for Version 2018.4.1.0 or later.

### Procedure

1. Ensure that DataPower Gateway firmware version you plan to install matches the API Connect Management server version. For example, DataPower 2018.4.1.6 and API Connect 2018.4.1.6. It is recommended that you use the latest version of each product, as viewable in the SPCR reports. See [IBM API Connect Version 2018 software product compatibility requirements](#).
2. Obtain your DataPower files from IBM [Passport Advantage®](#).  
Note that the DataPower files from [Passport Advantage](#) have entitled services for API Connect users.  
  
See also [Download the firmware image from Passport Advantage](#).
3. Follow the DataPower Gateway installation information:  
Note: Set the timezone to UTC on all DataPower installations for use with API Connect.



- For gateways on physical appliances, see [DataPower Gateway 2018.4.1.x Installation](#)
  - For virtual gateways, see [Virtual DataPower Gateways](#).
4. Continue with [Configuring DataPower Gateway for API Connect](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring DataPower Gateway for API Connect

You can configure the IBM® DataPower® Gateway to prepare for a registration with the API Connect Management server.

### About this task

API Connect and DataPower supports two different types of gateway configurations. The DataPower Gateway (v5 compatible) provides the same support as the gateway support that was available with API Connect version 5.x. The DataPower API Gateway is an enhanced gateway that is performance-focused. See [API Connect gateway types](#) for more information about the differences between the gateway types.

Continue with the instructions for the type of Gateway Service you are configuring:

- [Configuring DataPower API Gateway](#)  
You can configure the DataPower API Gateway to prepare for a registration with the API Connect Management server.
- [Configuring DataPower Gateway \(v5 compatible\)](#)  
You can configure the DataPower Gateway (v5 compatible) to prepare for a registration with the API Connect Management server.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring DataPower API Gateway

You can configure the DataPower® API Gateway to prepare for a registration with the API Connect Management server.

### Before you begin

- These instructions are for DataPower Gateway deployments in a non-Kubernetes environment. Do not use these instructions if you installed API Connect on Kubernetes, with your DataPower Gateway Service also in the Kubernetes environment. To review deployment scenarios, see [Deploying DataPower Gateway](#).
- Ensure you have installed the version of DataPower Gateway that matches the version of the API Connect Management server. See [Installing DataPower Gateway](#).
- A shared certificate and private key is used for securing the communication between the API Connect Management server and the gateway. See [Generating keys and certificates](#) in the DataPower Gateway IBM documentation for instructions on how to create them with the DataPower tools.
- Ensure that the time zone for the DataPower Gateway is set to UTC.

### About this task

- These instructions provide the basic steps for configuring a gateway service with a single gateway server. The lowest-level configuration objects are created first, then used in other configuration objects. The procedures for configuring the two types of gateways are very similar, so only one procedure is provided. Any specific differences are identified.
- Adding gateways to configure a peering environment is similar to creating the first gateway, and is recommended for resiliency in a production environment. A minimum of three gateway servers in a gateway service is recommended for high availability. See [Gateway peering](#) for more information about configuring additional gateways for peering. See [Providing gateway service for API Connect](#) in the DataPower Gateways IBM Knowledge Center content for more details about the DataPower settings and procedures.

### Procedure

To configure a DataPower gateway to communicate with API Connect, complete the following steps:

**Note:** Use these instructions only for DataPower API Gateway . If you are configuring DataPower Gateway (v5 compatible) see [Configuring DataPower Gateway \(v5 compatible\)](#).

1. Open the DataPower WebGUI interface.  
Most of the configuration procedure is done in the DataPower WebGUI interface, not in the Blueprint Console.
2. Enable the XML management interface in the **default** domain, if required. The XML management interface is optional for DataPower API Gateway.
  - a. Search for **XML management interface** in the navigation search bar, and select it.
  - b. Set the Administrative state to enabled.
  - c. You can specify a different port number if you do not want to use the default of **5550**.
  - d. Select Apply to make the changes
  - e. Save changes to the default domain by selecting Save Configuration.
3. Create an application domain.  
This domain receives your traffic.

- a. Search for Application domain in the navigation search bar, and select it.
- b. Select Add to create the application domain.
- c. Enter a unique name for your domain.
- d. Ensure that enabled is selected for the Administrative state.
- e. Ensure that the default domain is listed in the Visible application domain list.
- f. Select Apply.
- g. Change to your new application domain by selecting Domain in the menu bar, and selecting the domain that you created.
- h. Select Save changes and switch domains.
  - All of the remaining steps on the DataPower gateway must be done in the application domain that you created.
  - i. Save changes to the domain by selecting Save Configuration.
4. Ensure that your deployment includes an NTP server to synchronize time between each of the DataPower Gateways.
  - See [Managing the NTP service](#).
5. Ensure that you have set a unique Appliance name (**System Identifier**) for each DataPower gateway. See [Initializing the DataPower Gateway](#).
6. Create a self-signed certificate and private key to be used to protect the traffic between the management server and the API gateway service process. You can generate a certificate and private key using DataPower or by using other tools, such as **OpenSSL**. See [Generating keys and certificates](#) in the DataPower Gateway IBM documentation for instructions on how to create a crypto key with the DataPower tools.
7. Upload your private crypto key file to the domain.
  - a. Search for Crypto key in the navigation search bar, and select it.
  - b. Select Add to create a key object.
  - c. Create a unique name for the key object in the *Name* field.
  - d. Select Upload....
  - e. Browse for the key file (which must be a .pem or .p12 file) and select it.
  - f. If you want to rename it, enter a new name for the file.
  - g. Select Upload to move it to the server in the cert:// folder.
  - h. Select Apply to save the changes.
8. Upload your crypto certificate file to the domain.
  - Note: If your certificate is signed by an Intermediate CA, you must include the entire chain in a single key file (either .pem or .p12) for uploading.
  - a. Search for Crypto certificate in the navigation search bar, and select it.
  - b. Select Add to create a certificate object.
  - c. Create a unique name for the certificate object in the *Name* field.
  - d. Select Upload....
  - e. Browse for the key file (which must be a .pem or .p12 file) and select it.
  - f. If you want to rename it, enter a new name for the file.
  - g. Select Upload to move it to the server in the cert:// folder.
  - h. Select Apply to save the changes.
9. Associate the Crypto key with the Crypto certificate by setting the Identification credential.
  - a. Search for Crypto Identification Credentials in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a name for your credential.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. In the Crypto Key field, select the name of the key object that you created from the drop-down menu.
  - f. In the Certificate field object, select the name of the certificate object that you created from the drop-down menu.
  - g. Select Apply to commit your changes.
10. Create your SSL Client profile.
  - a. Search for SSL Client profile in the navigation search bar, and select it.
  - b. Select Add to create a client profile.
  - c. Create a unique name for the profile in the *Name* field.
  - d. Select your Identification credential from the drop-down list.
  - e. Ensure that the value of *Validate server certificate* is set to off.
  - f. Select Apply to save the changes.
11. Create your SSL Server profile.
  - a. Search for SSL Server Profile in the navigation search bar, and select it.
  - b. Select Add to create a server profile.
  - c. Create a unique name for the profile in the *Name* field.
  - d. Select your Identification credential from the drop-down list.
  - e. Ensure that the value of *Request client authentication* is set to off.
  - f. Select Apply to save the changes.
12. For the DataPower API Gateway only: Define a configuration sequence.
  - The API Connect gateway service uses the configuration sequence to configure DataPower to implement the APIs that are defined in API Connect.
  - a. Search for Configuration sequence in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a name for your configuration sequence.
    - The name **apic-config** is not allowed because it is already used internally.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. Ensure that the value in the *Location profiles* field is set to local:///
    - This is the default value, so you might not need to change it.
  - f. Select the Access profile. See [Configuring the access profile for a configuration sequence](#) in the DataPower Gateway IBM Knowledge Center for instructions on how to create access profiles.
  - g. Change the value of the *Configuration execution interval* field to 3000.
    - The other fields can retain their default settings.
  - h. Select Apply to commit your changes.
13. Configure your gateway peering object for the API Connect Gateway Service.
  - This step is required when you set up a peer group of gateways, even if there is only a single gateway server in the gateway service.
  - a. Search for Gateway peering in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a unique name for your gateway peering object.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. Select a local address for the communications among the members of the peer group.
  - f. Select a local port for the communication.

- You can use the default value of 16380.
- g. Select a monitor port for the communication.  
You can use the default value of 26380.
  - h. Because this procedure uses only one gateway, ensure that Peer group mode is not selected.
  - i. Clear the Enable SSL checkbox. SSL is not needed for a single peer.
  - j. Set the Persistence location value to **Memory** for either physical DataPower appliance or virtual DataPower appliance.
  - k. Select Apply to commit your changes.
14. Configure your gateway peering object for rate limit information.  
Note: Version 2018.4.1.7 or later is required.
- a. Search for Gateway peering in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a unique name for your gateway peering object.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. Select a local address for the communications among the members of the peer group.
  - f. Select a local port for the communication.  
Use a unique port, different than the ports used for communication by other gateway peering objects.
  - g. Select a monitor port for the communication.  
Use a unique port, different than the ports used for monitoring by other gateway peering objects.
  - h. Because this procedure uses only one gateway, ensure that Peer group mode is not selected.
  - i. Clear the Enable SSL checkbox. SSL is not needed for a single peer.
  - j. Set the Persistence location value to **Memory** for either physical DataPower appliance or virtual DataPower appliance.
  - k. Select Apply to commit your changes.
15. Configure your gateway peering object for subscription information.  
Note: Version 2018.4.1.7 or later is required.
- a. Search for Gateway peering in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a unique name for your gateway peering object.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. Select a local address for the communications among the members of the peer group.
  - f. Select a local port for the communication.  
Use a unique port, different than the ports used for communication by other gateway peering objects.
  - g. Select a monitor port for the communication.  
Use a unique port, different than the ports used for monitoring by other gateway peering objects.
  - h. Because this procedure uses only one gateway, ensure that Peer group mode is not selected.
  - i. Clear the Enable SSL checkbox. SSL is not needed for a single peer.
  - j. Set the Persistence location value to **Memory** for either physical DataPower appliance or virtual DataPower appliance.
  - k. Select Apply to commit your changes.
16. Configure the gateway peering manager.  
Note: Version 2018.4.1.7 or later is required.
- a. Search for Gateway Peering Manager in the navigation search bar, and select it.
  - b. Set the Administrative state to enabled.
  - c. In the pull-down menu next to API Connect Gateway Service, select the gateway peering object configured in Step [13](#) for the API Connect Gateway Service.
  - d. In the pull-down menu next to Rate Limit, select the gateway peering object configured in Step [14](#) for rate limit information.
  - e. In the pull-down menu next to Subscription, select the gateway peering object configured in Step [15](#) for subscription.
  - f. Select Apply to commit your changes.
17. Set the API Connect Gateway service to define the communication interface with the API Connect Management server and for API transactions.
- a. Search for API Connect Gateway service in the navigation search bar, and select it.
  - b. Ensure that the Administrative state is set to enabled.
  - c. In the *Local address* field, enter the IP address of the DataPower gateway to which you want the traffic from the API Connect Management server to be sent.
  - d. Specify a value for Local Port. You can use the default port value of 3000, or specify a different port value.  
Note: The Local port specifies the port through which API Connect connects to manage the API Connect Gateway Service. Use this port when you configure a Gateway Service on API Connect. Beyond this port, the gateway service uses two additional consecutive ports after the defined local port to bind to a loopback address. Therefore, you must ensure that there are no conflicts on all three consecutive ports that start from the defined local port.
  - e. In the *SSL client* field drop-down list, select the name of the SSL client profile that you created.
  - f. In the *SSL server* field drop-down list, select the name of the SSL server profile that you created.
  - g. In the *API gateway address* field, enter the IP address for the DataPower gateway to which you want the API traffic sent.
  - h. Use the default port value of 9443 for the API gateway port.  
If the port is not being used by another service, you can also change it to port 443 if you want API transactions to be sent to the default port for HTTPS.
  - i. For DataPower API Gateway, set the Gateway Peering to **(none)**.  
When no gateway peering object is configured for the DataPower API Gateway, the peering configuration defined in the Gateway Peering Manager configuration is used.
  - j. Select whether you want the DataPower Gateway (v5 compatible) or the DataPower API Gateway.  
When the option is selected, it enables the registration of a DataPower Gateway (v5 compatible) gateway. Clear it to enable a DataPower API Gateway.
18. Register the gateway service in the API Connect Cloud Manager console:
- a. Open the API Connect Cloud Manager console.
  - b. Navigate to Configure Topology.
  - c. Select Register Service.
  - d. Select DataPower API Gateway for the DataPower API Gateway.
  - e. Add a title, name, and summary for the gateway connection.
  - f. Optional: Configure the OAuth Shared Secret.  
This setting allows OAuth tokens to be shared across multiple gateway services.
  - g. Enter one of the following values in the *API Invocation Endpoint* field:
    - IP address of the load balancer for the API transactions
    - IP address or host name of one of the gateways
 For example: `https://192.0.2.0:9443/`
  - h. Enter the one of the following values in the *Management Endpoint* field:
    - IP address of the load balancer for the management server traffic set to port 3000
    - IP address or hostname of one of the gateways
 For example: `https://192.0.2.0:3000/`
19. Select the default TLS Client Profile

20. Optional: Configure Server Name Indication (SNI) profiles.  
SNI profiles allow different TLS certificates to be used for API transaction requests from different host names.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring DataPower Gateway (v5 compatible)

You can configure the DataPower® Gateway (v5 compatible) to prepare for a registration with the API Connect Management server.

### Before you begin

- These instructions are for DataPower Gateway deployments in a non-Kubernetes environment. Do not use these instructions if you installed API Connect on Kubernetes, with your DataPower Gateway Service also in the Kubernetes environment. To review deployment scenarios, see [Deploying DataPower Gateway](#).
- Ensure you have installed the version of DataPower Gateway that matches the version of the API Connect Management server. See [Installing DataPower Gateway](#).
- A shared certificate and private key is used for securing the communication between the API Connect Management server and the gateway. See [Generating keys and certificates](#) in the DataPower Gateway IBM documentation for instructions on how to create them with the DataPower tools.
- Ensure that the time zone for the DataPower Gateway is set to UTC.

### About this task

- These instructions provide the basic steps for configuring a gateway service with a single gateway server. The lowest-level configuration objects are created first, then used in other configuration objects. The procedures for configuring the two types of gateways are very similar, so only one procedure is provided. Any specific differences are identified.
- Adding gateways to configure a peering environment is similar to creating the first gateway, and is recommended for resiliency in a production environment. A minimum of three gateway servers in a gateway service is recommended for high availability. See [Gateway peering](#) for more information about configuring additional gateways for peering. See [Providing gateway service for API Connect](#) in the DataPower Gateways IBM Knowledge Center content for more details about the DataPower settings and procedures.

### Procedure

To configure a DataPower gateway to communicate with API Connect, complete the following steps:

Note: Use these instructions only for DataPower Gateway (v5 compatible). If you are configuring DataPower API Gateway, see [Configuring DataPower API Gateway](#).

1. Open the DataPower WebGUI interface.  
Most of the configuration procedure is done in the DataPower WebGUI interface, not in the Blueprint Console.
2. Enable the XML management interface in the **default** domain, if required. The XML management interface is required for DataPower Gateway (v5 compatible). The XML management interface is optional for DataPower API Gateway.
  - a. Search for **XML management interface** in the navigation search bar, and select it.
  - b. Set the Administrative state to enabled.
  - c. You can specify a different port number if you do not want to use the default of **5550**.
  - d. Select Apply to make the changes
  - e. Save changes to the default domain by selecting Save Configuration.
3. Create an application domain.  
This domain receives your traffic.
  - a. Search for Application domain in the navigation search bar, and select it.
  - b. Select Add to create the application domain.
  - c. Enter a unique name for your domain.
  - d. Ensure that enabled is selected for the Administrative state.
  - e. Ensure that the default domain is listed in the Visible application domain list.
  - f. Select Apply.
  - g. Change to your new application domain by selecting Domain in the menu bar, and selecting the domain that you created.
  - h. Select Save changes and switch domains.  
All of the remaining steps on the DataPower gateway must be done in the application domain that you created.
    - i. Save changes to the domain by selecting Save Configuration.
4. Ensure that your deployment includes an NTP server to synchronize time between each of the DataPower Gateways.  
See [Managing the NTP service](#).
5. Ensure that you have set a unique Appliance name (**System Identifier**) for each DataPower gateway. See [Initializing the DataPower Gateway](#).
6. For the DataPower Gateway (v5 compatible) only: Enable statistics in the domain you created for **API Connect**.
  - a. Search for and select Statistics settings in the navigation search.
  - b. Select enabled for the Administrative state.
  - c. Select Apply.
7. Create a self-signed certificate and private key to be used to protect the traffic between the management server and the API gateway service process. You can generate a certificate and private key using DataPower or by using other tools, such as **OpenSSL**. See [Generating keys and certificates](#) in the DataPower Gateway IBM documentation for instructions on how to create a crypto key with the DataPower tools.
8. Upload your private crypto key file to the domain.
  - a. Search for Crypto key in the navigation search bar, and select it.
  - b. Select Add to create a key object.
  - c. Create a unique name for the key object in the *Name* field.
  - d. Select Upload....
  - e. Browse for the key file (which must be a .pem or .p12 file) and select it.

- f. If you want to rename it, enter a new name for the file.
  - g. Select Upload to move it to the server in the `cert://` folder.
  - h. Select Apply to save the changes.
9. Upload your crypto certificate file to the domain.
- Note: If your certificate is signed by an Intermediate CA, you must include the entire chain in a single key file (either .pem or .p12) for uploading.
- a. Search for Crypto certificate in the navigation search bar, and select it.
  - b. Select Add to create a certificate object.
  - c. Create a unique name for the certificate object in the *Name* field.
  - d. Select Upload....
  - e. Browse for the key file (which must be a .pem or .p12 file) and select it.
  - f. If you want to rename it, enter a new name for the file.
  - g. Select Upload to move it to the server in the `cert://` folder.
  - h. Select Apply to save the changes.
10. Associate the Crypto key with the Crypto certificate by setting the Identification credential.
- a. Search for Crypto Identification Credentials in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a name for your credential.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. In the Crypto Key field, select the name of the key object that you created from the drop-down menu.
  - f. In the Certificate field object, select the name of the certificate object that you created from the drop-down menu.
  - g. Select Apply to commit your changes.
11. Create your SSL Client profile.
- a. Search for SSL Client profile in the navigation search bar, and select it.
  - b. Select Add to create a client profile.
  - c. Create a unique name for the profile in the *Name* field.
  - d. Select your Identification credential from the drop-down list.
  - e. Ensure that the value of *Validate server certificate* is set to off.
  - f. Select Apply to save the changes.
12. Create your SSL Server profile.
- a. Search for SSL Server Profile in the navigation search bar, and select it.
  - b. Select Add to create a server profile.
  - c. Create a unique name for the profile in the *Name* field.
  - d. Select your Identification credential from the drop-down list.
  - e. Ensure that the value of *Request client authentication* is set to off.
  - f. Select Apply to save the changes.
13. Configure your gateway peering object for the API Connect Gateway Service.
- This step is required when you set up a peer group of gateways, even if there is only a single gateway server in the gateway service.
- a. Search for Gateway peering in the navigation search bar, and select it.
  - b. Select Add.
  - c. Enter a unique name for your gateway peering object.
  - d. Ensure that the Administrative state has a value of enabled.
  - e. Select a local address for the communications among the members of the peer group.
  - f. Select a local port for the communication.  
You can use the default value of 16380.
  - g. Select a monitor port for the communication.  
You can use the default value of 26380.
  - h. Because this procedure uses only one gateway, ensure that Peer group mode is not selected.
  - i. Clear the Enable SSL checkbox. SSL is not needed for a single peer.
  - j. Set the Persistence location value to **Memory** for either physical DataPower appliance or virtual DataPower appliance.
  - k. Select Apply to commit your changes.
14. Set the API Connect Gateway service to define the communication interface with the API Connect Management server and for API transactions.
- a. Search for API Connect Gateway service in the navigation search bar, and select it.
  - b. Ensure that the Administrative state is set to enabled.
  - c. In the *Local address* field, enter the IP address of the DataPower gateway to which you want the traffic from the API Connect Management server to be sent.
  - d. Use the default port value of 3000 for the Local port.
  - e. In the *SSL client* field drop-down list, select the name of the SSL client profile that you created.
  - f. In the *SSL server* field drop-down list, select the name of the SSL server profile that you created.
  - g. In the *API gateway address* field, enter the IP address for the DataPower gateway to which you want the API traffic sent.
  - h. Use the default port value of 9443 for the API gateway port.  
If the port is not being used by another service, you can also change it to port 443 if you want API transactions to be sent to the default port for HTTPS.
  - i. For DataPower Gateway (v5 compatible), select the gateway peering object that you created in Step [13](#).
  - j. Select whether you want the DataPower Gateway (v5 compatible) or the DataPower API Gateway.  
When the option is selected, it enables the registration of a DataPower Gateway (v5 compatible) gateway.
15. Register the gateway service in the API Connect Cloud Manager console:
- a. Open the API Connect Cloud Manager console.
  - b. Navigate to Configure Topology.
  - c. Select Register Service.
  - d. Select DataPower Gateway (v5 compatible) for the gateway that was available in version 5.
  - e. Add a title, name, and summary for the gateway connection.
  - f. Optional: Configure the OAuth Shared Secret.  
This setting allows OAuth tokens to be shared across multiple gateway services.
  - g. Enter one of the following values in the *API Invocation Endpoint* field:
    - IP address of the load balancer for the API transactions
    - IP address or host name of one of the gateways
For example: `https://192.0.2.0:9443/`
  - h. Enter the one of the following values in the *Management Endpoint* field:
    - IP address of the load balancer for the management server traffic set to port 3000
    - IP address or hostname of one of the gateways
For example: `https://192.0.2.0:3000/`
16. Select the default TLS Client Profile

17. Optional: Configure Server Name Indication (SNI) profiles.

SNI profiles allow different TLS certificates to be used for API transaction requests from different host names.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Sample configuration for multiple peering objects on gateway services external to Kubernetes

Sample for reconfiguring the API Connect domain configuration for your gateway services to include multiple peering objects.

- This reference shows how to configure API Connect domain configuration with gateway-peering configuration. To review domain configuration settings, see [Configuring DataPower Gateway for API Connect](#).
- This sample applies only to gateway services that are deployed external to Kubernetes, on either a physical or virtual appliance. This sample does not apply to gateway services that are deployed on Kubernetes.

The following examples show sample configuration for 3 different gateways, listed here with sample IP addresses of 1.1.1.1, 2.2.2.2, and 3.3.3.3. Note the following:

- The priority should be set differently on each of the three gateways. Set the lowest priority for the gateway that will run as primary.
- **persistence** should be set to **memory** for all configured peering objects.
- Optionally, add SSL configuration.

Configuration on gateway 1 (1.1.1.1)

```
top; configure terminal;
domain apiconnect; visible default; exit;

sw apiconnect;

loglevel debug;
logging target gwd-log
  type file
  format text
  timestamp syslog
  size 50000
  local-file logtemp:///gwd-log
  event apic-gw-service debug
exit

config-sequence "apiconnect"
location "local:/"
watch "on"
delete-unused "on"
match "(.*)\.cfg$"
summary "Toolkit Reboot configuration"
run-sequence-interval 3000
optimize-for-apic on
exit

crypto
key sth_apic sharedcert:///sth_apic-privkey.cer
certificate sth_apic sharedcert:///sth_apic-sscert.cer
idcred sth_apic sth_apic sth_apic
ssl-client gwd_to_mgmt
  idcred sth_apic
  no validate-server-cert
exit
ssl-server gwd_to_mgmt
  idcred sth_apic
  no request-client-auth
  validate-client-cert off
exit
exit

gateway-peering subs
admin enabled
local-address 1.1.1.1
local-port 15222
monitor-port 26222
priority 100
enable-ssl off
enable-peer-group on
peer 2.2.2.2
peer 3.3.3.3
persistence memory
exit

gateway-peering rate-limit
admin enabled
local-address 1.1.1.1
local-port 15223
monitor-port 26223
priority 100
enable-ssl off
enable-peer-group on
```

```

peer 2.2.2.2
peer 3.3.3.3
persistence memory
exit

```

```

gateway-peering gwd
admin enabled
local-address 1.1.1.1
local-port 15224
monitor-port 26224
priority 100
enable-ssl off
enable-peer-group on
peer 2.2.2.2
peer 3.3.3.3
persistence memory
exit

```

```

gateway-peering probe
admin enabled
local-address 1.1.1.1
local-port 15225
monitor-port 26225
priority 100
enable-ssl off
enable-peer-group on
peer 2.2.2.2
peer 3.3.3.3
persistence memory
exit

```

```

gateway-peering gws-rate-limit
admin enabled
local-address 1.1.1.1
local-port 15226
monitor-port 26226
priority 100
enable-ssl off
enable-peer-group on
peer 2.2.2.2
peer 3.3.3.3
persistence memory
exit

```

```

gateway-peering-manager
admin enabled
apic-gw-service gwd
rate-limit rate-limit
subscription subs
apiprobe probe
ratelimit-module gws-rate-limit

```

```

apic-gw-service
admin-state enabled
local-address 0.0.0.0
local-port 3000
api-gw-address 0.0.0.0
api-gw-port 9443
v5-compatibility-mode off
ssl-server gwd_to_mgmt
ssl-client gwd_to_mgmt
exit

```

```
write mem
```

Configuration on gateway 2 (2.2.2.2)

```

top; configure terminal;
domain apiconnect; visible default; exit;

sw apiconnect;

loglevel debug;
logging target gwd-log
type file
format text
timestamp syslog
size 50000
local-file logtemp:///gwd-log
event apic-gw-service debug
exit

config-sequence "apiconnect"
location "local:/// "
watch "on"
delete-unused "on"
match "(.*)\.cfg$"
summary "Toolkit Reboot configuration"
run-sequence-interval 3000
optimize-for-apic on
exit

crypto
key sth apic sharedcert:///sth_apic-privkey.cer
certificate sth apic sharedcert:///sth_apic-sscert.cer
idcred sth apic sth apic sth apic

```

```
ssl-client gwd_to_mgmt
idcred sth_apic
no validate-server-cert
exit
ssl-server gwd_to_mgmt
idcred sth_apic
no request-client-auth
validate-client-cert off
exit
exit
```

```
gateway-peering subs
admin enabled
local-address 2.2.2.2
local-port 15222
monitor-port 26222
priority 105
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 3.3.3.3
persistence memory
exit
```

```
gateway-peering rate-limit
admin enabled
local-address 2.2.2.2
local-port 15223
monitor-port 26223
priority 105
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 3.3.3.3
persistence memory
exit
```

```
gateway-peering gwd
admin enabled
local-address 2.2.2.2
local-port 15224
monitor-port 26224
priority 105
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 3.3.3.3
persistence memory
exit
```

```
gateway-peering probe
admin enabled
local-address 2.2.2.2
local-port 15225
monitor-port 26225
priority 105
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 3.3.3.3
persistence memory
exit
```

```
gateway-peering gws-rate-limit
admin enabled
local-address 2.2.2.2
local-port 15226
monitor-port 26226
priority 105
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 3.3.3.3
persistence memory
exit
```

```
gateway-peering-manager
admin enabled
apic-gw-service gwd
rate-limit rate-limit
subscription subs
apiprobe probe
ratelimit-module gws-rate-limit
exit
```

```
apic-gw-service
admin-state enabled
local-address 0.0.0.0
local-port 3000
api-gw-address 0.0.0.0
api-gw-port 9443
v5-compatibility-mode off
ssl-server gwd_to_mgmt
ssl-client gwd_to_mgmt
exit
```



```
write mem
```

Configuration on gateway 3 (3.3.3.3)

```
top; configure terminal;
domain apiconnect; visible default; exit;

sw apiconnect;

loglevel debug;
logging target gwd-log
type file
format text
timestamp syslog
size 50000
local-file logtemp:///gwd-log
event apic-gw-service debug
exit

config-sequence "apiconnect"
location "local:///"
watch "on"
delete-unused "on"
match "(.*)\.cfg$"
summary "Toolkit Reboot configuration"
run-sequence-interval 3000
optimize-for-apic on
exit

crypto
key sth_apic sharedcert:///sth_apic-privkey.cer
certificate sth_apic sharedcert:///sth_apic-sscert.cer
idcred sth_apic sth_apic sth_apic
ssl-client gwd_to_mgmt
idcred sth_apic
no validate-server-cert
exit
ssl-server gwd_to_mgmt
idcred sth_apic
no request-client-auth
validate-client-cert off
exit
exit

gateway-peering subs
admin enabled
local-address 3.3.3.3
local-port 15222
monitor-port 26222
priority 110
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 2.2.2.2
persistence memory
exit

gateway-peering rate-limit
admin enabled
local-address 3.3.3.3
local-port 15223
monitor-port 26223
priority 110
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 2.2.2.2
persistence memory
exit

gateway-peering gwd
admin enabled
local-address 3.3.3.3
local-port 15224
monitor-port 26224
priority 110
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 2.2.2.2
persistence memory
exit

gateway-peering probe
admin enabled
local-address 3.3.3.3
local-port 15225
monitor-port 26225
priority 110
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 2.2.2.2
persistence memory
exit
```

```
gateway-peering gws-rate-limit
admin enabled
local-address 3.3.3.3
local-port 15226
monitor-port 26226
priority 110
enable-ssl off
enable-peer-group on
peer 1.1.1.1
peer 2.2.2.2
persistence memory
exit
```

```
gateway-peering-manager
admin enabled
apic-gw-service gwd
rate-limit rate-limit
subscription subs
apiprobe probe
ratelimit-module gws-rate-limit
exit
```

```
apic-gw-service
admin-state enabled
local-address 0.0.0.0
local-port 3000
api-gw-address 0.0.0.0
api-gw-port 9443
v5-compatibility-mode off
ssl-server gwd_to_mgmt
ssl-client gwd_to_mgmt
exit
```

```
write mem
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Installing the toolkit

You can install the toolkit that provides CLI commands, and the API Designer user interface, for IBM® API Connect.

### About this task

The toolkit is provided as executable files, so no actual installation is necessary, you just need to download the required compressed file and extract the contents.

There are two toolkit options available:

- CLI: provides a command line environment for working with IBM API Connect.
- CLI + LoopBack + Designer: provides a command line environment for working with IBM API Connect, including LoopBack® support, and the API Designer user interface.

To install the toolkit, download the compressed file that is appropriate for your chosen toolkit option and platform, then extract the contents to a chosen location on your local machine. The compressed file contains an executable file for running CLI commands and, if you choose the CLI + LoopBack + Designer option, an executable file for launching the API Designer user interface.

You can download the toolkit compressed file in either of the following ways:

- From IBM Fix Central.
- From either the Cloud Manager or API Manager user interface.

The following table identifies the name of the compressed file that you need to download, depending on your chosen toolkit option and platform:

Table 1. Toolkit file names, by option and platform

Toolkit option	Mac OS X	Linux®	Windows
CLI	toolkit-mac.zip	toolkit-linux.tgz	toolkit-windows.zip
CLI + LoopBack + Designer	toolkit-loopback-designer-mac.zip	toolkit-loopback-designer-linux.tgz	toolkit-loopback-designer-windows.zip

## Procedure

To install and run the toolkit, complete the following steps:

1. Download the toolkit compressed file.
  - To download the toolkit from IBM Fix Central, complete the following steps:
    - Open the [IBM Fix Central site](#) in your browser.
    - In the Product selector field, enter `API Connect`, then select IBM API Connect from the drop down list.
    - Select your installed 2018.x.y version from the Installed Version list, then click Continue. If you do not know your installed IBM API Connect version, contact your administrator.
    - In the Text field, enter `toolkit`, then click Continue.
    - Select the required file, as identified in [Table 1](#).

- Note: When you download from IBM Fix Central, the release number is appended to the file name.
- Click Continue, then follow the instructions to complete the download operation.
  - To download the toolkit from either the Cloud Manager or API Manager user interface, complete the following steps:
    - Cloud Manager or API Manager user interface.
    - Select Help ( ? ) in the navigation.
    - Select Install API Connect CLI & API Designer.
    - Select CLI or CLI + LoopBack + Designer according to your preferred option.
    - Select your platform to download the toolkit compressed file.
    - Close the Install API Connect CLI & API Designer window.
2. Extract the contents of the toolkit compressed file to a folder of your choice.  
The contents of the file depend on the your chosen toolkit option and platform, as follows:

Table 2. Toolkit compressed file contents, by option and platform

Toolkit option	Mac OS X	Linux	Windows
CLI	apic-slim	apic-slim	apic-slim.exe
CLI + LoopBack + Designer	apic api_designer-mac.zip: contains the API Designer user interface application.	apic api_designer-linux	apic.exe api_designer-win.exe

The apic-slim or apic-slim.exe file is the CLI for IBM API Connect.

The apic or apic.exe file is the CLI for IBM API Connect including LoopBack support.

Tip: If you are using the CLI option, then if you rename the apic-slim file to apic, or the apic-slim.exe file to apic.exe, you can run the CLI commands exactly as documented, copy and paste sample commands from the documentation, and use any command scripts as-is if you later move to the CLI + LoopBack + Designer option.

The api\_designer-*platform* file is the API Designer user interface application for the specified platform.

3. Run the CLI.
- For the Mac OS X or Linux platforms, complete the following steps:
    - Open a terminal instance and navigate to the folder where you extracted the contents of the toolkit compressed file.
    - Make the CLI file an executable file by entering the following command:

```
chmod +x download_name
```

Where *download\_name* is the name of the toolkit file that you downloaded, either apic or apic-slim.

- Run CLI commands as follows:

```
./apic command_name_and_parameters
```

or

```
./apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

- For the Windows platform, complete the following steps:
  - Open a terminal window and navigate to the folder where you extracted the contents of the toolkit compressed file.
  - Run CLI commands as follows:

```
apic command_name_and_parameters
```

or

```
apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

Tip: Add the folder location of your CLI file to your PATH variable so that you can run CLI commands from anywhere in your file system.

4. Launch the API Designer user interface by running the application from the location to which you extracted the contents of the toolkit compressed file.

Note:

- To uninstall the API Designer application on a Windows platform with a non Administrator account, complete the following steps:
  - In Windows File Explorer, navigate to the USER\_HOME\AppData\Local\Programs\api-designer folder.
  - Run the **Uninstall API Designer application** application. Do **not** use the **Add or remove programs** window.
- To uninstall the API Designer application on a Windows platform with an Administrator account, you can either run the **Uninstall API Designer application** application, or you can use the **Add or remove programs** window.

## Results

The IBM API Connect toolkit CLI and, if selected, the API Designer user interface application are installed on your local system.

For information on using the API Designer user interface, see [Developing your APIs and applications](#).

For information on using the toolkit CLI, see [Using the developer toolkit command-line tool](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Upgrading in a VMware environment

You can upgrade an existing deployment of API Connect on VMware to a newer version while retaining your data.

- [Requirements for upgrading on VMware](#)  
Before upgrading API Connect on VMware, ensure that your deployment meets all the upgrade requirements.
- [Upgrading API Connect subsystems in a VMware environment](#)  
Complete the following steps to upgrade API Connect subsystems.
- [Troubleshooting the API Connect upgrade on VMware](#)  
Troubleshoot your API Connect upgrade on VMware.
- [Upgrading DataPower Gateway Service](#)  
Use these instructions when upgrading DataPower Gateway Service in a non-Kubernetes environment, during upgrade of API Connect Version 2018.4.1.0 or later.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading on VMware

Before upgrading API Connect on VMware, ensure that your deployment meets all the upgrade requirements.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

The instructions for upgrading apply to the *latest* Fix Pack version. Ensure that you are upgrading to the latest Fix Pack version. For information on the available Fix Packs, see [What's New in the latest release](#).

Note: For more information if you are moving to v2018.4.1.10, see the Tech note at [IBM Support](#).

See the requirements for the version you are upgrading from:

- [Requirements for upgrading from v2018.4.1.8 or later on VMware](#)  
Review the requirements for upgrading from Version 2018.4.1.8 or later on VMware.
- [Requirements for upgrading from v2018.4.1.7 on VMware](#)  
Review the requirements for upgrading from Version 2018.4.1.7 on VMware.
- [Requirements for upgrading from v2018.4.1.5 or v2018.4.1.6 on VMware](#)  
Review the requirements for upgrading from Version 2018.4.1.5 or Version 2018.4.1.6 on VMware.
- [Requirements for upgrading from v2018.4.1.4 or earlier on VMware](#)  
Review the requirements for upgrading from Version 2018.4.1.4 or earlier on VMware.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading from v2018.4.1.8 or later on VMware

Review the requirements for upgrading from Version 2018.4.1.8 or later on VMware.

Restriction:

- A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.
- You cannot upgrade directly to FP24 from versions older than FP20; you must upgrade to FP20 first.
- Upgrading to FP20 from any version older than FP15 requires that you first upgrade to at least one version in the FP15-to-FP19 range.
- Before starting the upgrade, verify that API Connection deployment is fully operational. See [Checking cluster health on VMware](#).  
When upgrading to v2018.4.1.10 or later, this step is optional because a health check will be run automatically as part of the `apicup subsys install` command in [Upgrading API Connect subsystems in a VMware environment](#). Note that you might still want to run the health check when preparing for the upgrade, to ensure the health of the backup image you must create prior to running `apicup subsys install`.
- When upgrading from v2018.4.1.13 or later, use the following command to remove any previously used upgrade files:  

```
apic clean-upgrade-files
```
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management subsystem in VMware environments](#), [Backing up and restoring the Developer Portal in a VMware environment](#) and [Backing up and restoring the analytics database](#).

- Ensure that port 9178 is open between the Analytics, Management, and Portal subsystems and the system running `apicup`.  
An i/o timeout error can occur if the firewall rejects traffic to the appliance. Make sure port 9178 is open and try again. Example error message:

```
apicup subsys install test-analytics upgrade_analytics_lts_v2018.4.1.5.tgz
INFO[0000] Preparing update
INFO[0010] Sending update packet to cluster (connected to testrv0192.subnet1.example.com)
Error: failed to install the subsystem: unable to open file send stream:
rpc error: code = Unavailable desc = all SubConns are in TransientFailure,
latest connection error: connection error:
desc = "transport: Error while dialing dial tcp 1.2.3.4:9178: i/o timeout"
```

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a VMware deployment](#).
- If you are upgrading any subsystem that has less than three nodes and was installed in **standard** mode, before you upgrade you must convert to **Dev** mode and remove replicaset. Conversion is necessary because **standard** mode is not supported on less than three nodes. See [Converting installation mode](#). To review installation modes, see [Requirements for initial deployment on VMware](#).
- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading from v2018.4.1.7 on VMware

Review the requirements for upgrading from Version 2018.4.1.7 on VMware.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- Before starting the upgrade, verify that API Connection deployment is fully operational. See [Checking cluster health on VMware](#). When upgrading to v2018.4.1.10 or later, this step is optional because a health check will be run automatically as part of the `apicup subsys install` command in [Upgrading API Connect subsystems in a VMware environment](#). Note that you might still want to run the health check when preparing for the upgrade, to ensure the health of the backup image you must create prior to running `apicup subsys install`.
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management subsystem in VMware environments](#), [Backing up and restoring the Developer Portal in a VMware environment](#) and [Backing up and restoring the analytics database](#).

- The upgrade of the management server *from* Version 2018.4.1.7 or earlier may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data, such as greater than 10 GB.

To reduce the amount of time required for the upgrade of the management database, use the `apicops` command to truncate the subscriber event table and remove unused snapshots:

```
apicops subscriber-queues:clear
apicops snapshots:clean-up
```

Use the latest release of the `apicops` command. Download it from <https://github.com/ibm-apiconnect/apicops/releases>.

- Ensure that port 9178 is open between the Analytics, Management, and Portal subsystems and the system running `apicup`. An i/o timeout error can occur if the firewall rejects traffic to the appliance. Make sure port 9178 is open and try again. Example error message:

```
apicup subsys install test-analytics upgrade_analytics_lts_v2018.4.1.5.tgz
INFO[0000] Preparing update
INFO[0010] Sending update packet to cluster (connected to testsrv0192.subnet1.example.com)
Error: failed to install the subsystem: unable to open file send stream:
rpc error: code = Unavailable desc = all SubConns are in TransientFailure,
latest connection error: connection error:
desc = "transport: Error while dialing dial tcp 1.2.3.4:9178: i/o timeout"
```

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a VMware deployment](#).
- If you are upgrading any subsystem that has less than three nodes and was installed in **standard** mode, before you upgrade you must convert to **Dev** mode and remove replicaset. Conversion is necessary because **standard** mode is not supported on less than three nodes. See [Converting installation mode](#). To review installation modes, see [Requirements for initial deployment on VMware](#).

- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Requirements for upgrading from v2018.4.1.5 or v2018.4.1.6 on VMware

Review the requirements for upgrading from Version 2018.4.1.5 or Version 2018.4.1.6 on VMware.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- Before starting the upgrade, verify that API Connection deployment is fully operational.
  - For Version 2018.4.1.6 or later, see [Checking cluster health on VMware](#).  
When upgrading *from* v2018.4.1.6 or later *to* v2018.4.1.10 or later, this step is optional because a health check will be run automatically as part of the `apicup subsys install` command in [Upgrading API Connect subsystems in a VMware environment](#). Note that you might still want to run the health check when preparing for the upgrade, to ensure the health of the backup image you must create prior to running `apicup subsys install`.
  - For Version 2018.4.1.5, see [Determining status of a cluster on VMware](#).
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management subsystem in VMware environments](#), [Backing up and restoring the Developer Portal in a VMware environment](#) and [Backing up and restoring the analytics database](#).

- The upgrade of the management server *from* Version 2018.4.1.7 or earlier may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data, such as greater than 10 GB.

To reduce the amount of time required for the upgrade of the management database, use the `apicops` command to truncate the subscriber event table and remove unused snapshots:

```
apicops subscriber-queues:clear
apicops snapshots:clean-up
```

Use the latest release of the `apicops` command. Download it from <https://github.com/ibm-apiconnect/apicops/releases>.

- When upgrading from Version 2018.4.1.5 or Version 2018.4.1.6, including any iFixes, obtain from Fix Central the Control Plane file that is required to upgrade the Kubernetes version to a supported version.

Version to upgrade from	Version to upgrade to	Control Plane file
2018.4.1.6	2018.4.1.9	<code>appliance-control-plane-1.14.1.tgz</code>
2018.4.1.5		

- Install the Control Plane file with each subsystem.
- `upgrade_management_lts_v2018.4.1.x.tgz` contains the required control-plane version(1.16.x)
- Note that Version 2018.4.1.10 the Control Plane artifact file is minor version 16 (major.minor.patch). Any version of the control plan matching the same minor version will work:

```
appliance-control-plane-1.16.[x].tgz
```

- You can install the Control Plane file and the upgrade file for the subsystem at the same time.  
For example, to upgrade the management subsystem from 2018.4.1.7 to 2018.4.1.10:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.10.tgz appliance-control-plane-1.15.[x].tgz
```

For example, to upgrade the management subsystem from 2018.4.1.5-`ifix1.0` to 2018.4.1.10:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.10.tgz appliance-control-plane-1.15.[x].tgz
appliance-control-plane-1.16.[x].tgz
```

- The ordering of files passed to `apicup subsys install` does not matter.
- If unsure of which files are required for upgrade, running `apicup subsys install` will interrogate the subsystem and error if any required files are missing. For example, upgrading from 2018.4.1.7 to 2018.4.1.10, but missing a control plane file:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.10.tgz
Error: failed to install the subsystem: Unable to execute appliance plan: Missing minor version 16 in control plane
upgrade path
```

The error indicates that you also need to provide a control plane artifact that is minor version 16 (major.minor.patch). Any version of the control plan matching the same minor version will work.

Notes:

- Install the Control Plane file with each subsystem.
- For Version 2018.4.1.9, the Control Plane artifact file is minor version 14 (major.minor.patch). Any version of the control plan matching the same minor version will work:

```
appliance-control-plane-1.14.[x].tgz
```

- You can install the Control Plane file and the upgrade file for the subsystem at the same time. For example, to upgrade the management subsystem from 2018.4.1.5-`ifix1.0` to 2018.4.1.9:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.9.tgz appliance-control-plane-1.14.[x].tgz
```

- The ordering of files passed to `apicup subsys install` does not matter.
- If unsure of which files are required for upgrade, running `apicup subsys install` will interrogate the subsystem and error if any required files are missing. For example, upgrading from 2018.4.1.5-`ifix1.0` to 2018.4.1.9, but missing a control plane file:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.9.tgz
Error: failed to install the subsystem: Unable to execute appliance plan: Missing minor version 14 in control plane
upgrade path
```

The error indicates that you also need to provide a control plane artifact that is minor version 14 (major.minor.patch). Any version of the control plan matching the same minor version will work.

- Ensure that port 9178 is open between the Analytics, Management, and Portal subsystems and the system running `apicup`. An i/o timeout error can occur if the firewall rejects traffic to the appliance. Make sure port 9178 is open and try again. Example error message:

```
apicup subsys install test-analytics upgrade_analytics_lts_v2018.4.1.5.tgz
INFO[0000] Preparing update
INFO[0010] Sending update packet to cluster (connected to teststrv0192.subnet1.example.com)
Error: failed to install the subsystem: unable to open file send stream:
rpc error: code = Unavailable desc = all SubConns are in TransientFailure,
latest connection error: connection error:
desc = "transport: Error while dialing dial tcp 1.2.3.4:9178: i/o timeout"
```

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a VMware deployment](#).
- If you are upgrading any subsystem that has less than three nodes and was installed in `standard` mode, before you upgrade you must convert to `Dev` mode and remove replicaset. Conversion is necessary because `standard` mode is not supported on less than three nodes. See [Converting installation mode](#). To review installation modes, see [Requirements for initial deployment on VMware](#).
- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, `myProject`) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
- Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM® API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Requirements for upgrading from v2018.4.1.4 or earlier on VMware



Review the requirements for upgrading from Version 2018.4.1.4 or earlier on VMware.

Restriction: A direct upgrade from any version of 2018 to API Connect 10.0.6.0 or later is not supported. You must upgrade to 10.0.5.x as an interim step before upgrading to 10.0.6.0.

- If you want to upgrade *from* an API Connect deployed version, 2018.4.1.4 or earlier *to* Version 2018.4.1.9 or later, you must first upgrade to Version 2018.4.1.5. You can then upgrade directly from Version 2018.4.1.5 to Version 2018.4.1.9 or later.

To upgrade *from* 2018.4.1.4 or earlier *to* 2018.4.1.5, you must upgrade the Fix Packs in the correct sequence. The sequence is required because Kubernetes imposes restrictions on skipping Kubernetes levels when upgrading. Complete the upgrade of all subsystems of one version before starting the upgrade to the next version. See the following table.

Deployed version	Required upgrade sequence
2018.4.1.0, 2018.4.1.1, or 2018.4.1.2	1. Version 2018.4.1.3 2. Version 2018.4.1.4 3. Version 2018.4.1.5
2018.4.1.3	1. Version 2018.4.1.4 2. Version 2018.4.1.5
2018.4.1.4	1. Version 2018.4.1.5

- Before starting the upgrade, verify that API Connection deployment is fully operational. On v2018.4.1.4 or earlier, see [Determining status of a cluster on VMware](#).
- When upgrading API Connect, complete a manual backup of the management database, Portal subsystem, and Analytics subsystem just prior to starting the upgrade. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management subsystem in VMware environments](#), [Backing up and restoring the Developer Portal in a VMware environment](#) and [Backing up and restoring the analytics database](#).

- The upgrade of the management server *from* Version 2018.4.1.7 or earlier may take longer than during previous fix pack upgrades. The extended time is due to underlying schema changes, and can be as long as several hours for long established deployments with a large amount of data, such as greater than 10 GB.

To reduce the amount of time required for the upgrade of the management database, use the `apicops` command to truncate the subscriber event table and remove unused snapshots:

```
apicops subscriber-queues:clear
apicops snapshots:clean-up
```

Use the latest release of the `apicops` command. Download it from <https://github.com/ibm-apiconnect/apicops/releases>.

- Ensure that port 9178 is open between the Analytics, Management, and Portal subsystems and the system running `apicup`. An i/o timeout error can occur if the firewall rejects traffic to the appliance. Make sure port 9178 is open and try again. Example error message:

```
apicup subsys install test-analytics upgrade_analytics_lts_v2018.4.1.5.tgz
INFO[0000] Preparing update
INFO[0010] Sending update packet to cluster (connected to teststrv0192.subnet1.example.com)
Error: failed to install the subsystem: unable to open file send stream:
rpc error: code = Unavailable desc = all SubConns are in TransientFailure,
latest connection error: connection error:
desc = "transport: Error while dialing dial tcp 1.2.3.4:9178: i/o timeout"
```

- If not already set up, send your deployment logs to a remote server. The upgrade process terminates pods, and the logs will be lost if not stored remotely. To send your deployment logs to a remote server, see [Configuring remote logging for a VMware deployment](#).
- If you are upgrading any subsystem that has less than three nodes and was installed in `standard` mode, before you upgrade you must convert to `Dev` mode and remove replicaset. Conversion is necessary because `standard` mode is not supported on less than three nodes. See [Converting installation mode](#). To review installation modes, see [Requirements for initial deployment on VMware](#).
- When upgrading to Version 2018.4.1.4 or later from Version 2018.4.1.3 (or earlier), and your subsystem was installed in `standard` mode, be sure to specify `mode=standard` as a parameter to the `apicup subsys set` `[SUBSYSTEM-NAME]` command. For Version 2018.4.1.4, the default mode switched from `standard` to `dev`. If you do not specify the `mode` parameter for Version 2018.4.1.4, the installation will proceed in `dev` mode. This is a change in behavior from Version 2018.4.1.3 and earlier. For Version 2018.4.1.4, if you are deploying into a production environment, such as a cluster or high availability deployment, you should use `standard` mode:

```
apicup subsys set [SUBSYS-NAME] mode=standard
```

- In Version 2018.4.1.4 the default value for subsystem `mode` configuration has changed from `standard` to `dev`. If you explicitly specify (`set`) the `mode` value when configuring your subsystems, this change does not impact you. If you are unsure whether you specified it, set it on each Version 2018.4.1.4 subsystem with the following command: `apicup subsys set [SUBSYSTEM-NAME] mode=[desired mode]`. For Version 2018.4.1.4, if you are deploying into a production environment, such as a cluster or high availability deployment, you should use `standard` mode, so be sure to set it. For example:

To review installation modes, see [Converting installation mode](#).

- If you are upgrading to IBM® API Connect Version 2018.4.1.0, some default values from `apiconnect-up.yml` have changed and you might receive validation errors when you run the `apicup subsys get/install` command. To bypass the validation, you can add the `--no-verify` flag to the `apicup subsys install` commands.
- If you are upgrading from Version 2018.4.1.4 or earlier, ensure that your deployment has addressed all known security vulnerabilities with SSH. New installations of Version 2018.4.1.5 and later prevent use of weak SSH Message Authentication Code (MAC) and Key exchange algorithms (KexAlgorithms) with TCP over port 22. Use of weak algorithms can allow an attacker to recover the plain text message from the encrypted text. However, the upgrade to Version 2018.4.1.5 must necessarily retain your existing SSH configuration. If you have not yet addressed the security vulnerabilities with weak MACs and KexAlgorithms, manually update your configuration:

1. Add the following lines to `/etc/ssh/sshd_config`:

```
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com
```

2. Restart the SSH daemon:

```
sudo systemctl restart sshd
```



- If you are upgrading Analytics subsystem from Version 2018.4.1.3 or earlier, and you are upgrading an Analytics subsystem that has only *one* node, and was installed in `dev` mode, you must remove replicaset before you upgrade. Use `ssh` to access the VM, and use the following command:

```
sudo kubectl --kubeconfig /etc/kubernetes/admin.conf delete rs --all
```

For Version 2018.4.1.4 or later, when Analytics is installed in `dev` mode, you do not need to remove replicaset.

- If you are upgrading *from* Version 2018.4.1.0 to Version 2018.4.1.1, Version 2018.4.1.2 or Version 2018.4.1.3 in a multi-node OVA environment: When upgrading a multi-node cluster from Version 2018.4.1.0 only, create an update directory on each cluster member before running `apicup subsys install`:

Note: This step is not needed on Version 2018.4.1.1 or later.

```
$ sudo mkdir -p /var/lib/apiconnect/upgrades/
$ sudo chmod 700 /var/lib/apiconnect/upgrades/
$ sudo chown _apt /var/lib/apiconnect/upgrades/
```

If you do not precisely follow the previous instructions, the upgrade operation might fail, with errors similar to the following being written repeatedly to the log file on one of the nodes:

```
Dec 05 20:23:14 myhostname apic[5541]: time="2018-12-05T20:23:14Z" level=debug msg="podStage.Upgrade waiting for ip:1.2.3.4-kube-scheduler-xyzdev123 KUBE_SCHEDULER_POD(Running) gcr.io/google_containers/kube-scheduler-amd64:v1.10.6 => gcr.io/google_containers/kube-scheduler-amd64:v1.12.3"
```

To resolve this problem, run the following commands on each node:

```
$ sudo chmod 700 /var/lib/apiconnect/upgrades/
$ sudo apt-get update
$ sudo apt-get upgrade -y appliance-base (press enter in response to any prompts)
$ sudo systemctl restart appliance-manager
```

- **Troubleshooting tip:** After upgrading to 2018.4.1.4, if kubelet fails to start with the error `no space left on device`, run the following command on each virtual machine:

```
sudo sysctl -p /etc/sysctl.d/10-sysctl-appliance.conf
```

- **Troubleshooting tip:** If your Analytics subsystem does not start after upgrading to 2018.4.1.3, ensure that you removed ReplicaSets as described in [the requirement on this page for removing replicaset](#). If you removed the ReplicaSets and find that the Analytics pod still has replicas, such as version 2018.4.1.1 (running) and version 2018.4.1.3 (pending), delete the running pod with 2018.4.1.1, and the pending pod with the 2018.4.1.3 will start. This applies to both analytics-client and analytics-ingestion.

- Download the Install Assist (APICUP) installer from the same URL where you download your Fix Pack version. You must use the latest Install Assist package with the latest product version.
- The original project directory created with the APICUP installer during the initial product installation (for example, *myProject*) is required to both restore the database and to upgrade your deployment. You cannot restore the database or perform an upgrade without the initial project directory because it contains pertinent information about the cluster. A good practice is to back up the original project directory to a location from which it can always be retrieved.
- If the upgrade is initiated from the original project directory, the certificates are automatically copied into the upgraded version. The original project directory contains the `apiconnect-up.yml` file. An upgrade using the same project directory as the one used for installation will transfer the `encryption-secret` and all other certificates for the project to the upgraded version.
- All external traffic to the management server must be blocked during the upgrade of the management subsystem. This restriction applies to traffic from all sources via the user interfaces (Cloud Manager, API Manager), CLI, and REST API. The Cloud Manager and API Manager UIs cannot be used during the upgrade of the management subsystem. The Developer Portal also cannot be used to create, update, or delete any applications, subscriptions, memberships, or consumer organizations during the time.
- The upgrade order for subsystems is important. Upgrade the subsystems in the following order: (1) Management, (2) Analytics, (3) Portal, (4) Gateway. Analytics and Portal may be switched between second or third, but Management must be upgraded first. Gateway must be upgraded after Management for the following reasons:
  - Upgrading the Management service before the Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
  - After a completed upgrade, the Management server and Gateway firmware versions must match, for example, APIC 2018.4.1.2 with DP 2018.4.1.2.
 Wait until the upgrade is finished on a subsystem before proceeding to the next one.
- Downgrading any of the IBM API Connect subsystems to an earlier fix pack level is **not** supported.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Upgrading API Connect subsystems in a VMware environment

Complete the following steps to upgrade API Connect subsystems.

### Before you begin

Restriction:

- You cannot upgrade directly to FP24 from versions older than FP20; you must upgrade to FP20 first.
- Upgrading to FP20 from any version older than FP15 requires that you first upgrade to at least one version in the FP15-to-FP19 range.
- Ensure that you have met the requirements for upgrading API Connect subsystems in a VMware environment. See [Requirements for upgrading on VMware](#).
- Ensure that you are upgrading to the *latest* Fix Pack version. These instructions apply to upgrading to the *latest* Fix Pack version. To access the Fix Packs, see the link in [What's New in the latest release](#).

## About this task

When you apply an upgrade, the new level of the subsystem overwrites the existing level. Your user configuration, APIs, Products, and subsystem configurations (Management, Analytics, and Developer Portal) are retained.

Note: **Upgrading to 2018.4.1.13 - iFix3.0 or later:** Permissions are added to the pre-supplied API Connect user roles as follows:

- A role that had the `Api-Drafts:View` permission, but not the `Api-Drafts:Manage` permission, has the `Product-Drafts:View` permission added if not already present.
- A role that has both the `Api-Drafts:View` and `Api-Drafts:Manage` permissions has the `Product-Drafts:View` and `Product-Drafts:Manage` permissions added if not already present.

The permission settings for custom roles are not changed.

For more information on user roles, and assigning permissions to roles, see [API Connect user roles](#) and [Creating custom roles](#).

## Procedure

1. Verify the API Connect deployment is healthy and fully operational:

Version to upgrade from	Instructions
v2018.4.1.6 or later	See <a href="#">Checking cluster health on VMware</a> When upgrading <i>from</i> v2018.4.1.6 or later <i>to</i> v2018.4.1.10 or later, this step is optional because a health check is run automatically as part of the <code>apicup subsys install</code> command in step 7. Note that you might still want to run the health check now, as preparation for the manual backup in step 2.
v2018.4.1.5 or earlier	See <a href="#">Determining status of a cluster on VMware</a>

2. Complete a manual backup of the API Connect subsystems. See [Backing up and restoring](#).

3. If necessary, prepare your management database for the upgrade:

Version to upgrade from	Instructions
v2018.4.1.8 or later	No preparation required. Skip this step, go directly to Step 4
v2018.4.1.7 or earlier	Due to schema changes, upgrade of the management database takes longer than previous upgrades. For long established deployments with a large amount of data, such as over 10 GB, upgrade can take as long as several hours. To reduce this time, use the <code>apicops</code> interface to truncate the subscriber event table and remove unused snapshots.  Download the latest release of the <code>apicops</code> interface from <a href="https://github.com/ibm-apiconnect/apicops/releases">https://github.com/ibm-apiconnect/apicops/releases</a> , and run the following commands:  <pre>apicops subscriber-queues:clear apicops snapshots:clean-up</pre>

4. Download the appropriate images from IBM® Fix Central.

To access the Fix Packs, see the link in [What's New in the latest release](#). On the Fix Pack page, select the version you want to install. When the version contents are displayed, access the files by clicking the link **Status: Available**.

The upgrade files are distributed in compressed tar format. The filename structure is:

```
upgrade_management_lts_<version>.tgz
upgrade_analytics_lts_<version>.tgz
upgrade_portal_lts_<version>.tgz
```

5. If necessary, download from the same Fix Pack page any Control Plane files that are needed.

Control Plane files provide support for specific Kubernetes versions. The file `upgrade_management_lts_<version>.tgz` contains the latest Control Plane file. An upgrade from the most recent API Connect version to the current version does not need a separate Control Plane file. However, when upgrading from older versions of API Connect, you must install one or more control plane files to ensure that all current Kubernetes versions are supported.

Consult the following table to see if your deployment needs one or more separate Control Plane files.

Note: If you want to upgrade to an older fix pack than the current release, see the Control Plane lists in [Control planes needed for upgrading to earlier fix packs](#).

Version to upgrade from	Instructions for upgrading to v2018.4.1.24
v2018.4.1.20 and iFixes	No control plane needed.
v2018.4.1.19 and iFixes v2018.4.1.18 was not released	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> </ul>
v2018.4.1.17 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> </ul>
v2018.4.1.16 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>

Version to upgrade from	Instructions for upgrading to v2018.4.1.24
v2018.4.1.15 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.12 and iFixes</li> <li>• v2018.4.1.11 and iFixes</li> <li>• v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.9 and iFixes</li> <li>• v2018.4.1.8 and iFixes</li> <li>• v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.6 and iFixes</li> <li>• v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> <li>• <code>appliance-control-plane-1.14.x.tgz</code></li> </ul>

- Control Plane files are distributed on the same Fix Pack page as the API Connect distribution files.
- You will install the Control Plane files with each subsystem as part of Step 7.

6. Download the version of Install Assist (`apicup`) that matches your upgrade version of API Connect.

7. For each API Connect subsystem, in turn, run the `install` command. The syntax is:

```
apicup subsys install [SUBSYS-NAME] [path_to_subsystem_upgrade_tar_archive]
```

- When you run the `install`, the program sends the compressed tar file, which contains the upgrade images, to all cluster members. The compressed tar file is about 2 GB, and transfer can take some time. When the `install` command exits, the compressed tar file has arrived at each member. The upgrade process is now underway, and might continue for some time depending on factors such as the size of your deployment, your network speed, etc.
- If you downloaded Control Plane files in Step 5, install them at the same time as each subsystem. The syntax is:

```
apicup subsys install [SUBSYS-NAME] [path_to_subsystem_upgrade_tar_archive] [path_to_control_plane_file]
```

- You must install the Control Plane file and the upgrade file for the subsystem at the same time. You can install multiple Control Plane files with one `apicup subsys install` command. The ordering of files passed to `apicup subsys install` does not matter.
  - For example, to upgrade the management subsystem from 2018.4.1.7 to 2018.4.1.12:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.10.tgz appliance-control-plane-1.15.[x].tgz
```

- For example, to upgrade the management subsystem from 2018.4.1.5-ifix1.0 to 2018.4.1.12:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.10.tgz appliance-control-plane-1.15.[x].tgz appliance-control-plane-1.14.[x].tgz
```

- If unsure of which files are required for upgrade, running `apicup subsys install` interrogates the subsystem and returns an error if any required files are missing. For example, upgrading from 2018.4.1.7 to 2018.4.1.12, but missing a control plane file:

```
$ apicup subsys install management upgrade_management_lts_v2018.4.1.12.tgz
Error: failed to install the subsystem: Unable to execute appliance plan: Missing minor version 15 in control plane upgrade path
```

The error indicates that you also need to provide a Control Plane artifact that is *minor* version 15. Note that the Control Plane artifact file naming convention is *major.minor.patch*. Any *patch* version for the same *minor* version will work:

```
appliance-control-plane-1.15.[x].tgz
```

- The `apicup subsys install` command automatically runs `apicup health-check` prior to attempting the upgrade. An error is displayed if a problem is found that will prevent successful upgrade.
8. For each subsystem, verify that the upgrade was successful.
- Use `ssh` to access the appliance and run `sudo apic status`. Verify that the `Upgrade stage` property has the value `UPGRADE_DONE`. Note that after the upgrade completes, it can take several minutes for all servers to start. If you see the error message `Subsystems not running`, wait a few minutes, try the command again, and review the output in the `Status` column of the Pods Summary section. For example, here is output from an upgrade of the Analytics subsystem.

```
#sudo apic status
INFO[0000] Log level: info

Cluster members:
- testsrv0233.subnet1.example.com (1.2.152.233)
  Type: BOOTSTRAP MASTER
  Install stage: DONE
  Upgrade stage: UPGRADE_DONE
  Docker status:
  Systemd unit: running
  Kubernetes status:
  Systemd unit: running
  Kubelet version: testsrv0233 (4.4.0-138-generic) [Kubelet v1.10.6, Proxy v1.10.6]
  Etd status: pod etcd-testsrv0233 in namespace kube-system has status Running
  Addons: calico, dns, helm, kube-proxy, metrics-server, nginx-ingress,
Etd cluster state:
- etcd member name: testsrv0233.subnet1.example.com, member id: 12836860275847862867, cluster id:
14018872452420182423,
  leader id: 12836860275847862867, revision: 365042, version: 3.1.17
```

Pods Summary:

NODE STATUS	NAMESPACE REASON	NAME	READY
	default	apic-analytics-analytics-client-76956644b9-cmgx8	0/0
Pending			
testsrv0233	default	apic-analytics-analytics-client-76956644b9-v1qp2	1/1
Running			
testsrv0233	default	apic-analytics-analytics-cronjobs-retention-1541381400-hp9fc	0/1
Succeeded			
testsrv0233	default	apic-analytics-analytics-cronjobs-rollover-1541445300-c5n6z	0/1
Succeeded			
	default	apic-analytics-analytics-ingestion-547f875467-8mhs1	0/0
Pending			
testsrv0233	default	apic-analytics-analytics-ingestion-547f875467-s7flj	1/1
Running			
	default	apic-analytics-analytics-mtls-gw-85b8676855-jmh8c	0/0
Pending			
testsrv0233	default	apic-analytics-analytics-mtls-gw-85b8676855-sw6ps	1/1
Running			
testsrv0233	default	apic-analytics-analytics-storage-basic-8cckh	1/1
Running			
testsrv0233	kube-system	calico-node-8crtp	2/2
Running			
testsrv0233	kube-system	coredns-87cb95869-6flvn	1/1
Running			
testsrv0233	kube-system	coredns-87cb95869-rccvb	1/1
Running			
testsrv0233	kube-system	etcd-testsrv0233	1/1
Running			
testsrv0233	kube-system	ingress-nginx-ingress-controller-f7b9z	1/1
Running			
testsrv0233	kube-system	ingress-nginx-ingress-default-backend-6f58fb5f56-nklmv	1/1
Running			
testsrv0233	kube-system	kube-apiserver-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-apiserver-proxy-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-controller-manager-testsrv0233	1/1
Running			
testsrv0233	kube-system	kube-proxy-2vw9b	1/1
Running			
testsrv0233	kube-system	kube-scheduler-testsrv0233	1/1
Running			
testsrv0233	kube-system	metrics-server-5558db4678-9drz6	1/1
Running			
testsrv0233	kube-system	tiller-deploy-84f4c8bb78-vx65c	1/1
Running			

- To check the status of a cluster on v2018.4.1.6 or later, see [Checking cluster health on VMware](#).
- After completing an upgrade of the Portal subsystem, there may be a delay while the existing sites are upgraded to the new platform version. Once all Portal pods have been upgraded, run the following commands from your project directory. To see the progress of sites being upgraded to the new platform version, use the following commands:
  - `apicup subsys exec portal list-sites sites`  
Any sites currently upgrading will be listed as `UPGRADING`. Once all sites have finished upgrading they should have the `INSTALLED` status and the new platform version listed.
  - Once all sites are in `INSTALLED` state and have the new platform listed, run:  
`apicup subsys exec portal list-sites platforms`  
The new version of the platform should be the only platform listed.

Important: DO NOT reboot any of the virtual machines until ALL of the Portal sites are in `INSTALLED` state and there is only one platform returned, even if you're instructed to do so by the `apicup subsys health-check` during the upgrade process.

9. When the upgrade completes successfully, **ssh** into the appliance. If a message indicates that a reboot is necessary, reboot the virtual machine to complete the operating system upgrades.

10. **Version 2018.4.1.9 iFix1.0 and later:** After completion of the upgrade, verify that all tasks are running.

Due to a known limitation in versions prior to v2018.4.1.9 iFix1.0, some tasks may have stopped running. Complete the following steps:

- a. Download the **apicops** utility from <https://github.com/ibm-apiconnect/apicops/releases>.
- b. Run the following command to remove any pending tasks:

```
$ apicops task-queue:fix-stuck-tasks
```

- c. Run the following command to verify that the returned list (task queue) is empty.

```
$ apicops task-queue:list-stuck-tasks
```

11. If you upgraded the Analytics subsystem from Version 2018.4.1.9 (or earlier) to Version 2018.4.1.10 (or later), and you enabled the Analytics message queue, you must clean and restart the Analytics message queue and ingestion pods.

With root permissions, run the following script on a single Analytics OVA post upgrade.

```
#sudo su
#!/bin/bash
kubectl get pods --no-headers -o custom-columns=:metadata.name | grep zookeeper | while read POD ; do
  echo "-----Deleting zookeeper data for $POD"
  kubectl exec $POD -- rm -rf /var/lib/zookeeper/data
  kubectl exec $POD -- rm -rf /var/lib/zookeeper/log
  kubectl exec $POD -- rm /var/lib/zookeeper/zookeeper.entries
  kubectl exec $POD -- rm /var/lib/zookeeper/nodes.json
done
kubectl get pods --no-headers -o custom-columns=:metadata.name | grep zookeeper | while read POD ; do
  echo "-----Deleting pod"
  kubectl delete pod $POD --grace-period 0 --force &
done
sleep 120
kubectl get pods --no-headers -o custom-columns=:metadata.name | grep kafka | while read POD ; do
  echo "-----Deleting pod"
  kubectl delete pod $POD --grace-period 0 --force &
done
sleep 120
kubectl get pods --no-headers -o custom-columns=:metadata.name | grep ingestion | while read POD ; do
  echo "-----Deleting pod"
  kubectl delete pod $POD --grace-period 0 --force &
done
```

When the script completes, verify that the zookeeper pods are running:

```
kubectl get pods
```

If any of the zookeeper pods are not running, run the script again.

## What to do next

If you encounter problems with the cassandra or calico pods after completing the API Connect upgrade, see [Troubleshooting the API Connect upgrade on VMware](#) for suggested resolutions.

When you have successfully upgraded all of the API Connect subsystems, upgrade your DataPower Gateway Service. See [Upgrading DataPower Gateway Service](#).

- [Control planes needed for upgrading to earlier fix packs](#)

If you want to upgrade to an earlier fix pack than the current release, review this guide to determine which control planes you need.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Control planes needed for upgrading to earlier fix packs

If you want to upgrade to an earlier fix pack than the current release, review this guide to determine which control planes you need.

### Control planes for upgrading to 2018.4.1.24

Version to upgrade from	Instructions for upgrading to v2018.4.1.24
v2018.4.1.20 and iFixes Fix packs 21, 22, and 23 were not released.	No control plane needed.
v2018.4.1.19 and iFixes v2018.4.1.18 was not released	Download: <ul style="list-style-type: none"><li>• <code>appliance-control-plane-1.23.x.tgz</code></li></ul>
v2018.4.1.17 and iFixes	Download: <ul style="list-style-type: none"><li>• <code>appliance-control-plane-1.23.x.tgz</code></li><li>• <code>appliance-control-plane-1.22.x.tgz</code></li><li>• <code>appliance-control-plane-1.21.x.tgz</code></li></ul>

Version to upgrade from	Instructions for upgrading to v2018.4.1.24
v2018.4.1.16 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>
v2018.4.1.15 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.12 and iFixes</li> <li>• v2018.4.1.11 and iFixes</li> <li>• v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.9 and iFixes</li> <li>• v2018.4.1.8 and iFixes</li> <li>• v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.6 and iFixes</li> <li>• v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.23.x.tgz</code></li> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> <li>• <code>appliance-control-plane-1.14.x.tgz</code></li> </ul>

## Control planes for upgrading to 2018.4.1.20

Version to upgrade from	Instructions for upgrading to v2018.4.1.20
v2018.4.1.19 and iFixes v2018.4.1.18 was not released	No control plane needed.
v2018.4.1.17 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> </ul>
v2018.4.1.16 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>

Version to upgrade from	Instructions for upgrading to v2018.4.1.20
v2018.4.1.15 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.12 and iFixes</li> <li>• v2018.4.1.11 and iFixes</li> <li>• v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.9 and iFixes</li> <li>• v2018.4.1.8 and iFixes</li> <li>• v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.6 and iFixes</li> <li>• v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.22.x.tgz</code></li> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> <li>• <code>appliance-control-plane-1.16.x.tgz</code></li> <li>• <code>appliance-control-plane-1.15.x.tgz</code></li> <li>• <code>appliance-control-plane-1.14.x.tgz</code></li> </ul>

## Control planes for upgrading to 2018.4.1.19

Note: Fix pack 18 was not released.

Version to upgrade from	Instructions for upgrading to 2018.4.1.19
v2018.4.1.17 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.21.x.tgz</code></li> </ul>
v2018.4.1.16 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.20.x.tgz</code></li> </ul>
v2018.4.1.15 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> </ul>
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> </ul>
<ul style="list-style-type: none"> <li>• v2018.4.1.12 and iFixes</li> <li>• v2018.4.1.11 and iFixes</li> <li>• v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>• <code>appliance-control-plane-1.19.x.tgz</code></li> <li>• <code>appliance-control-plane-1.18.x.tgz</code></li> <li>• <code>appliance-control-plane-1.17.x.tgz</code></li> </ul>

Version to upgrade from	Instructions for upgrading to 2018.4.1.19
<ul style="list-style-type: none"> <li>v2018.4.1.9 and iFixes</li> <li>v2018.4.1.8 and iFixes</li> <li>v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> <li>appliance-control-plane-1.16.x.tgz</li> <li>appliance-control-plane-1.15.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.6 and iFixes</li> <li>v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> <li>appliance-control-plane-1.16.x.tgz</li> <li>appliance-control-plane-1.15.x.tgz</li> <li>appliance-control-plane-1.14.x.tgz</li> </ul>

## Control planes for upgrading to 2018.4.1.17

Version to upgrade from	Instructions for upgrading to 2018.4.1.17
v2018.4.1.16 and iFixes	No Control Plane file needed.
v2018.4.1.15 and iFixes	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> </ul>
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.12 and iFixes</li> <li>v2018.4.1.11 and iFixes</li> <li>v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.9 and iFixes</li> <li>v2018.4.1.8 and iFixes</li> <li>v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> <li>appliance-control-plane-1.16.x.tgz</li> <li>appliance-control-plane-1.15.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.6 and iFixes</li> <li>v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.19.x.tgz</li> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> <li>appliance-control-plane-1.16.x.tgz</li> <li>appliance-control-plane-1.15.x.tgz</li> <li>appliance-control-plane-1.14.x.tgz</li> </ul>

## Control planes for upgrading to 2018.4.1.16

Version to upgrade from	Instructions for upgrading to v2018.4.1.16
v2018.4.1.15 and iFixes	No Control Plane file needed.
v2018.4.1.13 and iFixes	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.18.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.12 and iFixes</li> <li>v2018.4.1.11 and iFixes</li> <li>v2018.4.1.10 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> </ul>
<ul style="list-style-type: none"> <li>v2018.4.1.9 and iFixes</li> <li>v2018.4.1.8 and iFixes</li> <li>v2018.4.1.7 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li>appliance-control-plane-1.18.x.tgz</li> <li>appliance-control-plane-1.17.x.tgz</li> <li>appliance-control-plane-1.16.x.tgz</li> <li>appliance-control-plane-1.15.x.tgz</li> </ul>



Version to upgrade from	Instructions for upgrading to v2018.4.1.16
<ul style="list-style-type: none"> <li>v2018.4.1.6 and iFixes</li> <li>v2018.4.1.5 and iFixes</li> </ul>	Download: <ul style="list-style-type: none"> <li><code>appliance-control-plane-1.18.x.tgz</code></li> <li><code>appliance-control-plane-1.17.x.tgz</code></li> <li><code>appliance-control-plane-1.16.x.tgz</code></li> <li><code>appliance-control-plane-1.15.x.tgz</code></li> <li><code>appliance-control-plane-1.14.x.tgz</code></li> </ul>

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting the API Connect upgrade on VMware

Troubleshoot your API Connect upgrade on VMware.

Note: Do not install any extra software or run any commands on the VMs that are not documented in the IBM documentation or otherwise advised by IBM. When troubleshooting API Connect, do not use `kubectl exec` commands to access API Connect pods unless advised by IBM. Do not make any changes on the deployed VMs unless documented here or otherwise advised by IBM. Attempting to manually update packages, adding new users, or installing new software will likely cause problems. Operating system updates are handled by API Connect fix packs.

- [Restarting unhealthy cassandra pods](#)
- [Recreating a calico pod that is in the CrashLoopBackOff state](#)

---

### Restarting unhealthy cassandra pods

If you complete an upgrade and find that one or more of the cassandra pods (prefixed with `apiconnect-apiconnect-cc`) are not responding:

1. Run the following checks to verify that the pods require a restart:

- Check the status of the pods to see if they are marked as `Running` but not `Ready` as in:

```
apiconnect-apiconnect-cc-g4rhq 0/1 Running
```

- Check the `describe` of the pods for an event log with a warning similar to the following example:

```
Warning Unhealthy 2m6s (x177 over 38m) kubelet Readiness probe failed: Cassandra is either decommissioning or upgrading
```

2. Restart each pod by deleting it with the following command (which recreates the pod):

```
kubectl delete pod <pod_name>
```

Note: If there are multiple cassandra pods in this state, then delete them one at a time' wait until the previously deleted pod comes up and shows as `Running` in the `Ready` state before deleting the pod.

---

### Recreating a calico pod that is in the CrashLoopBackOff state

If you complete an upgrade and find that one or more of the calico pods are in the `CrashLoopBackOff` state, you can delete the pod to recreate it. Complete the following steps:

1. Get the names of the calico pods:

```
kubectl -n kube-system get pods | grep calico
```

2. Recreate each pod by deleting it with the following command:

```
kubectl -n kube-system delete pod <calico_pod_name>
```

- [Checking cluster health on VMware](#)  
You can use `apicup` to check the health of the API Connect clusters in your VMware deployment.
- [Determining status of a cluster on VMware](#)  
You can determine the health of an API Connect cluster on VMware
- [Obtaining simple health check data of Developer Portal sites by using a REST API call](#)  
Call a simple health check API from your external load balancer to dynamically determine whether a specific Developer Portal site in a cluster is working. This API call can be used by a load balancer to help determine where to route traffic.
- [Monitoring the network with SNMP](#)  
Presents a table of OID trees that you can poll using SNMP Get.
- [Gathering logs for a VMware environment](#)  
The `generate_postmortem.sh` script gathers all logs for troubleshooting and diagnostics.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Upgrading DataPower Gateway Service

Use these instructions when upgrading DataPower Gateway Service in a non-Kubernetes environment, during upgrade of API Connect Version 2018.4.1.0 or later.

## Before you begin

---

Ensure you have completed the upgrade of the API Connect subsystems (Management service, Developer Portal, Analytics), before upgrading DataPower Gateway, for the following reasons:

- Upgrading the Management service before the DataPower Gateway ensures that any new policies and capabilities will be available to a previously registered Gateway service.
- The API Connect Management server and DataPower Gateway firmware versions must match once upgrade is complete. For example, APIC 2018.4.1.2 with DataPower 2018.4.1.2.

To review API Connect upgrade on VMware, see [Upgrading API Connect subsystems in a VMware environment](#).

## About this task

---

- The DataPower Gateway Service in a non-Kubernetes environment can be on either a physical appliance or in a virtual DataPower deployment.
- The upgrade of DataPower Gateway Service in a non-Kubernetes environment, for use in API Connect deployments in a VMware environment, does not use the API Connect Install Assist program (**apicup**).

Note: To upgrade the DataPower Gateway Service in a Kubernetes environment, do not use the following procedure. Instead, see [Upgrading API Connect in a Kubernetes environment](#).

## Procedure

---

1. When upgrading a high-availability cluster, ensure that you meet the requirements:

- Gateways must be updated one at a time.
- Before starting the upgrade, a single gateway must be running as primary for all gateway-peering definitions.
- When upgrading multiple gateways, the primary gateway must be upgraded last.

To determine which gateway is running as primary, use either the **show gateway-peering-status** command in the DataPower CLI, or use the Gateway Peering Status display in the WebGUI in the API Connect application domain. To move the primary to the DataPower on which you're currently working, you can issue the **gateway-peering-switch-primary <peering-object-name>** command.

2. Follow the upgrade instructions on the DataPower Gateway documentation. See [Installation operations](#).

Note: If upgrading a cluster (high-availability) deployment, ensure that you have identified which gateway is running a primary in gateway-peering definitions, and that you upgrade that gateway last.

3. **Version 2018.4.1.9 and later:** All gateway extensions and gateway script policies which were in place prior to the upgrade to Version 2018.4.1.9 are automatically moved from **local://** to **temporary://**. Therefore, any references to **local** in those extensions or policies must be changed to refer to **temporary**.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Maintaining a VMware deployment

You can use utilities to complete maintenance tasks such as backup, restore, and certificate management in a VMware environment.

Note: Do not install any extra software or run any commands on the VMs that are not documented in the IBM documentation or otherwise advised by IBM. When maintaining API Connect, do not use **kubectl exec** commands to access API Connect pods unless advised by IBM.

Do not make any changes on the deployed VMs unless documented here or otherwise advised by IBM. Attempting to manually update packages, adding new users, or installing new software will likely cause problems. Operating system updates are handled by API Connect fix packs.

- [Backing up and restoring](#)  
You can backup and restore API Connect subsystems.
- [Disabling the Analytics subsystem on VMware](#)  
Disable the Analytics subsystem by shutting down the VM that hosts it.
- [Using VM snapshots for infrastructure backup and disaster recovery](#)  
You can use VM snapshots to backup and restore the infrastructure used by API Connect on VMware, and also for infrastructure disaster recovery.
- [Using APICUP to reconfigure](#)  
You can use APICUP to change configuration of a subsystem after completion of initial installation.
- [Setting rate limits for public APIs on the management service for a VMware environment](#)  
Describes the procedure for setting a rate limit for public APIs on the management service. Rate limits provide protection from DDoS (distributed denial of service) attacks.
- [Dynamically re-registering and reconfiguring a Gateway service in a VMware deployment](#)  
In API Connect, Gateway services do not persist their configuration settings by default. Instead, the master configuration is stored on the Management server and the *Dynamic Reregistration and Reconfiguration* (DRR) mechanism resynchronizes configuration data when needed. The DRR process is used when proper High Availability/Disaster Recovery (HA/DR) is not configured, or if a manual resynchronization is required.
- [Adding disk space to a VMware appliance](#)  
Increase disk space on the VMware appliance.

- [Running a filesystem check on a VMware root partition](#)  
You can run a filesystem check on the appliance root filesystem at boot time.
- [Managing an appliance data disk](#)  
You can use `apic` commands to manage appliance data disks in your VMware deployment.
- [Troubleshooting the API Connect upgrade on VMware](#)  
Troubleshoot your API Connect upgrade on VMware.
- [Changing logging levels](#)  
You can enable logging for entry and exit trace and for large payloads for apim-v2 pods.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Backing up and restoring

You can backup and restore API Connect subsystems.

Note:

- Backups are intended for recovery of the Management, Portal, and Analytics subsystems onto the same deployment from which they were taken, or onto a new replacement installation in the same environment for disaster recovery. The same environment means the same network configuration and project directory as the original installation.
- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.
- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.
- The API Connect backup and restore procedures back up all of the state required by API Connect but do not include the underlying infrastructure software or other internal state. In some scenarios, you may want to perform backups and disaster recovery at an infrastructure layer. You can do this by taking snapshots of the virtual machines or underlying storage volumes. See [Using VM snapshots for infrastructure backup and disaster recovery](#).
- [Backing up the management subsystem in VMware environments](#)  
Backups of the management database can be performed based upon a cron-like schedule or on-demand from the command line.
- [Restoring a management subsystem in a VMware environment](#)  
The management database can be restored as a complete restoration. Partial restorations are not supported.
- [Backing up and restoring the Developer Portal in a VMware environment](#)  
How to backup and restore your Developer Portal service in your VMware environment.
- [Backing up and restoring the analytics database](#)  
The analytics database can be backed up and restored from an S3 repository. S3 compatible object storage is required, for example, IBM Cloud Object Storage.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Backing up the management subsystem in VMware environments

Backups of the management database can be performed based upon a cron-like schedule or on-demand from the command line.

### Before you begin

---

The backup and restore feature is enhanced for API Connect Version 2018.4.1.

### About this task

---

Database backups can be used to restore the database for disaster recovery or for transferring data during an upgrade.

- For scheduled backups, see [Configuring scheduled backups](#)
- For on-demand backups, see [Performing on-demand backups](#)

We strongly recommend that you configure a backup schedule for your management database using the `cassandra-backup-schedule` setting during installation of the management subsystem. If you did not do so when you installed API Connect in your VMware runtime environment, you have the option to perform an on-demand backup. We also recommend that you perform a backup (either scheduled or on-demand) of the management database prior to upgrading.

There are two options for performing backups of the management database: scheduled backups and on-demand backups. Both options require the `cassandra-backup-x` settings to be configured when installing the management subsystem. Automatic scheduled backups are performed according to the cron-like job configured by the `cassandra-backup-schedule` setting. On-demand backups also require the backup settings, but are run on-demand from the command line.

Note: You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.

## Procedure

---

• **Configuring scheduled backups**

1. To configure scheduled backups, enter the `cassandra-backup-x` settings when installing the management subsystem as described in [Deploying the Management subsystem in a VMware environment](#). The `cassandra-backup-x` parameters are also described in the following table:

The parameters are as follows:

Parameter	Description
<code>cassandra-backup-protocol</code>	The backup protocol. Specify one of the following values: <ul style="list-style-type: none"> <li>• <code>sftp</code> - for secure file transfer protocol</li> <li>• <code>objstore</code> - for S3 compatible object storage</li> </ul> Note: <ul style="list-style-type: none"> <li>• For the management subsystem, IBM Cloud and Amazon Web Services (AWS) are supported S3 object store providers.</li> <li>• The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code>.</li> </ul>
<code>cassandra-backup-path</code>	The full path to the directory where the backup files will be stored. This must point to a directory on the backup server. For object storage ( <code>objstore</code> ), the path can be set to the <code>bucket</code> value or the <code>bucket/subfolder</code> value.
<code>cassandra-backup-host</code>	The fully qualified domain name of the backup server. Ensure that the Kubernetes nodes can access this host. If using object store, enter <code>Endpoint/Region</code> . (The <code>/</code> character between the endpoint and region are required for this setting.)
<code>cassandra-backup-port</code>	The port for the protocol to connect to the <code>cassandra-backup-host</code> . The backup port is not required for object storage.
<code>cassandra-backup-schedule</code>	Cron like schedule for performing automatic backups. The format for the schedule is: <ul style="list-style-type: none"> <li>• <code>*****</code></li> <li>• <code>-----</code></li> <li>• <code>     </code></li> <li>• <code>     +-----</code> day of week (0 - 6) (Sunday=0)</li> <li>• <code>   +-----</code> month (1 - 12)</li> <li>• <code>  +-----</code> day of month (1 - 31)</li> <li>• <code> +-----</code> hour (0 - 23)</li> <li>• <code>+-----</code> min (0 - 59)</li> </ul> The backup schedule defaults to <code>0 0 * * *</code> . This means a backup is run every day at midnight and minute zero. The timezone for backups is UTC.  When you configure a host, if you do not specify a value for <code>cassandra-backup-schedule</code> , the default backup schedule is automatically set. Note that the default backup schedule is not set, and scheduled backups not enabled, until host configuration is completed.  Note: Cassandra <code>repair</code> cron schedule is set to <code>00 1 * * 0,2,4,6</code> . This means the repair runs at 01:00 on Sunday, Tuesday, Thursday, and Saturday. By default, the Cassandra <code>backup</code> cron schedule should not run within one hour of the repair cron schedule. Please make sure to modify the current backup configuration as needed. If backups and repairs run at the same time, backup processes can fail intermittently.
<code>cassandra-backup-auth-user</code>	The username for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Key ID.
<code>cassandra-backup-auth-pass</code>	The password for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Access Key parameter. The password will be stored in Base64 encoded format. For example: <pre>apicup subsys set mgmt cassandra-backup-auth-pass '&lt;password&gt;'</pre> Note that you cannot use the <code>`</code> sign to assign the password to <code>cassandra-backup-auth-pass</code> .

For example:

```
cassandra-backup-protocol: sftp
cassandra-backup-host: mybackuphost.com
cassandra-backup-port: 22
cassandra-backup-path: /backups
cassandra-backup-schedule: 0 0 * * *
cassandra-backup-auth-user: myusername
cassandra-backup-auth-pass: mypassword
```

2. These settings will be activated by the `apicup subsys install <SUBSYS_NAME>` command.
3. To verify that automatic backups are in progress, you will see a pod labeled `<cassandra cluster name>-backup` at the scheduled time for the backup.
4. When the scheduled back is complete, the backup files are stored at the location specified by the `cassandra-backup-path` parameter. The file name varies depending upon the API Connect version that performed the backup, but the file name must be compatible with the version used for restoring the database. For details about the file name, see [Restoring a management subsystem in a VMware environment](#).
5. List the backups by entering: `apicup <subsys exec <SUBSYS_NAME> list-backups`. You can view a list of backups with the ID and Status in the output. Following is an example:

Cluster	Namespace	ID	Timestamp	Status
rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC	Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC	Complete

• **On-demand backups**

1. Configure the backup parameters. (The same parameters are required for on-demand backup that are required for a scheduled backup, but you will bypass the schedule.) If you did not enter the `cassandra-backup-x` parameters when you installed API Connect, you will need to enter them from the command line and re-install the management subsystem before performing an on-demand backup.
2. After the backup parameters have been specified, run the `apicup subsys install <SUBSYS_NAME>` command to activate the new parameters. This step is not required if you specified these parameters during the initial installation.
3. Enter the following command: `apicup subsys exec <SUBSYS_NAME> backup` where `SUBSYS_NAME` is the name of your management subsystem.
4. When the backup is completed, the backup files are stored at the location specified by the `cassandra-backup-path` parameter. The file name varies depending upon the API Connect version that performed the backup, but the file name must be compatible with the version used for restoring the database. For details about the file name for the backups, see [Restoring a management subsystem in a VMware environment](#).
5. List the backups by entering: `apicup <subsys exec <SUBSYS_NAME> list-backups`. You can view a list of backups with the ID and Status in the output. Following is an example:

Cluster	Namespace	ID	Timestamp	Status
rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC	Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC	Complete

## Results

Use the ID generated for the backup files to restore the database from the backup. Usually the database is restored from a backup file after an upgrade is performed and to recover from a disaster. See [Restoring a management subsystem in a VMware environment](#) and [Upgrading API Connect subsystems in a VMware environment](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Restoring a management subsystem in a VMware environment

The management database can be restored as a complete restoration. Partial restorations are not supported.

### Before you begin

- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- Restoring the Management Service requires database downtime and is a destructive process that deletes current data and copies backup data. During the restoration process, external traffic must be stopped.
- In a Disaster Recovery scenario, do not log in to the administration UI or attempt to configure or change any settings prior to restoring the backup. Restore the backup immediately after installing the subsystem.
- To restore the management database, you must use the original project directory that was created with `apicup` during the initial product installation. You cannot restore the database without the initial project directory because it contains pertinent information about the cluster. The endpoints and certificates cannot change; the same endpoints and certificates will be used in the restored system. Successful restoration depends on use of a *single apicup* project for all subsystems, even those in a different cluster. Multiple projects will result in multiple certificate chains which will not match.

Note: Endpoints for the components cannot change between deployments. However, the endpoints for the VMware hosts can be modified for the new deployment.

- Map the DNS entries from the source cluster to the corresponding IP addresses on the target cluster. Record the DNS entries for each endpoint before starting the restore.
- When restoring the management database, the endpoints (on the new cluster which is the target for the restoration) have to be the same as those on the old cluster (the source of the backup). This includes all the endpoints for API Connect: `api-manager-ui`, `cloud-admin-ui`, `consumer-api`, `platform-api`; `api-gateway`, `api-gw-service`; `analytics-ingestion`, `analytics-client`; and `portal-admin`, `portal-www`.
- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.

### About this task

Follow the procedure on this page to restore your management database. You must complete the prerequisite steps before beginning the restore.

Note that in a disaster recovery scenario you must first re-establish the management subsystem. The procedure includes an optional first step for disaster recovery.

If you encounter errors, see [Troubleshooting restoration of management database on VMware](#). Note that the troubleshooting page includes [Overview of restore process for management database](#).

### Procedure

1. If the restoration is for a disaster recovery scenario, complete this step to first install a new Management subsystem. If the restoration is *not* for a disaster recovery (meaning that you have a *running* Management subsystem to use for restoration), skip this step and go directly to Step 2.
  - a. Copy the project folder that corresponds to the backup files into a new location.
  - b. If, for the previous installation, you have redirected the configuration to an optional output folder using the `apicup subsys install mgmt --out=mgmt-out` command as explained in [Deploying the Management subsystem in a VMware environment](#), delete all output folders prior to starting the new installation.
  - c. Perform a fresh installation of the management subsystem using the following command:

```
apicup subsys install <SUB_SYS>
```

Important:

Do not make any other changes in the project directory nor run any other `apicup` operations before proceeding to the next step.

2. Verify that your deployment meets the prerequisites for restoring a management database:
  - a. You must restore onto a deployment that has the same number of OVA VM (nodes) as the deployment where the backup was created. For example, if the management service backup deployment had 3 OVA nodes, the restore deployment must have 3 OVA nodes. You cannot restore onto a deployment with fewer OVA nodes because the management service uses one Cassandra pod per node, and Cassandra data is sharded across the pods. To successfully restore, first create the matching number of OVA VM nodes. Note: Creation of multiple Cassandra pods on one management service OVA node is not supported.
  - b. Ensure that all Cassandra pods are on-line and running normally.
3. Restoration of a management subsystem requires access to backup tar files. Obtain your backup files, as follows:
  - a. Enter `apicup subsys exec <MANAGEMENT_SUBSYS_NAME> list-backups`. The output lists the current backups in your namespace with Backup ID and status.

Cluster	Namespace	ID	Timestamp
Status			

rf0c7310d07-apiconnect-cc	e2edemo	1537987501522014136	2018-09-26 18:45:01.522014136 +0000 UTC	Complete
rf0c7310d07-apiconnect-cc	e2edemo	1537920006787257385	2018-09-26 00:00:06.787257385 +0000 UTC	Complete

The backup files are stored at the location specified by the `cassandra-backup-path` parameter.

Examine the backup filename to identify the backup ID for use with the restore command.

Table 1. Backup file naming convention

Backup file name	BackupID (for use with restore command)
<backupId>-<pod index>-<# of pods>.tar.gz	[backupId]
Example:	Example:
1534954510365016356-0-3.tar.gz	1534954510365016356

Note: **Disaster Recovery:** If you are restoring on a new deployment, you will not have any backups. You will have to find the `backupID` by going to your backup host and looking for the backup files from previous deployments.

b. Confirm that you have a backup file for each Cassandra pod in your cluster. If you have  $n$  pods, you must have the same number of backup files.

c. **Optional but recommended:** Verify the integrity of the backup tar files.

Ensure that the tar files are not corrupt. For example, on Linux:

```
tar -tzf <backup_file>
```

d. **Optional but recommended:** Determine whether your environment has sufficient space to perform the restore. You need enough free space such that the size of the backup file, when multiplied by four, does not exceed 85% of the available free space.

For example, on Linux, you can use the following steps:

i. Exec a bash terminal on the Cassandra pod:

```
kubectl exec -it <Cassandra-pod> --bash
```

ii. Paste the following script to calculate space available for restore:

```
# current available space
avail=$(df /var/db/ | awk 'NR==2{print $4}')

#space occupied by /var/db/data/
db_data=$(du -s /var/db/data/ | awk '{print$1}')

# Estimated available space after cleanup (avail + db_data) and a buffer of 15%
total_space_avail=$((($avail + $db_data) * 1024) * 85 / 100)
echo $total_space_avail
```

The value obtained in the above script is in bytes and must be calculated for every pod, and compared against  $4x$  the value, where  $x$  is the backup tar size of the corresponding backup file.

Each backup file is in format `<backup-id>-<ordinal-of-cassandra pod>-<number-of-cassandra-pods in cluster>.tar.gz`

In the example script above, if the backup tar size of `<backup-id>-0-3.tar.gz` is  $15*1024*1024$  bytes, the value for `$total_space_avail` in Cassandra cc-0 pod must be around  $60*1024*1024$  bytes.

If free space is insufficient, create additional free space before starting the restore.

4. Restore the management database by entering:

```
apicup subsys exec <MANAGEMENT_SUBSYS_NAME> restore <backupID>
```

The `restore` command will restore all backups from files with the same backupID. You must have the same number of management database (Cassandra) pods running as the number of backup files that match the backupID.

5. Verify that the restore process completed successfully.

Ensure that the restore job is marked as completed. Note, however, that it is possible for the restore job to be marked complete, but the Cassandra restore is not complete.

The best way to ensure that the Cassandra restore is complete is to review the `CassandraRestoreStatus` field in the `CassandraClusters` Custom Resource. When the `CassandraRestoreStatus` is `completed`, the Cassandra database is successfully restored.

Example command flow to verify the restore:

a. Examine the restore job and pod:

```
# kubectl get jobs | grep restore
restore-plnfs 0/1 71s
```

```
# kubectl get pods | grep restore
restore-plnfs-hr2rh 0/2 Init:0/2 0 54s
```

b. Since the status of the restore pod is in `Init:0/2`, the first init container (Container restore) is being executed. In this case, watch the `ClusterRestoreStatus` inside `CassandraCluster` (cc) Custom Resource to see the current status of the Cassandra restore process.

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Running Retrieve checks on backup file 1583268283534609276-1-3.tar.gz for pod rdd94fb4a21-apiconnect-cc-1
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Running Retrieve checks on backup file 1583268283534609276-2-3.tar.gz for pod rdd94fb4a21-apiconnect-cc-2
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: Restore prelim checks passed for rdd94fb4a21-apiconnect-cc-2
```

c. When the restore process is complete, examine the restore pod and job status. Make sure `ClusterRestoreStatus` is marked as `completed`.

```
kubectl get jobs | grep restore
restore-plnfs 1/1 10m 7h17m
```

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: completed
```

If you encounter errors, see [Troubleshooting restoration of management database](#).

6. **Version 2018.4.1.9 iFix1.0 and later:** After completion of the restore, verify that all tasks are running. Complete the following steps:

- Download the `apicops` utility from <https://github.com/ibm-apiconnect/apicops/releases>.
- Run the following command to remove any pending tasks:

```
$ apicops task-queue:fix-stuck-tasks
```

- Run the following command to verify that the returned list (task queue) is empty.

```
$ apicops task-queue:list-stuck-tasks
```

- [Troubleshooting restoration of management database on VMware](#)

You can troubleshoot problems with restoring the management database.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting restoration of management database on VMware

You can troubleshoot problems with restoring the management database.

Review the information on this page to understand the steps you can take to troubleshoot a failed restore. Be sure to first read the [Overview of restore process for management database](#) to understand the logging and error reporting.

Note that for Version 2018.4.1.10 there is a known limitation with error reporting. See [Known limitation: failed restores not reporting properly](#).

- [Overview of restore process for management database](#)
- [Error: invalid backup host credentials](#)
- [Error: insufficient disk space](#)
- [Error: failure on preliminary backup checks on cc-1 and cc-2](#)
- [Known limitation: failed restores not reporting properly](#)
- [Frequently asked questions](#)

If you cannot resolve a failed restore, contact IBM Support for assistance.

## Overview of restore process for management database

The API Connect restore process is started by the command `apicup subsys exec <management_subsys> restore <backupID>`. The apicup installer starts a Kubernetes job named `restore-<id>`, which in turn starts a pod named `restore-<job-id>-<pod-id>`.

API Connect restore process

The management restoration pod consists of:

- `cassandra-restore --` The main responsibility of this init container is to perform the Cassandra database restore. If this container encounters an error, the restoration job is Failed with pod status of `InitError (0/2)`.  
Note: For Version 2018.4.1.10 and earlier, this container is called `job-container`.
- `Cassandra-health-check` - An init container that verifies that the Cassandra health status is in a healthy state. .
- `Lur-upgrade-job` - This container checks for schema mismatches between restored data and the currently running installation. If necessary, the container performs a schema upgrade.
- `Apim-upgrade-job` - This container checks for schema mismatches between restored data and the currently running installation. If necessary, the container performs a schema upgrade. The container also resyncs all the gateway services.

The init containers start sequentially. The second init container starts only once the first init container has succeeded. Containers inside the pod start immediately after all the init containers in the pod are started. For more info, see the Kubernetes documentation: [Understanding pod status](#)

Cassandra restore process

The Cassandra restore process is performed by the init container `<cassandra-restore>`. Backups are required from each Cassandra pod. The container process flow is:

1. Perform preliminary retrieval checks.
2. Download the backup tar file onto the Cassandra container.
3. Verify the backup tar file integrity.
4. Stop the Cassandra process.
5. Perform the Cassandra restore.
6. Start the Cassandra process with restored data.

If any of the preliminary checks fail, error messages are returned. The error conditions must be fixed, and then the restore process can be run again.

Note that API Connect Version 2018.4.1.10 (and later) performs extensive preliminary checks prior to beginning a restore. Running the checks extends the time required to complete a restore. In particular, restoration of large backup files (larger than 5 GB) take longer.

Logging

Both init container and container logs are available during the restore process. To obtain logs from a restore pod:

```
kubectl logs <restore-pod> -n <namespace> -c <init container/container name>
```

The `apicup` command produces logs from all init and main containers sequentially as soon they finish, either successfully or with errors.

#### Cassandra restore logging

The logs for the init container are updated only upon completion (success or failure). To get accurate status information for the current state of the Cassandra restore process, review the `ClusterRestoreStatus` field in the `CassandraClusters` Custom Resource:

```
kubectl get cc -n <namespace> -o yaml | grep -A 2 ClusterRestoreStatus
```

## Error: invalid backup host credentials

Example output when this problem occurs:

```
./apicup subsys exec mgmt restore 158326828353460927
```

```
Cluster      Namespace      Backup Name      Backup Retrieval Timeout(hrs)      Status
rdd94fb4a21-apiconnect-cc  niharns1      1583268283534609276      24      Started
```

Restore failed

```
Error: rpc error: code = Unknown desc =
Pod name: rdd94fb4a21-apiconnect-cc-0
```

Error:

```
ssh: Could not resolve hostname 9.30.251.186.xxx: Name or service not known
Couldn't read packet: Connection reset by peer
**** [ Wed Mar  4 04:04:25 UTC 2020 ]prelimRetrieve 0: Preliminary Retrieve checks      FAILED ****
[ Wed Mar  4 04:04:25 UTC 2020 ] Preliminary retrieve checks failed on all 1 attempts.      ABORTING restore
```

```
Error: unable to get log stream for container cassandra-health-status, pod restore-hhrsc
-2slbm, job restore-hhrsc: container "cassandra-health-status" in pod "restore-hhrsc
-2slbm" is waiting to start: PodInitializing
```

ClusterRestoreStatus in Cassandra Clusters CR:

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
```

```
ClusterRestoreStatus: 'Restore Failed: Retrieve preliminary checks failed for rdd94fb4a21-apiconnect-cc-0'
```

```
kubectl get pods | grep restore
```

```
restore-hhrsc-2slbm      0/2      Init:Error      0      26m
```

```
kubectl get jobs | grep restore
```

```
restore-hhrsc      0/1      27m      27m
```

## Error: insufficient disk space

**Issue:** cc-0 prelim checks can reject restore process complaining about insufficient disk space.

**Workaround:** Increase the disk space (4X size of Cassandra backup tar file size) allocated to all Cassandra nodes on a fresh install and re-attempt restore. See [Freespace check](#).

## Error: failure on preliminary backup checks on cc-1 and cc-2

**Issue:** Failure on preliminary backup checks on cc-1 and cc-2 can leave the system in non-ready state

**Workaround:**

1. Always look at `ClusterRestoreStatus` in `CassandraClusters` Custom Resource for failed or completed Cassandra restore status, regardless of restore job status. In this case, the restore job is stuck on health status check, and Cassandra cc-0 will be in a non-ready state.
2. Execute the following command on each Cassandra pod sequentially starting with cc-0. Wait for the Cassandra pod to become ready (1/1) before executing on other Cassandra pods. If the Cassandra pod is already in ready state (1/1), you do not need to wait, just run the command.

```
kubectl exec -it <cassandra-pod-X> -n <namespace>
-- sh -c 'rm -rf /var/db/.restore && rm -rf /var/db/restore/*'
```

In the above command the X in `<cassandra-pod-X>` stands for the numerical value (ordinal) of the Cassandra pod, such as (0,1,2).

3. Review the Cassandra operator logs to figure out why restore has failed and fix the problem.  
To see an example of this type of failure, review [Example 2: Corrupted backup tar file cc-1 not reporting properly](#).

## Known limitation: failed restores not reporting properly

**Issue:** Failure on preliminary backup checks on cc-0, such as a corrupt tar file or incomplete download of a backup tar file, can cause the restore job to complete, but the underlying `ClusterRestoreStatus` is marked as `Restore Failed`:  
<Reason>.

**Workaround:** Always check `ClusterRestoreStatus` in the `CassandraClusters` custom resource for failed or completed Cassandra restore status, regardless of restore job status. Consult the Cassandra operator logs to figure out why restore failed, and fix the problem.

See:

- [Example 1: Corrupted backup tar file cc-0 not reporting properly](#)
- [Example 2: Corrupted backup tar file cc-1 not reporting properly](#)

Example 1: Corrupted backup tar file cc-0 not reporting properly

In this example, restoration was started with `apicup`:



```
./apicup subsys exec restore 1583268283534609276
```

1. View the initial status of restore job and restore pod:

```
kubectl get jobs | grep restore
restore-xs85r          0/1          38s          38s

kn get pods | grep restore
restore-xs85r-st554    0/2          Init:1/2     0            92s
```

2. Note that the restore completes, according to the job and pod status:

```
kubectl get pods | grep restore
restore-xs85r-st554    0/2          Completed    0            3m43s

kubectl get jobs | grep restore
restore-xs85r          1/1          2m51s        4m1s
```

3. Observe, however, that the `apicup` restore command output includes the following failure status:

```
./apicup subsys exec mgmt restore 1583268283534609276
Cluster      Namespace   Backup Name      Backup Retrieval Timeout (hrs)  Status
rdd94fb4a21-apiconnect-cc niharns1    1583268283534609276 24                               Started

Cluster      Namespace   Backup Name      Status
rdd94fb4a21-apiconnect-cc niharns1    1583268283534609276 Restore Failed: Backup retrieve checks failed
```

4. Check the value of `ClusterRestoreStatus` in the Custom Resource. Note that `ClusterRestoreStatus` is marked as `Restore Failed`.

```
kubectl get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: 'Restore Failed: Backup retrieve checks failed'
```

5. Check the Cassandra operator pod logs and see errors:

```
time="2020-03-04T04:30:33Z" level=error msg="Restore Failed: Backup retrieve checks failed with an error: \n
Pod name: rdd94fb4a21-apiconnect-cc-0\n
Error: \n
gzip: stdin: not in gzip format\n
tar: Child returned status 1\n
tar: Error is not recoverable: exiting now\n
retrieveCheck 0: RetrieveCheck of backup file 1583268283534609276-0-3.tar.gz FAILED\n
**** [ Wed Mar 4 04:30:32 UTC 2020 ] retrieveCheck 0: Retrieve process END ****\n\n"
time="2020-03-04T04:30:33Z" level=info msg="Updating status to Restore Failed: Backup retrieve checks failed"
```

**Explanation:** Even though restore job says it completed, Cassandra restore never completed due to corrupt backup tar file. The restore job believes it reached completion, but in this case the error logging is incorrect. Therefore, you must check `ClusterRestoreStatus` in `CassandraCluster` Custom Resource to see if restore really completed. Note that `CassandraClusterRestore` status does not state exactly why restore failed, hence you must look at the Cassandra operator pod logs.

Table 1. Limitation with error reporting when Cassandra tar file for cc-0 is corrupted

Expected flow	Actual flow
1. Cassandra operator detects that cc-0 backup is corrupt	1. Cassandra operator detects that cc-0 backup is corrupt
2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code>	2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code>
3. Operator passes an error message back to restore init container (cassandra-restore).	3. Operator does <i>not</i> pass an error message back to restore init container (cassandra-restore)
4. <code>apicup</code> restore command line should exit with an error message on why restore init container (cassandra-restore) failed. Restore pod status should be <code>Init:Error</code> .	4. Restore pod moves forward and executes other init container and upgrade containers.
5. Restore job is marked as Failed	5. Restore pod is marked as completed

6. **Workaround:** The Cassandra cluster remains up and running as restore process failed internally, so you must review the Cassandra operator logs to figure out why exactly Cassandra restore failed. In this case, Cassandra restore failed due to corrupt backup tar file. Locate and use a non-corrupted Cassandra backup.

Example 2: Corrupted backup tar file cc-1 not reporting properly

In this example, restoration was started with `apicup`:

```
./apicup subsys exec restore 1583268283534609276
```

1. View the initial status of restore job and restore pod

```
kubectl get jobs | grep restore
restore-v7nvt 0/1 81s 81s

kubectl get pods | grep restore
restore-v7nvt-2jdzq 0/2 Init:0/2 0 101s
```

2. Continue to view pods, and observe that the restore pod is stuck.

The restore pod first init container (cassandra-restore), which performs the Cassandra restore, is marked as complete. The second init container is stuck waiting for the Cassandra cluster to become healthy.

```
kn get pods | grep restore
restore-v7nvt-2jdzq 0/2 Init:1/2 0 4m8s
```

The status `Init:1/2` means that first init container completed, but the process is waiting on second init container to finish.

3. Note that the output from the `apicup` restore command gives a status of `Restore Failed`:

```
./apicup subsys exec mgmt restore 1583268283534609276
Cluster      Namespace   Backup Name      Backup Retrieval Timeout (hrs)  Status
```

rdd94fb4a21-apiconnect-cc	niharns1	1583268283534609276	24	Started
Cluster	Namespace	Backup Name	Status	
rdd94fb4a21-apiconnect-cc	niharns1	1583268283534609276	Restore Failed: Backup retrieve checks failed	

4. View the `ClusterRestoreStatus` in the Cassandra Cluster Custom Resource:

```
kn get cc -o yaml | grep -A 1 ClusterRestoreStatus
ClusterRestoreStatus: 'Restore Failed: Backup retrieve checks failed'
```

5. View the Cassandra operator pod logs:

```
time="2020-03-04T05:04:18Z" level=error msg="Restore Failed: Backup retrieve checks failed with an error: \n
Pod name: rdd94fb4a21-apiconnect-cc-1\n
Error: \ngzip: stdin: not in gzip format\n
tar: Child returned status 1\n
tar: Error is not recoverable: exiting now\n
retrieveCheck 0: RetrieveCheck of backup file 1583268283534609276-1-3.tar.gz FAILED\n
**** [ Wed Mar  4 05:04:18 UTC 2020 ] retrieveCheck 0: Retrieve process END ****\n\n"
time="2020-03-04T05:04:18Z" level=info msg="Updating status to Restore Failed: Backup retrieve checks failed"
```

6. View the Cassandra pod status:

rdd94fb4a21-apiconnect-cc-0	0/1	Running	3	11h
rdd94fb4a21-apiconnect-cc-1	1/1	Running	2	11h
rdd94fb4a21-apiconnect-cc-2	1/1	Running	2	11h

**Explanation:** In this scenario, the restore process is stuck waiting for the Cassandra cluster to become healthy. Cassandra pod cc-0 is in a non-ready state. Due to a known limitation with error logging, the only way to accurately determine whether the restore is complete is to view `ClusterRestoreStatus` in `CassandraCluster` Custom Resource. Since `CassandraClusterRestore` status does not state exactly why the restore failed, you must look at Cassandra operator pod logs.

Table 2. Limitation with error reporting when corrupted Cassandra backup tar file cc-1 is corrupted

Expected flow	Actual flow
1. Cassandra operator detects that cc-1 backup is corrupt	1. Cassandra operator detects that cc-1 backup is corrupt
2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code> .	2. Operator updates <code>ClusterRestoreStatus</code> as <code>Restore Failed</code> .
3. Operator cleans up the restore process content and makes sure that Cassandra cluster is healthy using the existing data.	3. Operator <i>does not</i> perform any cleanup and <i>does not</i> makes sure that the Cassandra cluster is in a healthy state
4. Operator passes an error message back to restore init container (cassandra-restore) to indicate that the restore has failed.	4. Operator <i>does not</i> pass an error message back to restore init container (cassandra-restore) to indicate that the restore has failed.
5. The <code>apicup</code> restore command line should exit with a proper error message on why restore init container (cassandra-restore) failed. The restore pod status should be <code>Init:Error</code> . The restore job should be marked as Failed.	5. Restore pod moves forward from the init container but becomes stuck on second init container.

7. **Workaround:**

- a. Since the Cassandra cluster is in a degraded state, and the Cassandra pod cc-0 is in a non-ready state (0/1), run the following command to bring the Cassandra cluster up and running.

```
kubectl exec -it <cassandra-pod-X> -n <namespace> -- sh -c
'rm -rf /var/db/.restore && rm -rf /var/db/restore/*'
```

Note that you must run this command sequentially on all Cassandra pods, starting with cc-0. Make sure that the cc-x status is (1/1), before running this command on the next pod (cc-x+1).

- b. Review the Cassandra operator logs to determine why the Cassandra restore failed. In this scenario, since the Cassandra restore failed due to a corrupt backup tar file, the solution is to choose a non-corrupted Cassandra backup.

Note that this Cassandra restore issue applies not only specifically to a corrupted backup tar file, but also to any of the restore issues which may happen on Cassandra-x+1 pods (x is a numerical value starting with 0) and can leave previous Cassandra pods in a non-ready state.

## Frequently asked questions

Where can I find the status of the restore process?

See: [Cassandra restore logging](#)

In what cases can I re-run the restore process on existing installations?

The answer depends on which stage of the restore failed. If the restore process failed because of a corrupted tar file, you can re-initiate the restore process using a different backup ID.

In what cases I need to redeploy a whole new cluster before re-attempting a failed restore?

If the restore failed due to space allocation problems or any other reasons except corrupted backup tar, a complete new installation is needed. You must fix the problem reported by the restore process.

What action to take if I get the error message: Not enough space to download the tar file?

You must re-install the management subsystem, allocating sufficient disk space. Please look at system requirements section for recommended disk sizes.

What if the downloaded backup tar file is corrupted?

Use a different backup. Run backup tar integrity checks prior to attempting the restore.

If the backup tar integrity succeeds in your local system but restore process is failing, gather the required logs and contact IBM Support.

What if there is not enough space to perform restore using the downloaded backup tar file?

You must re-install the management subsystem, allocating sufficient disk space. See [Error: insufficient disk space](#).

What if the restore job finishes but I still don't see any restored data?

See [Known limitation: failed restores not reporting properly](#).

What if the restore process remains stuck on the health check status for a very long time?

This might be due to a known limitation. See [Example 2: Corrupted backup tar file cc-1 not reporting properly](#).

What if the restore process fails on Lur-upgrade-job and Apim-upgrade-job containers?

Run the restore process again. If the error persists, contact IBM Support.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Backing up and restoring the Developer Portal in a VMware environment

How to backup and restore your Developer Portal service in your VMware environment.

### About this task

It is strongly recommended that you configure the backup parameters for your portal service during installation. If you did not do so when you installed API Connect in your runtime environment, you must configure the backups for the Developer Portal before performing an upgrade. These backups can then be used to restore the Developer Portal if required. When your Developer Portal subsystem is running, you can also make on-demand backups by using the command line.

The default Developer Portal backup schedule is once every 24 hours, but the schedule can be changed in the backup settings. The Developer Portal saves all system and site backups locally, and also saves them remotely based on the configured SFTP and s3 settings.

The local backups are automatically maintained so that the latest three backups of each site and of the system are kept, and older backups are removed. This maintenance means that the Developer Portal retains the latest three backups for each site and for the system however old they are, but there is no deletion of the old backups on the remote server. If a site is deleted, then all of the local backups for that site are also deleted, as otherwise the backup volume might become full of old site backups. For remote backups, you can configure a retention policy on your remote server to remove the old backup files as required.

These instructions cover the following backup and restore actions:

- [How to configure the location and timing of your backups](#)
- [How to run on-demand backups](#)
- [How to restore a Developer Portal service](#)
- The backup secret is a Kubernetes secret that contains your username and password for your backup database (sftp/s3). Only password-based authentication is supported for sftp and s3, not authentication based on public certificates and private keys. Password-based authentication for s3 requires that you generate an access key and secret. For example:
  - IBM (Cloud Object Storage): [Service credentials](#).
  - AWS: [Managing access keys](#).

Note:

- The backup and restore procedures are the same for both Kubernetes and VMware environments.
- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.
- When restoring, the Gateway and all deployed subsystems (Management, Analytics, and Developer Portal) must be at the same version level.

### Procedure

- How to configure the location and timing of your backups
  1. Open your API Connect installation project directory.
  2. Run the following commands to set the location and timing of your backups:

```
apicup subsys set ptl site-backup-host mybackuphost.com
apicup subsys set ptl site-backup-port 22
apicup subsys set ptl site-backup-auth-user mybackupauthusername
apicup subsys set ptl site-backup-auth-pass mybackupauthpassword
apicup subsys set ptl site-backup-path /site-backups
apicup subsys set ptl site-backup-protocol sftp
apicup subsys set ptl site-backup-schedule "0 2 * * *"
```

The backup parameters are detailed in the following table.

Table 1. Portal backup parameters

Parameter	Description
site-backup-host	The fully qualified domain name of the backup server, in lowercase only. Ensure that the Kubernetes nodes can access this host. If using object storage, enter <i>Endpoint/Region</i> . (The / character between the endpoint and region is required for the object storage setting.)
site-backup-port	The port for the protocol to connect to the <i>site-backup-host</i> . Defaults to 22 if not explicitly set. The backup port is not required for object storage.
site-backup-auth-user	The user name for the server specified in <i>site-backup-host</i> . If using object storage, the user name is the S3 Secret Key ID.
site-backup-auth-pass	The password for the server specified in <i>site-backup-host</i> . If using object storage, the password is the S3 Secret Access Key parameter. The password is stored in Base64 encoded format, and must not be edited directly in the apiconnect-up.yml file.
site-backup-path	The full path to the directory where the backup files are stored. For object storage, the path can be set to the <i>bucket</i> value or the <i>bucket/subfolder</i> value.

Parameter	Description
<code>site-backup-protocol</code>	The protocol that is used to communicate with your remote backup endpoint. Specify one of the following values: <ul style="list-style-type: none"> <li><code>sftp</code> - for secure file transfer protocol</li> <li><code>objstore</code> - for S3 compatible object storage</li> </ul> The default protocol is <code>sftp</code> . Note: The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code> .
<code>site-backup-schedule</code>	The schedule for how often automatic Portal backups are run. The format for the schedule is any valid cron string, as follows: <pre> * * * * * - - - - -                   +----- day of week (0 - 6) (Sunday=0)       +----- month (1 - 12)     +----- day of month (1 - 31)   +----- hour (0 - 23) +----- min (0 - 59) </pre> For example: <code>30 22 * * 1</code> performs backups at 10:30 pm on Mondays. The default backup schedule is <code>0 2 * * *</code> (runs every day at 2 am). The timezone for backups is UTC.

3. Optional: At any time you can view the current Portal subsystem values by running the following command:

```
apicup subsys get pt1
```

where `pt1` is the name that you assigned to your Portal service. The output from this command lists all of the subsystem settings, including backup, and indicates whether there are any errors. You must fix any errors before continuing.

4. Activate the backup settings by running the following command:

```
apicup subsys install pt1
```

where `pt1` is the name that you assigned to your Portal service.

5. Validate the installation with the new backup parameters by running the following command:

```
apicup subsys get pt1 --validate
```

where `pt1` is the name that you assigned to your Portal service. The output from this command lists all of the subsystem settings, including backup, and indicates whether the values are valid or invalid. You must correct any invalid values before continuing.

- How to run on-demand backups

You can make on-demand backups of your Developer Portal system, sites, and complete service, by running the following `exec` commands in your API Connect installation project directory.

- To backup the Portal system configuration (and not the sites), run the following command:

```
apicup subsys exec pt1 backup-system
```

- To backup a specific Portal site or all sites, run the following command:

```
apicup subsys exec pt1 backup-site arg
```

where `arg` is a specific site UUID or URL, or to backup all sites use the argument `installed`.

- To backup the entire Portal service (the system and all installed sites), run the following command:

```
apicup subsys exec pt1 backup
```

Where `pt1` is the name that you assigned to your Portal service.

Note: To restore a Developer Portal service, you need backups of both the Portal system and the installed sites.

- How to restore a Developer Portal service

You can restore your Developer Portal service by using the backups that exist on your remote server, by running the following `exec` commands in your API Connect installation project directory. Note that before you can run these commands, you must have configured your remote backup server details, and have valid backups of both the Portal system and the installed sites (see sections [How to configure the location and timing of your backups](#) and [How to run on-demand backups](#)).

Note:

- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- Restoration requires a functioning Developer Portal. In a disaster recovery scenario, you might need to reinstall the Developer Portal subsystem before you can restore the backed-up data. To reinstall, refer to [Deploying the Developer Portal in a VMware environment](#).
- To see what Portal system and site backups that you currently have on your remote backup server, run the following command. (This command may be useful when performing some of the following commands.)

```
apicup subsys exec pt1 list-backups remote
```

- To see what restore actions are performed when the `apicup subsys exec pt1 restore-all` command is run, you can use the following command:

```
apicup subsys exec pt1 restore-all dry backup_from
```

where `backup_from` can be `now`, so the latest backup file that is available is used. Or you can specify a timestamp in the format of `YYYYMMDD.HHMMSS` to retrieve the nearest backup file to a specified time, searching backwards from the timestamp given.

Note: From IBM® API Connect Version 2018.4.1.17, the timestamp format changed to `YYYYMMDD.HHMMSS`. For Version 2018.4.1.16 and earlier, the timestamp format is `YYYY-MM-DD`

`HH:MM:SS`.

- To restore your Portal service, run the following command:

```
apicup subsys exec pt1 restore-all run backup_from
```

where `backup_from` can be `now`, so the latest backup file that is available is used. Or you can specify a timestamp in the format of `YYYYMMDD.HHMMSS` to retrieve the nearest backup file to a specified time, searching backwards from the timestamp given. This command executes the portal restore process,

which downloads the system backup and all of the site backups from the remote server, and installs them within the Portal stack. This process will then restore the system configuration from the found backup, and restore all of the sites. Note that if a backed up site is already installed on the current stack, then the site is reinstalled by using the backup (the site is overwritten by the backup). If there are multiple sites to restore, then these sites are queued for restoring. You can track the restoration process within the Portal **www admin** logs.

- o To view the list of installed and restoring sites, run the following command:

```
apicup subsys exec ptl list-sites sites
```

Note that any sites that are pending restore and still in the queue do not appear in this list.

- o To restore just the Portal system configuration, run the following command:

```
apicup subsys exec ptl restore-system arg
```

where **arg** can be the system backup **tgz** file name, to restore the system by using the specified backup file, or use the argument **latest** to restore the system by using the latest backup file on the remote server. Note that if Portal system configuration information exists on the current stack, running this command will overwrite that configuration.

- o To restore just the Portal sites, run the following command:

```
apicup subsys exec ptl restore-site arg
```

where **arg** can be the site backup **tgz** file name or URL, to restore a particular site by using the specified backup file (or the latest backup file found if using a URL). Or you can use the argument **all** to restore all of the Portal sites that have backup files on the remote server. Note that if any of the backed up sites are already installed on the current stack, then they are reinstalled by using the backup (the sites are overwritten by the backup).

## What to do next

---

For a full list of the Developer Portal **exec** commands, see [List of the Developer Portal exec commands](#).

- [List of the Developer Portal exec commands](#)

You can use the **exec** commands to backup, restore, and list some of the environment details of your Developer Portal service.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## List of the Developer Portal **exec** commands

You can use the **exec** commands to backup, restore, and list some of the environment details of your Developer Portal service.

The following Developer Portal **exec** commands can be run in your API Connect installation project directory, where **ptl** is the name of your Portal service.

Tip: Running **exec** commands leaves completed jobs and pods on your subsystem. Therefore, it is recommended that you regularly delete those jobs and pods that have completed successfully, and whose logs are not required.

```
$ apicup subsys exec ptl backup-system
Backup the portal system to the remote server
```

```
$ apicup subsys exec ptl backup-site <arg>
Backup a portal site to the remote server.
valid args:
  site uuid/url - will backup that site only to the remote server
  installed    - will backup all installed sites to the remote server
```

```
$ apicup subsys exec ptl backup
Backup the portal system and all its sites to the remote server
```

```
$ apicup subsys exec ptl list-backups <arg>
List the portal backups either locally or on the remote server.
valid args:
  local    - will list backups present on the pod's filesystem
  remote  - will list backups on the remote server
```

```
$ apicup subsys exec ptl list-platforms
List the platforms that exist on the portal
```

```
$ apicup subsys exec ptl list-sites <arg>
List the installed portal sites
valid args:
  sites      - list only the sites
  platforms  - list the sites and their associated platform
```

```
$ apicup subsys exec ptl restore-system <arg>
Restore the portal system from the remote server
valid args:
  system_backup-1234.tgz - restores the system from the given file which should exist on the pod's local filesystem or
the remote server
  latest                 - restores from latest system backup found from either local filesystem or remote server
```

```
$ apicup subsys exec ptl restore-site <arg>
Restore a portal site
valid args:
  site-1234.tgz - restore the site using this filename which should can be copied from the output of the list backups
command
  myportal.com  - find the latest backup for this URL (locally or remotely) and restore it.
```

```

    all - restore all sites found on the remote server
NOTES:
If you specify a site TGZ / URL, the command runs restore_site with -f so overwrites any existing site
If you specify 'all', any installed sites will be reinstalled from the found backup

$ apicup subsys exec ptl restore-all <arg1> <arg2>
Restore the portal system and all backed up sites from the remote server
valid arg1:
  dry - executes a dry-run of the command, returning the actions that will be performed
  run - executes the command, restoring and replacing the system files and all sites. Any existing sites will be
reinstalled
valid arg2:
  now - use the latest backups available
  <TIMESTAMP> - in 'YYYYMMDD.HHMMSS' format, specify a timestamp to retrieve the backup from. The nearest backup,
searching backwards from this timestamp, will be used.

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Backing up and restoring the analytics database

The analytics database can be backed up and restored from an S3 repository. S3 compatible object storage is required, for example, IBM Cloud Object Storage.

### Before you begin

If you configure Analytics for offloading and your endpoint requires a particular certificate, add trust certificates to the Analytics subsystem before attempting to back up or restore the Analytics database. For information on adding certificates to Analytics for back-up and restore, see [Adding certificates for Analytics for back-up and restore on VMware](#).

Tip: The [Procedure](#) section includes examples that illustrate how the back-up and restore commands work.

### About this task

These commands apply to OVA, pure Kubernetes, and IBM Cloud Private (ICP) deployments. To back up and restore the analytics database, you will need S3 compatible object storage. Backups are created on an as-needed basis and cannot be automated in API Connect.

**Important:** All arguments are required. Replace an argument with empty quotes (" ") to use the default setting.

Command	Values/Definition
<pre> apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; create-s3-repo </pre>	<p>Create the S3 repository to store analytics backups. These settings are identical to those used when creating a repository in Elasticsearch. The arguments are:</p> <ul style="list-style-type: none"> <li>REPO_NAME - Name of repository to be created.</li> <li>REGION - The region where bucket is located. Defaults to US Standard.</li> <li>BUCKET - The name of the bucket to be used for snapshots. Note: Analytics backup and restore supports virtual-host style bucket access, such as <code>bucket.s3-example.com</code>, but does not support path style bucket access such as <code>s3-example.com/bucket</code>.</li> <li>ENDPOINT - The endpoint to the S3 API.</li> <li>ACCESS_KEY - The access key to use for authentication.</li> <li>SECRET_KEY - The secret key to use for authentication.</li> <li>BASEPATH - The path name within the bucket where backup information is stored. The default is the root path. For S3 storage, you must include the port using the following format: <code>BASEPATH:PORT</code>.</li> <li>COMPRESS_TRUE_FALSE - Default is true. Determines whether metadata files are stored in compressed format.</li> <li>CHUNK_SIZE_GB - Default is 1GB. Large files can be stored as chunks when the snapshot is created. This setting specifies the size of the chunks as GB, MB, or KB.</li> <li>SERVER_SIDE_ENCRYPTION_TRUE_FALSE - Default is false. Determines whether files are encrypted. When set to true, files are encrypted on the server side using AES256.</li> </ul> <p>If the <code>create-s3-repo</code> command results in the following error, complete the steps in <a href="#">Adding certificates for Analytics for back-up and restore</a> to add the certificate to the Analytics subsystem and then run the command again:</p> <pre> PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target </pre>
<pre> apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; list-repos </pre>	<p>List repositories for analytics backups. There are no arguments or defaults for the <code>list-repos</code> command.</p>
<pre> apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; delete-repo </pre>	<p>Delete specified analytics backup repository. The argument is:</p> <ul style="list-style-type: none"> <li>REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> </ul> <p>Examples:</p> <ul style="list-style-type: none"> <li>If only one repo exists: <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; delete-repo ""</code></li> <li>If multiple repos exist: <code>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; delete-repo REPO_NAME</code></li> </ul>

Command	Values/Definition
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; backup</pre>	<p>Perform a backup of analytics data on an as-needed basis.</p> <ul style="list-style-type: none"> <li>INDICES - Default is <b>all</b> - All indices will be backed up. The INDICES arguments can consist of a series of index names, a single argument of comma-separated indices, or one or more keywords. Multiple keywords must be comma-separated. Values with whitespace must be enclosed in double-quotes. If no indices are specified, then all indices will be backed up. The keywords are mapped to indices or aliases in Elasticsearch. The keywords are as follows: <ul style="list-style-type: none"> <li><b>all</b> - Backup all data.</li> <li><b>apievents</b> - Backup all analytics data.</li> <li><b>ui</b> - Backup all UI visualizations and dashboards.</li> <li><b>config</b> - Backup all configuration information, such as retention period.</li> </ul> </li> <li>BACKUP_NAME - Enter the name of the backup.</li> <li>REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> <li>IGNORE_UNAVAILABLE_TRUE_FALSE - Default is false. If false, the restore will fail if an index is missing. If true, missing indices will be skipped and the restore will continue.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; list-backups</pre>	<p>List analytics backups per S3 repo.</p> <p>The argument is:</p> <ul style="list-style-type: none"> <li>REPO_NAME - Defaults to the existing repo if there is only one. The repository must be specified if there is more than one.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; details-backup</pre>	<p>Get details on specified analytics backup</p> <p>The arguments are:</p> <ul style="list-style-type: none"> <li>BACKUP_NAME - Defaults to <b>backup-<i>&lt;indices&gt;-<i>&lt;time date&gt;</i></i></b> if not explicitly set. The name must be all lowercase. For example, if the backup command was run as <b>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; backup ui apievents</b> then the backup name would be <b>backup-ui-apievents-2018-10-10t16:04:25z</b>. If more than three indices are specified, the backup name is truncated to <b>backup-<i>&lt;time date&gt;</i></b>.</li> <li>REPO_NAME - Sets the name of the repository where the backups will be stored. Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; delete-backup</pre>	<p>Delete specified analytics backup</p> <p>The arguments are:</p> <ul style="list-style-type: none"> <li>BACKUP_NAME - Enter the name of the backup to be deleted. You can view the names of backups using <b>apicup subsys exec &lt;ANALYTICS_SUBSYS&gt; list-backups</b>.</li> <li>REPO_NAME - Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; restore</pre>	<p>Restore analytics data</p> <ul style="list-style-type: none"> <li>INDICES - Default is <b>all</b> - All indices will be restored. The INDICES arguments can consist of a series of index names, a single argument of comma-separated indices, or one or more keywords. Multiple keywords must be comma-separated. Values with whitespace must be enclosed in double-quotes. If no indices are specified, then all indices will be backed up. The keywords are mapped to indices or aliases in Elasticsearch. The keywords are as follows: <ul style="list-style-type: none"> <li><b>all</b> - Restore all data.</li> <li><b>apievents</b> - Restore all analytics data.</li> <li><b>ui</b> - Restore all UI visualizations and dashboards.</li> <li><b>config</b> - Restore all configuration information, such as retention period.</li> </ul> </li> <li>BACKUP_NAME - Enter the name of the backup to be restored.</li> <li>REPO_NAME - Defaults to the existing repository if there is only one. The repository must be specified if there is more than one.</li> <li>IGNORE_UNAVAILABLE_TRUE_FALSE - Default is false. If false, the restore will fail if an index is missing. If true, missing indices will be skipped and the restore will continue.</li> <li>OVERRIDE_TRUE_FALSE - Default is false. If set to true, then the restore operation will override existing indices.</li> </ul>
<pre>apicup subsys exec &lt;ANALYTICS_SUBSYS &gt; restore-status</pre>	<p>Displays analytics status for determining the progress of the restore process.</p> <p>Note: The <b>restore-status</b> command is available in Version 2018.4.1.7 or later.</p>

Attention: Naming conventions for indices and back ups follow the Elasticsearch rules:

- Lowercase only
- Cannot include \, /, \*, ?, ", <, >, |, ` (space character), ,, #
- Colons (:) are not supported in 7.0+ (indices prior to 7.0 could contain a colon)
- Cannot start with -, \_, +
- Cannot be . or ..
- 255 byte limit (multi-byte characters will reach the 255 limit sooner)

## Procedure

- How to create a backup
  1. Create the S3 repository. The example creates a repository with the following values:
    - REPO\_NAME - myrepo
    - REGION - US
    - BUCKET - bucket
    - ENDPOINT - myrepo.s3repo.com
    - ACCESS\_KEY - access\_key
    - SECRET\_KEY - secret\_key

- BASEPATH - my\_folder
- COMPRESS\_TRUE\_FALSE - "" uses default of true
- CHUNK\_SIZE\_GB - "" uses default of 1GB.
- SERVER\_SIDE\_ENCRYPTION\_TRUE\_FALSE - "" sets to the default of false.

```
apicup subsys exec analytics create-s3-repo myrepo US bucket myrepo.s3repo.com access_key secret_key my_folder "" ""
```

OUTPUT:

Creating repository myrepo. List repos to see if creation completed, or check logs for errors.

Note:

With virtual host style repositories, the Analytics backup and restore process connects to the hostname formed by combining the values for **BUCKET** and **ENDPOINT**. For example, using the values given in this step, the host is `bucket.myrepo.s3repo.com`.

2. List the repositories. For example:

```
apicup subsys exec analytics list-repos
```

OUTPUT:

Name	Repo Type	Bucket	BasePath	Region	Endpoint	Chunk Size	Compress	Server Side Encryption
myrepo	s3	bucket	my_folder	US	myrepo.s3repo.com	1gb	true	

3. Create a backup with the following values:

- INDICES - all (or "" for the default of all)
- BACKUP\_NAME - mybackup
- REPO\_NAME - myrepo (use "" if only one repo exists)
- IGNORE\_UNAVAILABLE\_TRUE\_FALSE - "" sets to the default of false.

```
apicup subsys exec analytics backup all mybackup myrepo ""
```

OUTPUT:

Successfully created backup mybackup.

4. List backups in the repo named *myrepo*.

```
apicup subsys exec analytics list-backups myrepo
```

OUTPUT:

Name	Start Time	End Time	State
mybackup	2019-02-20T17:05:56.415Z	2019-02-20T17:06:09.833Z	SUCCESS

5. Display details for a backup named *mybackup* in the repo named *myrepo*.

```
apicup subsys exec analytics details-backup mybackup myrepo
```

OUTPUT:

```
Backup Name: mybackup
State: SUCCESS
Failures:
Shards Failed: 0
Shards Successful: 13
Start Time: 2019-02-20T17:05:56.415Z
End Time: 2019-02-20T17:06:09.833Z
Version: 5.6.8
Indices: .export-status, apic-api-2019.02.19-1, apic-api-2019.02.20-000002, .apic-config, .kibana-6
```

6. Delete a backup named *mybackup* in the repo named *myrepo*.

```
apicup subsys exec analytics delete-backup mybackup myrepo
```

OUTPUT:

Deleting backup mybackup. List backups to see the status, or check logs for errors.

- How to restore a backup

Note: Restoration requires a functioning Analytics subsystem. In a disaster recovery scenario, you might need to deploy Analytics before you can restore the backed-up data. To install and deploy, refer to [Deploying the Analytics subsystem in a VMware environment](#).

1. Restoring a backup

- INDICES - all (or "")
- BACKUP\_NAME - mybackup
- REPO\_NAME - myrepo (or "")
- IGNORE\_UNAVAILABLE\_TRUE\_FALSE - "" sets to the default of false.
- OVERRIDE\_TRUE\_FALSE - true

```
apicup subsys exec analytics restore all mybackup myrepo "" true
```

2. Check the status of the restore process. Enter the following command:

```
apicup subsys exec analytics restore-status
```

Note: The `restore-status` command is available in Version 2018.4.1.7 and later.

Following is example output:

Status	Active Primary Shards	Active Shards	Initializing Shards	Unassigned Shards
green	104	312	0	0

The restore process is successful when the status is green and there are no unassigned shards and no initializing shards. Note that the results are different if you are running in dev mode, or in standard mode with less than three nodes. For dev mode or standard mode with less than three nodes, the restore process will be finished when the initializing shards drops to 0. However, the status will remain yellow and unassigned shards will not drop to 0.

- Troubleshooting

Look for information in the `apicup` logs.

- Access the VM using `ssh`



- Locate the analytics-operator pod with: `sudo kubectl --kubeconfig /etc/kubernetes/admin.conf get pods`
- To view the logs, run: `sudo kubectl --kubeconfig /etc/kubernetes/admin.conf <analytics-operator-pod>`
- [Adding certificates for Analytics for back-up and restore on VMware](#)  
If you offload IBM API Connect Analytics data to an endpoint that is secured with a self-signed or private certificate, add the certificate to the Analytics subsystem to ensure connectivity with the endpoint during back-up and restore operations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Adding certificates for Analytics for back-up and restore on VMware

If you offload IBM® API Connect Analytics data to an endpoint that is secured with a self-signed or private certificate, add the certificate to the Analytics subsystem to ensure connectivity with the endpoint during back-up and restore operations.

### About this task

If you connect to an endpoint without using the required self-signed or private certificate, the `create-s3-repo` command results in the following error:

```
PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
```

Resolve the error by adding the endpoint's certificate to the Analytics subsystem as explained in the following steps.

### Procedure

1. Obtain trust certificates that allow a client to connect securely to the offload endpoint.  
The certificate files should be PEM encoded. There will typically be one or two trust certificates required to complete the secure connection.

2. Create a separate file for each of the trust certificates, naming them `cert1.pem` and `cert2.pem`.

3. Verify that you can connect to the endpoint:

- a. If you have multiple files, combine the certificates into a single file to use for validation by running the following command:

```
cat cert1.pem cert2.pem > certificates.pem
```

- b. Execute the following cURL command to connect to the endpoint using the certificates:

```
curl https://endpoint:port --cacert certificates.pem
```

A certificate verification failure indicates that the certificates obtained in step 1 failed to establish trust. Return to step 1 and try again. You cannot proceed to step 4 until the certificates are successfully verified.

- c. When the certificate verification is successful, proceed to step 4.

4. Create a Kubernetes secret to contain the certificates, and then apply it to the cluster where the Analytics subsystem runs.

- a. Log in to a server that has `kubectl` access to the Kubernetes cluster where API Connect Analytics is deployed.

For OVA deployments, you must log in to one of the running Analytics VMs as a user with `kubectl` access.

- b. Encode each of the certificate files from step 2 in base64.

There are several ways to encode the file. If `cat` and `base64` are available, then you can run the following command to encode each file:

```
cat cert1.pem | base64 -w 0
```

Copy the output for each `cat` command so that you can paste it in the next step.

- c. Create a Kubernetes secret and add the certificate.

- i. Create a YAML file to contain the secret.

```
apiVersion: v1
kind: Secret
metadata:
  # Change value of name to be whatever you want the secret to be called
  name: analytics-br-cert
data:
  cert1.pem: output of base64 encoded cert1.pem
  cert2.pem: output of base64 encoded cert2.pem if required
```

- ii. In the file's `data` section, create a field for each certificate, and paste the corresponding encoded certificate as shown in the example. The value is a single line, so you do not need to enclose it in quotation marks.

- iii. Save the file.

- d. Update the cluster with the new certificate.

Run the following command to update the cluster:

```
kubectl apply -f analytics-br-cert.yaml -n your_namespace
```

where:

- `analytics-br-cert.yaml` is the secret's file name.
- `your_namespace` is the name of your deployment's namespace. The namespace is only needed for Kubernetes deployments.

For OVA deployments, you can omit the entire `-n your_namespace` parameter from the command. By default, kubectl is already configured to match the namespace where your Analytics resources are deployed.

5. Update the Analytics subsystem so that it can use the new secrets.

- a. Create an `extra-values.yaml` file that specifies the names of your certificate and environment variables secret files.

Include the following lines in the `extra-values.yaml` file, making sure to use the name of your Kubernetes secret (the value of the `name` field in the `metadata` section of the secret, which might not match the file's name):

```
apic-analytics-storage:
  backupSecretCerts: analytics-br-cert
```

- b. Update the Analytics subsystem.

- i. Log in to the server where you run `apicup`, and navigate to the project directory.  
In Kubernetes deployments this is probably the same server that you used for step 1, but for OVA deployments it is a different server.
- ii. Run the following `apicup` commands to reinstall the Analytics subsystem using the configuration settings in the `extra-values.yaml` file.

```
apicup subsys set analytics extra-values-file=~/.extra-values.yaml
apicup subsys install analytics
```

6. Verify that the certificates were added in the storage logs.

Log in to the host node and use `sudo` to run the following command:

```
kubectl logs -n namespace analytics-storage-pod
```

For OVA deployments, you can omit the `namespace` parameter and value.

The following confirmation message displays when the pod starts up:

```
Adding /etc/velox/backup-certs/ca-analytics-backup.pem contents to the system keystore for backup and restore.
Certificate cert1.pem was added to keystore
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Disabling the Analytics subsystem on VMware

Disable the Analytics subsystem by shutting down the VM that hosts it.

---

### About this task

Disabling the Analytics subsystem stops the collection and storage of data as well as making the subsystem unavailable. When you disable Analytics, there will be no data in either the API Manager Dashboards or third-party offloads.

---

### Procedure

1. Back up your Analytics data as explained in [Backing up and restoring the analytics database](#).

2. Disable shard allocation for storage.

Disabling shard allocation is optional, but is recommended because it prevents new shards from being created for replication when a node is unavailable, and helps avoid corruption issues in full system restarts.

- a. Connect to the virtual machine as the API Connect administrator by completing the following steps:

- i. Run the following command to connect as the API Connect administrator, replacing `ip_address` with the appropriate IP address:

```
ssh ip_address -l apicadm
```

- ii. When prompted, select Yes to continue connecting.

- iii. When you are connected, run the following command to receive the necessary permissions for working directly on the appliance:

```
sudo -i
```

- b. Disable shard allocation for storage by running the following command:

```
kubectl -n namespace exec -it storage-master|shared_pod -- curl -s -XPUT -d '{"persistent": {"cluster.routing.allocation.enable": "none"}}'
```

where:

- `namespace` is the namespace where the Analytics subsystem is installed.
- `storage-master|shared_pod` is the name of any storage-master or storage-shared pod. You can get the name of your storage pods by running the following command and substituting in the namespace where Analytics is installed:

```
kubectl -n namespace get po
```

When you disable shard allocation, the response looks like the following example:

```
{"acknowledged": true, "persistent": {"cluster": {"routing": {"allocation": {"enable": "none"}}}}, "transient": {}}
```

Leave the connection open for the next step.

3. Perform a synced flush of storage.

Flushing the storage is optional, but is recommended because it helps to avoid losing the data that was not yet written to disk. This step flushes all current index operations synchronously and attempts to write everything that is in flux to disk. Perform a synced flush of storage by running the following command:

```
kubectl -n namespace exec -it storage-master|shared_pod -- curl -s -XPOST _flush/synced
```

When you flush storage, the response looks like the following example:

```
{ "_shards": {"total": 33, "successful": 13, "failed": 0}, ".export-status": {"total": 1, "successful": 1, "failed": 0}, ".apic-config": {"total": 1, "successful": 1, "failed": 0}, ".kibana-6": {"total": 1, "successful": 1, "failed": 0}, "apic-api-2020.06.19-000002": {"total": 15, "successful": 5, "failed": 0}, "apic-api-2020.06.18-1": {"total": 15, "successful": 5, "failed": 0}}
```

The operation often fails, and it is safe to rerun it multiple times until it passes and there are all "failed" counts are zero. If it continues to fail after running multiple attempts over the span of a couple minutes, you can proceed to the next step.

#### 4. Shut down the Analytics VMs.

You can shut down the VMs using one of the following methods:

- Use the VMware console
- `ssh` into the images and run the `shutdown` command

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using VM snapshots for infrastructure backup and disaster recovery

You can use VM snapshots to backup and restore the infrastructure used by API Connect on VMware, and also for infrastructure disaster recovery.

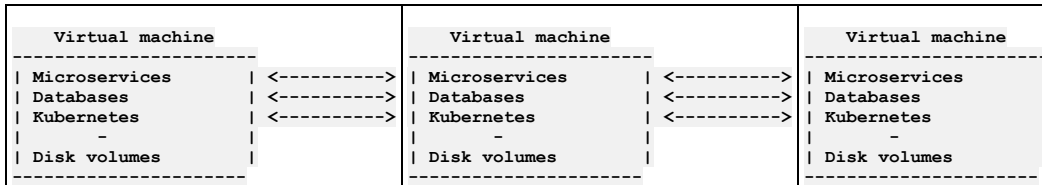
Backup and restore procedures for the databases used by IBM API Connect on VMware are described in [Backing up and restoring](#). The procedures back up all of the state required by API Connect but do not include the underlying infrastructure software or other internal state.

In some scenarios, you may want to perform backups and disaster recovery at an infrastructure layer. You can do this by taking snapshots of the virtual machines or underlying storage volumes, providing that you follow the constraints described here.

Note that performing backups at the infrastructure layer differs from the standard approach in that not only is the database state backed up, but also the precise state of the software. This approach can be useful, for example, when testing upgrades of production systems.

- Requirements for taking a consistent backup of an API connect system at the infrastructure level
  - An API Connect system can be approximated by the following diagram:

Table 1. API Connect Installation



- The system comprises multiple Virtual Machine images which are running Kubernetes, multiple databases, with the API Connect micro services running on top. There is transactional communication between the virtual machines which is occurring at all levels of this stack. This communication occurs all the time, even if the system is seemingly idle.
- Within the system there are multiple stateful or database layers on the software that maintain a consistency protocol. For example `etcd` upon which Kubernetes is based uses the [Raft](#) consensus algorithm. These algorithms depend for consistency on the basic assertion that time flows forwards on all systems together. If one or more of the virtual machine's state where to move backwards relative to the others by even the smallest quantum of time then consistency would be lost and the system might behave badly.

Clearly, if Virtual Machine snapshots are taken across multiple virtual machines it is certain that these snapshots will not be taken at precisely the same time. The result will be a backup that contains a system state that contains data from snapshots taken at slightly different times. It is possible that such a backup system may be restored and may appear to work correctly but hidden deep within it there may be undetected inconsistencies that will cause strange unpredictable errors and inconsistencies later. These issues can be extremely difficult to diagnose.

It is also possible that you can test this procedure successfully multiple times and see no apparent problem occurring. Irrespective of the apparent success of tests, corruption can be occurring in the system state. For this reason, taking VM snapshots or clones of a running API Connect system is not supported.

- How to take a consistent backup of an API connect system at the infrastructure level
  - The only way to take a consistent backup of an API connect system using VM snapshots and clones is to shut down ALL of the virtual machines comprising the system before taking the snapshot or clone.
  - Once the virtual machines are stopped at the VM level, time is effectively frozen. Then a snapshot can be taken on all the VMs. A clone can be made from the snapshot taken on each VM. This set of clones represent a valid, consistent state that the API Connect system was in when it was shut down.
  - Once all of the snapshots and clones have been taken the original API Connect system can be restarted.
- Restoring an API Connect System from VM clones
  - The Cloned VMs represent a valid state of the original API Connect system. They will restart in exactly the same way as the original system restarted.
  - If the objective is to stand up another instance of the cloned API connect system for testing or DR purposes, the following must be true:
    - The original system and the cloned system must be isolated from one another. The clone will be using the same hostnames and IP addresses as the original and so must be stood up in a separate virtual machine hosting environment with Network Address Translation between itself and any network to which the original system is stood up.
    - The cloned system must be stood up in an identical environment in terms of its hardware, software and network.
- Disaster Recovery approaches using Disk Replication:



You can use APICUP to change configuration of a subsystem after completion of initial installation.

The Install Assist utility (APICUP) is used to configure and complete initial installation of all subsystems, in both Kubernetes and VMware deployments. After initial installation, you can use the same `apicup` commands to reconfigure some of the settings from your existing deployment without having to completely reinstall the subsystem.

For example, if you did not configure optional features such as backup/restore or logging during initial installation, you can configure them later.

Note: Both Kubernetes and VMware deployments include subsystem settings that cannot be reconfigured without a complete deployment. For example, for the Management Service these settings include hosts, endpoints, interfaces (`public-iface`, `traffic-iface`), IP ranges of the Kubernetes pod and the service networks (`k8s-pod-network`, `k8s-service-network`), and backup volume size `cassandra-volume-size-gb`. Before reconfiguring, review the installation instructions for your subsystem to determine which settings are optional during initial installation and thus able to be updated later. See the section [Installing and upgrading on Kubernetes](#).

For both initial installation and reconfiguration, you specify configuration values by using `apicup subsys set` commands, and then activate them with `apicup subsys install`.

```
apicup subsys set <subsystem_name> <parameter_name>=<parameter_value>
apicup subsys install <subsystem_name>
```

For example, the following commands configure backup for the management subsystem `mgmt`, either during initial install or after initial install:

```
apicup subsys set mgmt cassandra-backup-auth-user=MyUsername cassandra-backup-auth-pass=MyPassword
apicup subsys set mgmt cassandra-backup-host=<hostname>
apicup subsys set mgmt cassandra-backup-path=</backups>
apicup subsys set mgmt cassandra-backup-port=<22>
apicup subsys set mgmt cassandra-backup-protocol=<sftp, objstore>
apicup subsys set mgmt cassandra-backup-schedule=<"0 0 * * *">
apicup subsys set mgmt cassandra-max-memory-gb=16 cassandra-cluster-size=3
apicup subsys set mgmt cassandra-volume-size-gb=<50>
apicup subsys set mgmt create-crd=<true>
apicup subsys set mgmt external-cassandra-host=<hostname>

apicup subsys install mgmt
```

On initial installation, the APICUP utility installs a Helm chart along with configuration values contained in `apiconnect-up.yml`. On subsequent invocations of `apicup subsys install`, APICUP checks to see if a saved combination of Helm chart and configuration values exists. If it does, APICUP checks to see if the configuration values have been updated, and if so, then a Helm update is triggered. If the combination of Helm chart and values is unchanged, then the subsystem is not changed. In this way, you can reconfigure as needed without having to completely redeploy and reinstall the subsystem.

- [Converting installation mode](#)  
You can switch an API Connect installation between `dev` mode and `standard` mode.
- [Increasing the memory allocation for the management database](#)  
You can increase the maximum memory allocation for the management database in a running deployment.
- [Enabling Analytics ingestion-only on VMware](#)  
Enable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` configuration setting.
- [Disabling Analytics ingestion-only on VMware](#)  
Disable the ingestion-only configuration by redeploying the IBM API Connect Analytics subsystem with the `ingestion-only` value set to false.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Converting installation mode

You can switch an API Connect installation between `dev` mode and `standard` mode.

### About this task

The supported installation mode options are: `dev` and `standard`. Mode is set for each subsystem type: Management, Analytics, Developer Portal, Gateway Service.

- Development mode (`dev`) deploys a subsystem with the scale of one; a single node, non-HA subsystem. The recommended use for `dev` mode is for development and testing.  
Note: While it is possible to use `dev` mode installations in a production deployment, it is not recommended. The `dev` mode does not provide the failover resiliency nor the downtime guarantee that is needed to maintain high availability (HA) production deployments.
- The `standard` mode deploys in HA mode for a production environment. The `standard` mode is supported for installation of production environments that consist of three or more nodes. The `standard` mode is not supported for less than three nodes.
- If not explicitly set, the installation mode defaults to `dev` (development). To specify `standard` mode:

```
apicup subsys set [subsys-name] mode=standard
```

Note: Prior to version 2018.4.1.4, the default mode was `standard`.

- If you installed in `standard` mode, but have less than three nodes, you can convert to `dev` mode. You must convert each subsystem.
- If you installed in `dev` mode, you can convert to `standard` mode. You must convert each subsystem.

Complete the following steps to determine your mode and, if necessary, convert the mode:

---

## Procedure

1. Review the output from the command: `apicup subsys get`

`<subsystem_name>`.

For example, `apicup subsys get mgmt`:

- VMware (OVA) environment output:

```
Appliance settings
=====
Name                Value                Description
-----
.
.
mode                standard
.
.
```

- Kubernetes output:

```
Kubernetes settings
=====
Name                Value                Errors
-----
.
.
mode                standard
.
.
```

2. If you need to convert the mode, following the appropriate instructions:

- [Converting dev mode to standard mode on VMware](#)
- [Converting dev mode to standard mode on Kubernetes](#)
- [Converting standard mode to dev mode on VMware](#)
- [Converting standard mode to dev mode on Kubernetes](#)
- [Converting dev mode to standard mode on VMware](#)  
You can convert your API Connect installation on VMware from `dev` mode to `standard` mode.
- [Converting standard mode to dev mode on VMware](#)  
You can convert your API Connect installation on VMware from `standard` mode to `dev` mode.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Converting dev mode to standard mode on VMware

You can convert your API Connect installation on VMware from `dev` mode to `standard` mode.

### About this task

To review the installation modes, see [Converting installation mode](#).

### Procedure

1. Use `apicup` to convert to `standard` mode.

```
apicup subsys set <subsys> mode=standard
```

2. Create hosts and interfaces for two more nodes, to bring the total number of nodes to three.

a. Create hosts:

```
apicup hosts create mgmt hostname.domainname hd_password
```

To review host creation for your subsystem type, see:

- Step [12](#) in [Deploying the Management subsystem in a VMware environment](#)
- Step [11](#) in [Deploying the Analytics subsystem in a VMware environment](#)
- Step [12](#) in [Deploying the Developer Portal in a VMware environment](#)

b. Create interfaces. For example:

```
apicup iface create mgmt hostname.domainname physical_network_id host_ip_address/subnet_mask gateway_ip_address
```

To review interface creation for your subsystem type, see

- Step [13](#) in [Deploying the Management subsystem in a VMware environment](#)
- Step [12](#) in [Deploying the Analytics subsystem in a VMware environment](#)
- Step [13](#) in [Deploying the Developer Portal in a VMware environment](#)

3. Install the new plan in a new `<plan-dir>`, which will generate 3 ISOs.

```
apicup subsys install <subsys> --out <plan-dir>
```

4. Configure the 3 virtual machines to attach their respective ISOs at boot.  
Follow the steps in the installation instructions for your subsystem type:

- [Deploying the Management subsystem in a VMware environment](#)
- [Deploying the Analytics subsystem in a VMware environment](#)
- [Deploying the Developer Portal in a VMware environment](#)

5. Restart the first node, which initially was a **dev** node.
6. Start the second node.
7. Start the third node.
8. To review your installation mode, use `apicup subsys get <subsystem_name>`.  
For example, `apicup subsys get mgmt`:

```
Appliance settings
=====
Name                               Value                               Description
----                               -
.
.
mode                                standard
.
.
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Converting standard mode to dev mode on VMware

You can convert your API Connect installation on VMware from **standard** mode to **dev** mode.

### About this task

---

To review the installation modes, see [Converting installation mode](#).

### Procedure

---

For each subsystem in your deployment, complete the following steps:

1. Use the following `apicup` command to convert to Dev mode:

```
# ./apicup subsys set [subsys-name] mode=dev
# ./apicup subsys install [subsys-name]
```

2. Remove all replica sets. Use `ssh` to access the VM, and use the following command:

```
sudo kubectl --kubeconfig /etc/kubernetes/admin.conf delete rs --all
```

3. To review your installation mode, use `apicup subsys get <subsystem_name>`.

For example, `apicup subsys get mgmt`:

```
Appliance settings
=====
Name                               Value                               Description
----                               -
.
.
mode                                dev
.
.
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Increasing the memory allocation for the management database

You can increase the maximum memory allocation for the management database in a running deployment.

### Before you begin

---

When increasing the memory allocation, complete a manual backup of the management database and Portal subsystem just prior to changing the allocation. Note that this backup is in addition to any scheduled backups that you have configured. The manual backup ensures that any data changes that occurred after the most recent scheduled backup are captured.

For details on creating a backup, see [Backing up the management subsystem in VMware environments](#) and [Backing up and restoring the Developer Portal in a VMware environment](#).

## About this task

Starting with API Connect version 2018.4.1.3, you can increase the maximum memory allocation for the management database in a running deployment. This procedure works for both Kubernetes and OVA environments. All the pods in the API Connect cluster must be up and running to increase the maximum memory value in a running deployment.

## Procedure

1. Ensure all pods are up and running using the `kubectl get pods` command.
2. Use the following `apicup` command to increase the maximum memory:

```
apicup subsys set <MANAGEMENT_SUBSYS_NAME> cassandra-max-memory-gb <new_value>
```

3. Reinstall the management subsystem to push the change to the cluster. This may take several minutes as each pod updates.

```
apicup subsys install <MANAGEMENT_SUBSYS_NAME>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Enabling Analytics ingestion-only on VMware

Enable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` configuration setting.

## Before you begin

Only complete this task if you want to offload all analytics data and disable the Analytics feature in the user interface. If you want to retain the data or allow users to access the Analytics user interface, skip this task and instead see [Configuring analytics offload for API Connect](#).

## About this task

If you choose to offload all analytics data to third-party services and have no need to retain the data in API Connect, you can configure the Analytics service with the ingestion-only setting. If you offload data and then separately disable the Analytics UI, the associated components are still deployed and require system resources. Configuring Analytics for ingestion-only removes unused Analytics components (such as analytics-storage and analytics-client) from the topology and only deploys the components required for offloading data (such as analytics-ingestion and analytics-mq-kafka). The reduced topology requires less CPU, memory, and storage.

## Procedure

1. Remove the Analytics service from your API Connect deployment by completing the following steps.
  - a. Unassociate the Analytics service from all gateway services as explained in step 7 in the topic, [Associating an analytics service with a gateway service](#).
  - b. Unregister the Analytics service from Cloud Manager.  
You can unregister the service from the Topology page: click the Options menu next to the service name and select Delete. For information on registering services, see [Registering an analytics service](#).
2. Create an extra values file to specify the URL of the offload endpoint where all analytics data will be routed.  
You can choose the name for the file, but you must use `.yaml` as the file-name extension. Format the file as shown in the following example, making sure to include required settings for the third-party system.

```
apic-analytics-ingestion:
  outputOffload: |-
    elasticsearch {
      hosts => "http://offload_endpoint:port"
      index => "api-call"
    }
```

For information about configuring analytics offloading, see [Configuring analytics offload for API Connect](#).

3. Run the following `apicup` commands to reinstall the Analytics subsystem using the configuration settings in the extra values file:

```
apicup subsys set analyt ingestion-only=true
apicup subsys set analyt extra-values-file=extra-values.yaml
apicup subsys install analyt
```

4. Add the new Analytics service to your API Connect deployment by completing the following steps.
  - a. Register the new Analytics service with the Cloud Manager.  
Attention: When you register an Analytics service that is configured for ingestion-only, you must use the ingestion endpoint and the ingestion TLS profile instead of the client endpoint and TLS profile.  
For information on registering a service with the Cloud Manager, see [Registering an analytics service](#).



- b. Associate the Analytics service with a gateway service as explained in [Associating an analytics service with a gateway service](#).

## Results

---

All analytics data is routed directly to the offloading endpoint, the Analytics UI is disabled, and no data is retained in API Connect.

In addition, the following Analytics configuration settings will not be validated or used when `ingestion-only` is set to enabled:

- `coordinating-max-memory-gb`
- `data-max-memory-gb`
- `data-storage-size-gb`
- `master-max-memory-gb`
- `master-storage-size-gb`
- `analytics-client`
- `es-storage-class`

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Disabling Analytics ingestion-only on VMware

Disable the ingestion-only configuration by redeploying the IBM® API Connect Analytics subsystem with the `ingestion-only` value set to false.

### About this task

---

When you previously configured Analytics for ingestion-only, you deployed only a subset of the Analytics components. If you want to disable the ingestion-only setting, redeploy the subsystem to ensure that all components are available.

### Procedure

---

1. Remove the Analytics service from your API Connect deployment by completing the following steps.
  - a. Unassociate the Analytics service from all gateway services as explained in step 7 in the topic, [Associating an analytics service with a gateway service](#).
  - b. Unregister the Analytics service from Cloud Manager.  
You can unregister the service from the Topology page: click the Options menu next to the service name and select Delete. For information on registering services, see [Registering an analytics service](#).
2. Generate the `analytics-client-ingress` certificate (if you initially created this certificate, it was removed when you deployed using the ingestion-only configuration).  
For information, see [Working with certificates](#).
3. Run the following `apicup` commands to install the complete Analytics subsystem:

```
apicup subsys set analyt ingestion-only=false
apicup subsys install analyt
```
4. Add the new Analytics service to your API Connect deployment by completing the following steps.
  - a. Register the new Analytics service with the Cloud Manager.  
For information on registering a service with the Cloud Manager, see [Registering an analytics service](#).
  - b. Associate the Analytics service with a gateway service as explained in [Associating an analytics service with a gateway service](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Setting rate limits for public APIs on the management service for a VMware environment

Describes the procedure for setting a rate limit for public APIs on the management service. Rate limits provide protection from DDoS (distributed denial of service) attacks.

### Before you begin

---

Note: This article refers to third-party software that IBM does not control. As such, the software may change and this information may become outdated. These instructions assume you have the kubectl command-line tool installed. For more information, see <https://kubernetes.io>.

### About this task

---

Rate limits can be set for public APIs on the management service. Rate limits on APIs help provide protection from DDoS (distributed denial of service) attacks. Without a rate limit, API calls from public APIs are unlimited.

The rate limit configuration requires that the header contains the actual client IP address. Any load balancer or proxy (for example, HAProxy) that is installed in front of the management service must be configured to pass the actual client IP address.

This procedure must be performed on a running API Connect deployment.

This feature is available in API Connect versions 2018.4.1.1 and higher.

Rate limits are calculated as requests per seconds per client.

## Procedure

---

- Set a rate limit:
  1. Add the following entries to an extra-values-file:

```
juhu:
  rateLimitPerClient: 10
  limitRequestOption: "burst=10 nodelay"
```

In this example, the first option sets the rate to 10 requests per second (10r/s). The second option allows 10 requests boosted without delay within < 1 second. You can customize as needed.

The `rateLimitPerClient` property sets `rate`, and `limitRequestOption` sets `[burst=number] [nodelay | delay=number]` in the following nginx configuration:

```
limit_req_zone key zone=name:size rate=rate;
limit_req zone=name [burst=number] [nodelay | delay=number];
```

- Note that `zone` has been pre-defined and can't be configured. For details, see the nginx.org doc [Module ngx\\_http\\_limit\\_req\\_module](#).
  - If you don't have an extra-values-file, you can create a new one. See [Creating an extra values file in a Kubernetes environment](#).
2. Run `apicup` to update the settings in the deployed Management subsystem.

```
apicup subsys install <management-subsystem>
```

Note: If the rate limit has been reached on the management subsystem, the client will get an HTTP error: **429 Too Many Requests**.

3. Validate that the juhu pod has restarted by listing the pods:

```
kubectl get pods -n <namespace> | grep juhu
```

4. Check the AGE column to ensure a new juhu pod has started.
5. If the older juhu pod is still running, delete it with the following command:

```
kubectl delete pods -n <namespace> <old-juhu-pod>
```

- Disable a rate limit:
    1. Validate that the juhu pod has restarted by listing the pods:
- ```
kubectl get pods -n <namespace> | grep juhu
```
2. Check the AGE column to ensure a new juhu pod has started.
  3. If the older juhu pod is still running, delete it with the following command:

```
kubectl delete pods -n <namespace> <old-juhu-pod>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Dynamically re-registering and reconfiguring a Gateway service in a VMware deployment

In API Connect, Gateway services do not persist their configuration settings by default. Instead, the master configuration is stored on the Management server and the *Dynamic Reregistration and Reconfiguration* (DRR) mechanism resynchronizes configuration data when needed. The DRR process is used when proper High Availability/Disaster Recovery (HA/DR) is not configured, or if a manual resynchronization is required.

If a Gateway service is not configured properly for resiliency and is restarted, the gateways in the Gateway service will lose the configuration from the Management server. Configuration data from the Management server is maintained on the gateway service according to the gateway peering configuration on the gateways.

## Preventing the loss of configuration data

---

For high availability in production environments, use a minimum of three gateways in the Gateway service. In non-production environments, if a single gateway is used, then availability can be improved by ensuring the persistence setting on the gateway peering configuration is set to `local://` or `RAID`; these settings apply to virtual and physical appliances only.

## Forcing re-population of the configuration data

---

To repopulate the configuration data, complete the steps in the following section to trigger an outage-less DRR.

## Triggering an outage-less DRR

---

To force a Dynamic Reregistration and Reconfiguration across a peer group from within the gateway service without requiring a restart, complete the following steps using the CLI. This process flushes the primary peering object configured for the apic-gw-service. This task can only be performed with CLI, and requires the default admin ID for running the `diag` command.

1. Use `ssh` to connect to gateway service.

2. Switch to the API Connect application:

```
switch <APIC_APP_DOMAIN>
```

3. Show the current gateway-peering-status:

```
show gateway-peering-status
```

4. If the server to which the ssh connection was made is not primary for the apic-gw-service gateway-peering object, force this server to become primary:

```
config; gateway-peering-switch-primary <gateway-peering-name-for-apic-gw-service>;
```

5. Flush the apic-gw-service gateway-peering object:

```
diag; gateway-peering-flush <gateway-peering-name-for-apic-gw-service>; exit
```

6. Disable and then re-enable the apic-gw-service object for every member of the gateway service:

```
config; apic-gw-service; admin-state disabled; exit
apic-gw-service; admin-state enabled; exit; exit
```

7. Confirm that the apic-gw-service was flushed and is now waiting for gateway service registration:

Look in the debug log target for the apic-gw-service, located in `logtemp:///` and check for a message similar to the following example:

```
20200618T232339.332Z [0x88e000d7][apic-gw-service][notice]
apic-gw-service(default): tid(2653): Waiting for gateway service registration.
```

The DRR will be triggered by the next arrival of a webhook event. Starting with API Connect Version 2018.4.1.2, the Management server sends a heartbeat to the gateway at five-minute intervals, prompting the gateway to check whether it has lost its configuration and if so, trigger a DRR. In Version 2018.4.1.8 and later, if the Management server has previously marked a Catalog or cloud as being unavailable for a particular gateway service, a successful heartbeat triggers synchronization for that Catalog or cloud on that gateway service.

8. (Optional) Force a BAU webhook event to be sent from API Manager to trigger the DRR:

If you do not want to wait for the Management server to send a heartbeat, you can trigger the DRR manually by completing the following steps:

- a. Open the API Manager.
- b. Open the Catalog.
- c. Click Settings > Edit.
- d. Update the Summary field with some text so that it is modified.
- e. Click Save.

When the event is received, the DRR is initiated.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding disk space to a VMware appliance

Increase disk space on the VMware appliance.

### About this task

---

For multi-node deployment of the appliance, only one VM at a time should have its disk space augmented. The subsystem should be in good health before proceeding to the next node. Health of the subsystem can be checked with `apicup subsys health-check <subsystem-name>`.

### Procedure

---

1. Shutdown the virtual machine, either by issuing a shutdown command as root from the VM, or by using the VMware console or command line interface. For example, use the VMware console user interface and click the red rectangle.
2. Use either the VMware console or command line interface to add a new device to the VM, of type Hard Disk and of the desired size to be added to the appliance. For example, on the VMware console:
  - a. Click Edit settings.
  - b. Click Add new device. From the drop-down menu, select Hard Disk.
  - c. Enter size in GB and click OK.
3. Use either the VMware console or the command line interface to power on the VM. As the VM starts it should detect the newly added disk and use it to augment the available capacity for the `/data/secure` mount point.

4. Optionally you may verify that a new disk `sdc` was added (or `sdd` if you already added one disk, and so on). Log in to the server and observe the new disk. For example:

```
root@apimdev1139:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0  100G  0 disk
├─sda1                               8:1    0  99.9G  0 part  /
├─sda14                              8:14   0    4M    0 part
├─sda15                              8:15   0   106M  0 part  /boot/efi
sdb                                  8:16   0   250G  0 disk
├─apicconnect-data                 252:0   0   260G  0 lvm
├─apicSecureDisk                  252:1   0   260G  0 crypt /data/secure
sdc                                  8:32   0    10G  0 disk
├─apicconnect-data                 252:0   0   260G  0 lvm
├─apicSecureDisk                  252:1   0   260G  0 crypt /data/secure
sr0                                  11:0    1    44K  0 rom
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Running a filesystem check on a VMware root partition

You can run a filesystem check on the appliance root filesystem at boot time.

### About this task

If the appliance root filesystem is mounted read-only because it is corrupt, use the following steps to run `fsck` on root partition:

### Procedure

1. Access to the VMware console for the Appliance, and after causing it to reboot, make sure to bring up a console to the appliance (as needed to give focus to the window by clicking the mouse into it), and press the `Esc` key repeatedly.  
You will see a boot screen similar to the following:

```
GNU GRUB v2.02
.
.
- Ubuntu
- Advanced options for Ubuntu
.
.
Press Enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line
```

2. Press `e` to edit the commands before booting, and add the lines `fsck.mode=force` and `fsck.repair=yes`.  
For example, the boot screen contents resemble:

```
GNU GRUB v2.02
.
.
set params 'Ubuntu'
  fsck.mode=force
  fsck.repair=yes
.
.
Press Ctrl-x or F10 to boot
.
```

3. Press `F10` to boot.
4. The following command should show that `fsck` was run recently:

```
root@ip-230:~# sudo tune2fs -l /dev/sda1 | grep "Last checked"
Last checked:          Wed Jul  1 19:32:25 2020
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing an appliance data disk

You can use `apic` commands to manage appliance data disks in your VMware deployment.

The API Connect deployment on VMware uses 2 partitions. The first contains the base operating system, while the second is encrypted so that any customer data that is stored on disk uses the encrypted volume. Encryption is done with Linux Unified Key Setup (LUKS) disk encryption.

During installation of each API Connect subsystem, you specify a password `HD-PASSWD` when configuring the host:

```
apicup hosts create <SUBSYS> <HOSTNAME> <HD-PASSWD>
```

If you want to force a restart of processes, without requiring a full restart of the virtual machine, you can use the command `apic lock` to stop the Kubernetes node on the virtual machine and lock the secured storage. When you are ready to restart processes, you can use `apic unlock` to restart the Kubernetes node. The command `apic unlock` uses the password to unlock the partition so that files can be read from it and written to it. The unlock command also starts the `apic` daemon, also known as the `appliance-manager` service.

You can check that status of this service with:

```
sudo systemctl status appliance-manager
```

One `appliance-manager` daemon runs on each node, where it manages the Kubernetes cluster, processes update and upgrade requests, and gathers logs.

| apic command         | Description                                               |
|----------------------|-----------------------------------------------------------|
| <code>lock</code>    | Shut down appliance services and lock the secured storage |
| <code>logs</code>    | Retrieve logs from all nodes in the cluster               |
| <code>status</code>  | Report on cluster status                                  |
| <code>unlock</code>  | Unlock the secure storage and start the appliance         |
| <code>version</code> | Get the API Connect appliance base version                |

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting the API Connect upgrade on VMware

Troubleshoot your API Connect upgrade on VMware.

Note: Do not install any extra software or run any commands on the VMs that are not documented in the IBM documentation or otherwise advised by IBM. When troubleshooting API Connect, do not use `kubect1 exec` commands to access API Connect pods unless advised by IBM. Do not make any changes on the deployed VMs unless documented here or otherwise advised by IBM. Attempting to manually update packages, adding new users, or installing new software will likely cause problems. Operating system updates are handled by API Connect fix packs.

- [Restarting unhealthy cassandra pods](#)
- [Recreating a calico pod that is in the CrashLoopBackOff state](#)

---

## Restarting unhealthy cassandra pods

If you complete an upgrade and find that one or more of the cassandra pods (prefixed with `apiconnect-apiconnect-cc`) are not responding:

1. Run the following checks to verify that the pods require a restart:

- Check the status of the pods to see if they are marked as `Running` but not `Ready` as in:

```
apiconnect-apiconnect-cc-g4rhq 0/1 Running
```

- Check the `describe` of the pods for an event log with a warning similar to the following example:

```
Warning Unhealthy 2m6s (x177 over 38m) kubelet Readiness probe failed: Cassandra is either decommissioning or upgrading
```

2. Restart each pod by deleting it with the following command (which recreates the pod):

```
kubect1 delete pod <pod_name>
```

Note: If there are multiple cassandra pods in this state, then delete them one at a time' wait until the previously deleted pod comes up and shows as `Running` in the `Ready` state before deleting the pod.

---

## Recreating a calico pod that is in the CrashLoopBackOff state

If you complete an upgrade and find that one or more of the calico pods are in the `CrashLoopBackOff` state, you can delete the pod to recreate it. Complete the following steps:

1. Get the names of the calico pods:

```
kubect1 -n kube-system get pods | grep calico
```

2. Recreate each pod by deleting it with the following command:

```
kubect1 -n kube-system delete pod <calico_pod_name>
```

- [Checking cluster health on VMware](#)

You can use `apicup` to check the health of the API Connect clusters in your VMware deployment.

- [Determining status of a cluster on VMware](#)

You can determine the health of an API Connect cluster on VMware

- [Obtaining simple health check data of Developer Portal sites by using a REST API call](#)

Call a simple health check API from your external load balancer to dynamically determine whether a specific Developer Portal site in a cluster is working. This API call can be used by a load balancer to help determine where to route traffic.

- [Monitoring the network with SNMP](#)  
Presents a table of OID trees that you can poll using SNMP Get.
- [Gathering logs for a VMware environment](#)  
The `generate_postmortem.sh` script gathers all logs for troubleshooting and diagnostics.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Checking cluster health on VMware

You can use `apicup` to check the health of the API Connect clusters in your VMware deployment.

Note: The `apicup` health check command is available on Version 2018.4.1.6 or later. For deployments on Version 2018.4.1.5 or earlier, use the instructions on [Determining status of a cluster on VMware](#).

The `health-check` command checks a number of criteria to determine the health of their cluster. When all criteria are successfully met, the command displays no output, and exits with a status of 0. When one or more criteria are not met, the command stops processing and displays a message with the failure and exits with a status of 1.

The command takes no arguments. The only option is the `--verbose` flag. If `--verbose` is set the command prints all checks that were performed.

The `health-check` is run against the namespace that is specified in `apiconnect-up.yaml`.

The command verifies that for the specified subsystem:

- The `apicup` version matches the API Connect release version.
- All cluster members are running the same API Connect release version.
- All cluster members have a deployment status of `Done`.
- Docker is running.
- `kubelet` is running
- The installed subsystem matches the subsystem deployed to Helm.
- For each expected Helm release, the deployed Chart version matches the `apicup` version.
- All Kubernetes-defined nodes are running.
- The Kubernetes Control Plane pods are running:
  - `etcd/kube-apiserver`
  - `kube-system.kube-controller`
  - `kube-scheduler`
  - `kube-apiserver-proxy`
- The add-on Kubernetes Deployments are fulfilled:
  - `coredns`
  - `metrics-server`
  - `tiller-deploy`
- The add-on Kubernetes DaemonSets are fulfilled:
  - `calico-node`
  - `ngress-nginx-ingress-controller`
  - `kube-proxy`

Syntax:

Usage:

```
apicup subsys health-check <SUBSYS> [flags]
```

Flags:

```
-h, --help                help for health-check
--kubeconfig string      (optional) absolute path to the kubeconfig file (default "/Users/<username>/.kube/config")
-v, --verbose            Verbose output
```

Global Flags:

```
--accept-license        Accept the license for API Connect
--debug                 Enable debug logging
```

Example usage, for a subsystem named `mgmt`:

```
../apicup subsys health-check mgmt
```

Note:

- The command flag `--kubeconfig string` does not apply to deployments on VMware.
- In a multi-node OVA cluster, the `apicup subsys health-check` command does not return status information if a majority of the nodes are down. To obtain status information, start more nodes. Status is returned only when a quorum is achieved.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Determining status of a cluster on VMware

You can determine the health of an API Connect cluster on VMware

## About this task

At any time, you can review the status of the API Connect cluster. You should review the status before and after doing configuration tasks such as upgrading.

Note: For Version 2018.4.1.6 or later, you can also use `apicup subsys health-check`. See [Checking cluster health on VMware](#)  
Do not use these instructions on Kubernetes. See [Determining status of a cluster on Kubernetes](#).

## Procedure

1. Verify that the installer version matches your API Connect product version.
  - a. Check the `apicup` version. For example:

```
$/apicup version
APIConnect 2018.4.1.3
Installer 4.0.0
```

Note that for API Connect 2018.4.1.5, the Installer version is 5.0.0.

- b. On each appliance, run `apic version`:  
Example output:

```
INFO[0000] Log level: info
Appliance base 1.3.0
GitCommitSha:2018.4.1.2-11-g617cf1f
Packages:
- appliance-base:1.3.0
- appliance-runtime-1.3.0:1.0.0
- subsystem-management:2018.4.1.3
```

Verify the version number on the `subsystem` line. The previous example shows, for the management subsystem, version `2018.4.1.3` on the line `subsystem-management:2018.4.1.3`.

The Analytics subsystem displays as `subsystem-analytics`. The Developer Portal displays as `subsystem-portal`.

2. Ensure that all nodes have an installation status of **DONE**.  
Run `apic status`:

```
sudo -i
apic status
```

Ensure that the output for each node says:

```
Install stage: DONE
Upgrade stage: UPGRADE_DONE
```

If a node has recently been restarted, the install or upgrade stage might not show **DONE**. In this case, wait a few minutes for background processing to complete and view the status again.

Note that the status command always returns a value for `Upgrade stage`, even when an upgrade has not been attempted. In this scenario, the value might not be `UPGRADE_DONE`. This is normal and can be ignored.

Note: In a multi-node OVA cluster, the `apic status` command does not return status information if a majority of the nodes are down. To obtain status information, start more nodes. Status is returned only when a quorum is achieved.

3. Ensure that all Kubernetes nodes have a status of **READY**:

```
kubect1 get nodes
```

4. Verify that the subsystem Helm release was successful

The Helm release name(s) depend on the subsystem type you are checking. Subsystem type to helm release name:

| Subsystem  | Helm release                   |
|------------|--------------------------------|
| management | apiconnect, cassandra-operator |
| analytics  | apic-analytics                 |
| portal     | apic-portal                    |

Run `helm ls` to show the releases. Ensure that each release for each subsystem has a status of **DEPLOYED**:

```
# helm ls
NAME          REVISION    UPDATED              STATUS      CHART
NAMESPACE
apiconnect    1           Wed Apr 3 11:58:54 2019  DEPLOYED   apiconnect-2.0.0
default
cassandra-operator 1           Wed Apr 3 11:58:38 2019  DEPLOYED   cassandra-operator-1.0.0
default
```

If any release is not **DEPLOYED**, open a ticket with IBM Support, to obtain assistance with troubleshooting the release.

5. Check that the deployed charts are the correct version.

This check needs to be performed on only one node. See the previous steps to determine the Helm release name(s) for the subsystem type.

- Version 2018.4.1.4 or later  
Run the command `helm get [release-name] | grep productVersion | head -1`  

```
# helm get values apiconnect | grep productVersion | head -1
productVersion: 2018.4.1.4
```

If the previous command returns `productVersion: ""`, the deployed charts are for a version prior to 2018.4.1.4. See the following instructions.

- Version 2018.4.1.3 or earlier
  - Management subsystem

• Run `helm get apiconnect | grep apiconnect/apim[:] | head -1`. Compare output to the following table to determine chart `productVersion`:

| Image output                                  | Chart productVersion |
|-----------------------------------------------|----------------------|
| image: "apiconnect/apim:2018.4.1-223-5e98505" | 2018.4.1.0           |
| image: "apiconnect/apim:2018.4.1-267-7151665" | 2018.4.1.1           |
| image: "apiconnect/apim:2018.4.1-339-0d1c113" | 2018.4.1.2           |
| image: "apiconnect/apim:2018.4.1-391-e1df735" | 2018.4.1.3           |

• Run `helm get cassandra-operator | grep apiconnect/cassandra-operator[:] | head -1`. Compare output to the following table to determine chart `productVersion`.

| Image output                                                                                               | Chart productVersion |
|------------------------------------------------------------------------------------------------------------|----------------------|
| image: "image: apiconnect/cassandra-operator:2018-10-27-11-34-27-b30560eb3775c558f8438abb8e084ed97ca8a34f" | 2018.4.1.0           |
| image: "image: apiconnect/cassandra-operator:2018-11-13-14-39-43-cac5b0e471f011258f38b92c6959e4c30bc21b08" | 2018.4.1.1           |
| image: "image: apiconnect/cassandra-operator:2019-01-07-17-52-36-a571b8b69248e7afd5aac06ca5196ffeb8fd7875" | 2018.4.1.2           |
| image: "image: apiconnect/cassandra-operator:2019-02-19-01-57-00-77482ccb867d08277e9985a3a75cb86385720523" | 2018.4.1.3           |

- Portal subsystem

Run `helm get apic-portal | grep apiconnect/portal-web[:] | head -1`. Compare output to the following table to determine chart `productVersion`.

| Image output                                                                       | Chart productVersion |
|------------------------------------------------------------------------------------|----------------------|
| image: apiconnect/portal-web:2018.4.1-385444eb87824c93e42c85412af8088eac8c7bb9-335 | 2018.4.1.0           |
| image: apiconnect/portal-web:2018.4.1-ea9a86af8cf6f9e241f29f9a7ad860991e017d29-514 | 2018.4.1.1           |
| image: apiconnect/portal-web:2018.4.1-8e3c67a1ec93f9eda7411286b75e19becaab0fb0-773 | 2018.4.1.2           |
| image: apiconnect/portal-web:2018.4.1-31b0250446fd1660cc4622f44b7f8317391c536d-948 | 2018.4.1.3           |

- Analytics subsystem

Run `helm get apic-analytics | grep apiconnect/analytics-client[:] | head -1`. Compare output to the following table to determine chart `productVersion`:

| Image output                                                                                     | Chart productVersion |
|--------------------------------------------------------------------------------------------------|----------------------|
| image: apiconnect/analytics-client:2018-10-18-17-32-08-ba640482b2659c37905271d6a318fa0fc902aebd" | 2018.4.1.0           |
| image: apiconnect/analytics-client:2018-11-21-11-19-56-ba640482b2659c37905271d6a318fa0fc902aebd" | 2018.4.1.1           |
| image: apiconnect/analytics-client:2019-01-07-15-16-59-9f8c1b7bb4a8f9a0bf9529bab166f752821b62e4" | 2018.4.1.2           |
| image: apiconnect/analytics-client:2019-02-12-12-52-40-700d61ffb1c2ff2c9860d6d75f1620fbceea64d6" | 2018.4.1.3           |

6. Review the results of the previous step, and ensure that the product version matches the `apicup` version and the appliance version.

If the chart version is wrong it is likely that either the wrong version of Install Assist was used for `apicup subsys install`, or an old `--plan-dir` was passed to the install command. Verify the correct version of `apicup` is being used and try to re-install by using `apicup subsy install`. It is safe to run the command multiple times.

7. Use `kubectl` to check the deployments and validate that the number of deployments matches for `DESIRED`, `CURRENT`, and `AVAILABLE`.

If the numbers don't match, it is possible that one of the pods is still starting, or is having problems. Another possibility is that Kubernetes encountered problems trying to scale down an old replica set. Check for replica-sets that are for the same deployment and have a number count greater than 0 for `DESIRED`. Note that the replica-sets for a deployment start with the same name.

You can use `kubectl get rs` to try to determine why the replica-set is not scaling down. For example, the following output shows a cluster in a non-healthy state.

```
# kubectl get deployments
NAME                                DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
apiconnect-a7s-proxy                3         3         3             1           2h
apiconnect-apim-v2                  3         3         3             3           2h
apiconnect-client-dl-srv            2         3         2             1           2h
apiconnect-juhu                     3         4         2             2           2h
apiconnect-ldap                     3         4         2             2           2h
apiconnect-lur-v2                   3         4         2             2           2h
apiconnect-ui                       2         3         2             1           2h
cassandra-operator-cassandra-operator 1         1         1             1           2h

# kubectl get rs
NAME                                DESIRED  CURRENT  READY  AGE
apiconnect-a7s-proxy-56c9f975c      3         3         1       9m
apiconnect-apim-v2-55d77df65        3         3         3       2h
apiconnect-client-dl-srv-56cfd98dd5 2         2         0       9m
```



|                                                  |   |   |   |    |
|--------------------------------------------------|---|---|---|----|
| apiconnect-client-dl-srv-6bc7d48477              | 1 | 1 | 1 | 2h |
| apiconnect-juhu-744db85f                         | 2 | 2 | 0 | 9m |
| apiconnect-juhu-744f9cbc89                       | 2 | 2 | 2 | 2h |
| apiconnect-ldap-6c855d4b                         | 2 | 2 | 2 | 2h |
| apiconnect-ldap-d599cb954                        | 2 | 2 | 0 | 9m |
| apiconnect-lur-v2-79556d69c                      | 2 | 2 | 0 | 9m |
| apiconnect-lur-v2-7f97c7b8b5                     | 2 | 2 | 2 | 2h |
| apiconnect-ui-69c8df97                           | 2 | 2 | 0 | 9m |
| apiconnect-ui-84c97657c5                         | 1 | 1 | 1 | 2h |
| cassandra-operator-cassandra-operator-7c88b6b889 | 1 | 1 | 1 | 2h |

8. Verify that the DaemonSets are fulfilled.

For each subsystem run `kubectl get ds` and ensure that the values match for **DESIRED**, **CURRENT**, and **READY**.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Obtaining simple health check data of Developer Portal sites by using a REST API call

Call a simple health check API from your external load balancer to dynamically determine whether a specific Developer Portal site in a cluster is working. This API call can be used by a load balancer to help determine where to route traffic.

### About this task

You can use the site health REST API to determine whether a particular Developer Portal site is running. The site health API returns the current system time of the site if both the database and web server are running. This API is fast and puts no load on the system, so it is ideal for use with load balancers to help them determine where to route traffic.

### Procedure

To call the site health REST API, append `/health` to the end of your Developer Portal site URL in your web browser, as follows:

`site_url/health`

Where `site_url` is the URL of the Developer Portal site that you want to check.

If both the database and web server of the site are running, the web browser returns the current system time. For example:

`1511367695`

If either the database or the web server of the site is not running, the web browser returns an error that the site can't be reached.

If you do get an error from the `/health` endpoint, refer to the [must gather logs page](#) and obtain the `portal-www web` logs, so that you can see more information on the error.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Monitoring the network with SNMP

Presents a table of OID trees that you can poll using SNMP Get.

Note:

- This information applies to API Connect deployments in VMware environments. It does not apply to Kubernetes deployments.
- Only SNMP v2 is supported.
- If the server has two or more network interfaces, the SNMP request is sent only to the first network interface.
- SNMP is enabled by default for all hosts in appliance versions 2018.2.x, 2018.3.x, 2018.4.1.0, 2018.4.1.1 and 2018.4.1.2
- SNMP is disabled by default for all hosts in all other appliance versions.

Simple Network Management Protocol (SNMP) collects information from network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP). SNMP employs addresses for network devices using a hierarchical numbering system called Object Identifiers (OID). The following table provides OID addresses for polling in SNMP to monitor API Connect servers.

Each entry in the table represents many individual items. Use `snmpwalk` or another SNMP polling utility to see the complete list. You can poll the following OID trees:

Table 1.

| OID            | SNMP Name          | Notes |
|----------------|--------------------|-------|
| .1.3.6.1.2.1.1 | SNMPv2-MIB::system |       |
| .1.3.6.1.2.1.2 | IF-MIB::interfaces |       |
| .1.3.6.1.2.1.4 | IP-MIB::ip         |       |
| .1.3.6.1.2.1.5 | IP-MIB::icmp.      |       |
| .1.3.6.1.2.1.6 | TCP-MIB::tcp       |       |

| OID                  | SNMP Name                     | Notes                                                                                                                                                          |
|----------------------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .1.3.6.1.2.1.7       | UDP-MIB::udp                  |                                                                                                                                                                |
| .1.3.6.1.2.1.11      | SNMPv2-MIB::snmp              |                                                                                                                                                                |
| .1.3.6.1.2.1.25.1    | HOST-RESOURCES-MIB::hrSystem  | Excluding .1.3.6.1.2.1.25.1.3 HOST-RESOURCES-MIB::hrSystemInitialLoadDevice<br>Excluding .1.3.6.1.2.1.25.1.4 HOST-RESOURCES-MIB::hrSystemInitialLoadParameters |
| .1.3.6.1.2.1.25.2    | HOST-RESOURCES-MIB::hrStorage |                                                                                                                                                                |
| .1.3.6.1.2.1.25.3    | HOST-RESOURCES-MIB::hrDevice  |                                                                                                                                                                |
| .1.3.6.1.4.1.2021.4  | UCD-SNMP-MIB::memory          |                                                                                                                                                                |
| .1.3.6.1.4.1.2021.10 | UCD-SNMP-MIB::laTable         | CPU Load Average                                                                                                                                               |
| .1.3.6.1.4.1.2021.11 | UCD-SNMP-MIB::systemStats     |                                                                                                                                                                |

## Enabling SNMP

Enable SNMP on each individual OVA node with the following commands from an OVA shell prompt as root:

```
systemctl start snmpd
systemctl enable snmpd
```

Sample output:

```
root@mgmt:~# systemctl start snmpd
root@mgmt:~# systemctl enable snmpd
snmpd.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install enable snmpd
insserv: warning: current start runlevel(s) (empty) of script `snmpd' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `snmpd' overrides LSB defaults (0 1 6).
```

## Disabling SNMP

Disable SNMP on each individual OVA node with the following commands from an OVA shell prompt as root:

```
systemctl stop snmpd
systemctl disable snmpd
```

Sample output:

```
root@mgmt:~# systemctl stop snmpd
root@mgmt:~# systemctl disable snmpd
snmpd.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install disable snmpd
insserv: warning: current start runlevel(s) (empty) of script `snmpd' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `snmpd' overrides LSB defaults (0 1 6).
```

## Checking SNMP status

Check whether SNMP is enabled on each individual OVA node with the following commands from an OVA shell prompt:

```
systemctl status snmpd
```

Sample output when SNMP is enabled and running:

```
apicadm@mgmt:~$ systemctl status snmpd
snmpd.service - LSB: SNMP agents
  Loaded: loaded (/etc/init.d/snmpd; bad; vendor preset: enabled)
  Active: active (running) since Wed 2019-02-06 03:53:02 UTC; 2 days ago
    Docs: man:systemd-sysv-generator(8)
   Tasks: 1
  Memory: 6.0M
     CPU: 3min 1.641s
  CGroup: /system.slice/snmpd.service
          └─1366 /usr/sbin/snmpd -Lsd -lf /dev/null -u snmp -g snmp -I -smux mteTrigger mteTriggerConf -p /run/snmpd.pid
...

```

Sample output when SNMP is stopped and disabled:

```
root@mgmt:~# systemctl status snmpd
snmpd.service - LSB: SNMP agents
  Loaded: loaded (/etc/init.d/snmpd; bad; vendor preset: enabled)
  Active: inactive (dead)
    Docs: man:systemd-sysv-generator(8)
...

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Gathering logs for a VMware environment

The `generate_postmortem.sh` script gathers all logs for troubleshooting and diagnostics.

## Before you begin

You can download the `generate_postmortem.sh` script directly to the OVA. If the OVA does not have internet access, first download the script to another machine and then transfer it to the OVA.

## About this task

When contacting IBM Support, several logs are required to assist in troubleshooting or diagnostics. The `generate_postmortem.sh` script gathers all the required logs using a single shell script.

## Procedure

1. Connect to the target appliance via SSH then switch to the `root user` with the following commands:

```
ssh apicadm@{ova appliance hostname}
sudo -i
```

2. Download the `generate_postmortem.sh` script. Enter the following command:

```
curl -o generate_postmortem.sh https://raw.githubusercontent.com/ibm-apiconnect/v2018-
postmortem/master/generate_postmortem.sh
```

3. Add execute permissions to the `generate_postmortem.sh` script. Enter the following command:

```
chmod +x generate_postmortem.sh
```

4. Run the postmortem tool using the following command:

```
./generate_postmortem.sh --ova --pull-appliance-logs
```

5. (Optional) Currently, log generation is supported for Portal and Management subsystems. Specify either management or portal, or all subsystems. Enter the following arguments:

```
Portal: ./generate_postmortem.sh --ova --diagnostic-portal --pull-appliance-logs
Management: ./generate_postmortem.sh --ova --diagnostic-manager --pull-appliance-logs
```

6. (Optional) If there are errors when running the `generate_postmortem.sh` script, enter the following command:

```
./generate_postmortem.sh --ova --debug &>debug.log
```

Note: Offboarding the logs to an external server is the recommended best practice for long term storage of log data. See [Configuring remote logging for a VMware deployment](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Changing logging levels

You can enable logging for entry and exit trace and for large payloads for apim-v2 pods.

## About this task

For apim-v2 pods in Version 2018.4.1.9 or later, the default logging settings do not include entry and exit trace and logging of large payloads.

Logging of large payloads is typically needed if you are logging `apim:webhookPayload` or other processes that log large variables.

You can also set the debug level without needing to restart the pods.

You can enable the settings through either the APIM deployment YAML file, the toolkit, or the REST APIs. Note, you must use the toolkit or REST API to set the debug level without restarting the pods.

## Procedure

- Updating the settings in the APIM deployment YAML file

Note: This action causes a pod restart.

- The DEBUG variable works the same as prior releases.

```
- env:
  - name: DEBUG
    value: audit,bhendi:error,bhendi:probe,bhendi:flags,bhendi:audit,bhendi:webhookAudit,bhendi:cassandra-
transactions,apim:server,apim:error,apim:routes:*,apim:routesc:*,apim:oidc,apim:oidc:*,apim:webhook:audit,apim:taskma
nager:info
```

To enable entry and exit trace, add `trace:*` to the start of the debug string:

```
- env:
  - name: DEBUG
    value: trace:*,audit,bhendi:error,bhendi:probe,bhendi:flags,bhendi:audit,bhendi:webhookAudit,bhendi:cassandra-
```

```
transactions, apim:server, apim:error, apim:routes:*, apim:routesc:*, apim:oidc, apim:oidc:*, apim:webhook:audit, apim:taskmanager:info
```

- o To enable large object logging:

```
- env:  
  - name: VELOX_LOG_FULL_PAYLOAD  
    value: "true"
```

- Updating the settings by using the toolkit

Note: This action does not cause a pod restart. Also, this action is for a single APIM pod environment only.

```
apic log-spec:get  
apic log-spec:update LOG_SPEC_FILE
```

Where the spec file contains:

```
{"specification": "apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error", "large_objects": true}
```

- Updating the settings by using the REST API

Note: This action does not cause a pod restart. Also, this action is for a single APIM pod environment only.

```
GET /cloud/api/log-spec  
PUT /cloud/api/log-spec
```

For example, to set debug level and large objects logging:

```
curl -k https://172.16.140.212:3003/api/cloud/log-spec -X PUT -H "Authorization: Bearer $BEARER"  
-H "Accept: application/json" -d '{"specification":  
"apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error", "large_objects": true}'  
-H "Content-Type: application/json"
```

The command responds with:

```
{  
  "specification": "apim:routes:log_spec,audit,bhendi:error,apim:server,apim:error",  
  "large_objects": true,  
  "message": "Successfully changed the log specification on all apim-v2 pods. Success on IP(s): 172.16.140.212,  
172.16.140.213"  
}
```

Note: The response might include the following warning, which can be ignored:

```
WARNING: Could not change the log specification on all apim-v2 pods. Success on IP(s): 4.5.6.7 Failed on IP(s): 1.2.3.4
```

To view the REST APIs for logging, go to [API Connect REST APIs](#), and select IBM API Connect Platform - Cloud Management API 2.0.0 reference\_>\_Resource: Log Spec.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## IBM Cloud Private

Use the instructions in this section to install, upgrade, and maintain API Connect on IBM Cloud Private.

- [Installing and upgrading on IBM Cloud Private](#)  
Use these instructions to install or upgrade a deployment of API Connect on IBM Cloud Private.
- [Backing up and restoring an IBM Cloud Private catalog deployment](#)  
You can back up and restore the API Connect Management subsystem and Portal subsystem on IBM Cloud Private.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing and upgrading on IBM Cloud Private

Use these instructions to install or upgrade a deployment of API Connect on IBM Cloud Private.

- [Deployment overview for endpoints and certificates](#)  
Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.
- [Deploying to an IBM Cloud Private environment](#)  
When you deploy IBM® API Connect to an IBM Cloud Private environment, you have the flexibility of accessing a cloud-based product and the added security of an on-premises product. Depending on the version of IBM Cloud Private, you can use the IBM Cloud Private catalog installation or the Install Assist installation method.
- [Installing the toolkit](#)
- [Upgrading with IBM Cloud Private Catalog](#)  
You can update an API Connect deployment with the IBM Cloud Private Catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

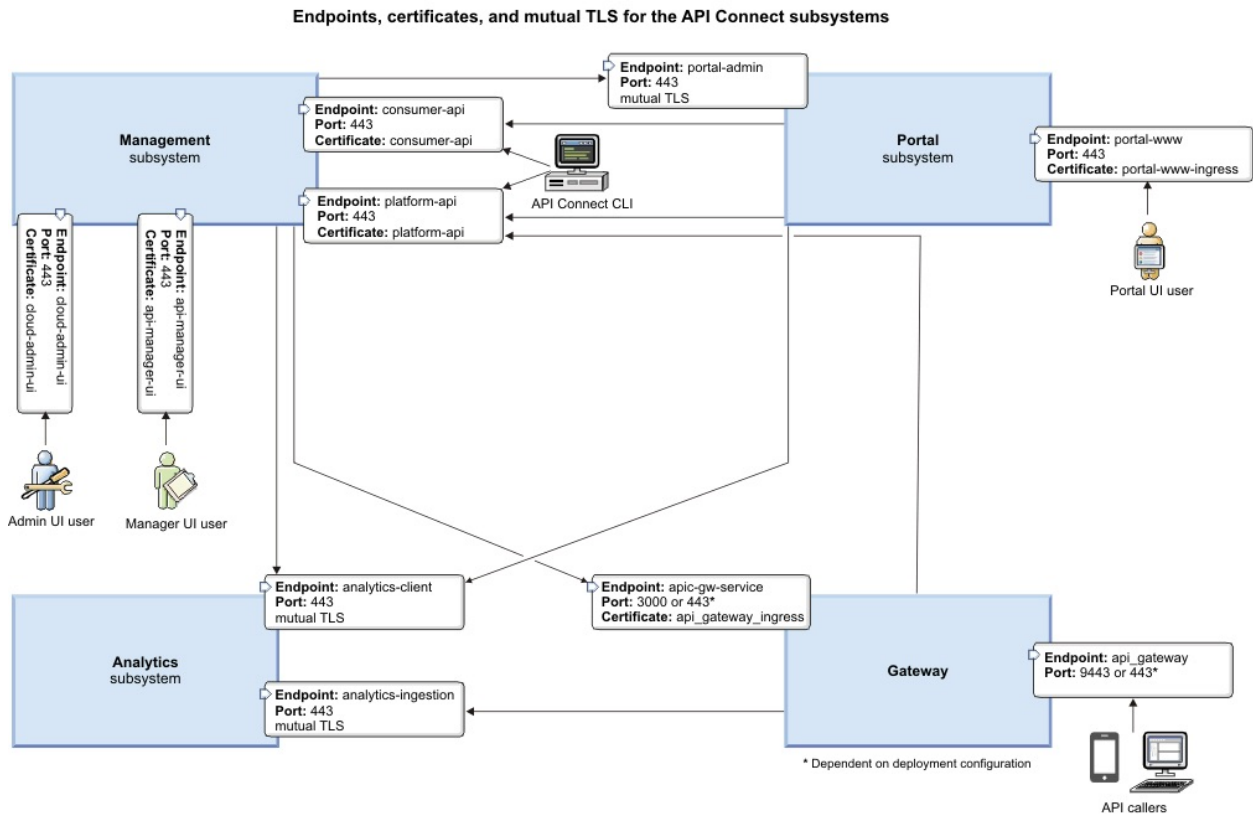
## Deployment overview for endpoints and certificates

Refer to this diagram to view the connections and dataflow among the API Connect subsystems, including the endpoints and custom certificates, and the mutual TLS points.

### Introduction

When deploying API Connect, you will create one or more endpoints for the subsystems and then configure certificates or mutual TLS for most endpoints. [Figure 1](#) shows the endpoints for each subsystem by name, the name of the certificate that secures the endpoint, and whether mutual TLS is required. It also shows the ports consumed by the endpoints, which are standard for HTTP and HTTPS.

Figure 1. Deployment Overview diagram



### Configuring endpoints

The endpoints are configured by the Install Assist program using the APICUP installer. They are set for each subsystem. Endpoints are also entered when configuring the Topology for the Gateway, Portal, and Analytics subsystems in Cloud Manager.

Instructions for installing into an IBM Cloud Private environment are here: [Deploying to an IBM Cloud Private environment](#).

| Subsystem  | Endpoints        | Description                                                                                                                                        | Certificates       |
|------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Management | cloud-admin-ui   | Configured using APICUP installer. Endpoint on the management server for communication with the Cloud Manager user interface.                      | cloud-admin-ui     |
|            | api-manager-ui   | Configured using APICUP installer. API Manager URL endpoint on the management server for communication with the API Manager user interface.        | api-manager-ui     |
|            | consumer-api     | Configured using APICUP installer. Platform REST API endpoint for running consumer APIs on the management server.                                  | consumer-api       |
|            | platform-api     | Configured using APICUP installer. Platform REST API endpoint for running admin and provider APIs on the management server.                        | platform-api       |
| Portal     | portal-admin     | Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS. | mutual TLS         |
|            | portal-www       | Configured using APICUP installer. Portal Web site URL entered in Cloud Manager. Used publicly to access Portal.                                   | portal-www-ingress |
| Analytics  | analytics-client | Configured using APICUP installer. Corresponds to Management Endpoint entered in Cloud Manager. Requires a TLS profile configured with mutual TLS. | mutual TLS         |

| Subsystem | Endpoints           | Description                                                                                                                                                                                  | Certificates            |
|-----------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
|           | analytics-ingestion | Configured using APICUP installer. The analytics-ingestion endpoint is used by the Gateway service to push data to the Analytics service. Requires a TLS profile configured with mutual TLS. | mutual TLS              |
| Gateway   | apic-gw-service     | Configured using APICUP installer. This is the endpoint the gateway uses for network communication. Enter this endpoint as the Management Endpoint entered in Cloud Manager.                 | apic-gw-service-ingress |
|           | api-gateway         | Configured using APICUP installer. This is the endpoint the gateway uses for API traffic. Enter this endpoint as the API Invocation Endpoint in Cloud Manager.                               | api-gateway-ingress     |

## Configuring certificates

The certificates are configured by the Install Assist program using the APICUP installer. The certificates for the endpoints are usually configured as custom certificates as described in [Setting custom certificates](#).

## Configuring mutual TLS

Mutual TLS is configured for TLS profiles in Cloud Manager. See [Creating a TLS Server Profile](#).

## Configuring a proxy

If a Developer Portal is deployed externally to the management server zone, it does not have access to the consumer and product APIs. You need to configure a proxy to enable communication. For more information, see [Configuring a proxy](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying to an IBM Cloud Private environment

When you deploy IBM® API Connect to an IBM Cloud Private environment, you have the flexibility of accessing a cloud-based product and the added security of an on-premises product. Depending on the version of IBM Cloud Private, you can use the IBM Cloud Private catalog installation or the Install Assist installation method.

- **Deploying with the IBM Cloud Private catalog**  
When you deploy IBM API Connect into the IBM Cloud Private Version 3.1.0 or later, you can use the IBM Cloud Private catalog deployment.
- **Deploying with the Install Assist tool**  
You can deploy IBM API Connect into any version of the IBM Cloud Private environment by using the Install Assist method.
- **Migrating an IBM Cloud Private catalog install to apicup**  
You can migrate an IBM Cloud Private installation that was done through the ICP catalog to an Install Assist (**apicup**) installation.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying with the IBM Cloud Private catalog

When you deploy IBM® API Connect into the IBM Cloud Private Version 3.1.0 or later, you can use the IBM Cloud Private catalog deployment.

## Before you begin

This procedure lists the steps to install using the IBM Cloud Private catalog installation. If you want to install using the API Connect Install Assist method, see [Deploying with the Install Assist tool](#).

To install API Connect in the IBM Cloud Private environment by using the IBM Cloud Private catalog method, you must have the following items:

- The IBM Cloud Private environment installed on your server or servers. See [IBM® Cloud Private v3.1.0 documentation](#) in the IBM Cloud Private documentation for additional information about how to install IBM Cloud Private.
- The IBM Cloud Private CLI. See [Installing the IBM Cloud Private CLI](#).
- Access to IBM Fix Central so you can download the latest version of the following files that you have purchased:
  - IBM API Connect Enterprise Production V2018.x for IBM Cloud Private
  - IBM API Connect Professional Production V2018.x for IBM Cloud Private

These packages contain the files that are required for the installation, and the Helm chart for the product.
- Authentication to your IBM Cloud Private Docker registry. See [Configuring authentication for the Docker CLI](#) for the required steps.
- A configured Helm CLI. See [Setting up the Helm CLI](#) for the procedure.
- Ensure you have supported software requirement versions. See [IBM API Connect Version 2018 software product compatibility requirements](#).

Note: For installation and upgrade prerequisites, configuration instructions, and changes between releases, refer to the Helm chart README and Release Notes in the IBM Cloud Private catalog entry for API Connect.

## About this task

IBM Cloud Private provides an environment based on Kubernetes and Docker where you can host API Connect on a local cluster. Members with access to the cluster can use the service by subscribing to it.

Important: For a CP4I or ICP catalog installation, the certificate secret files are required to restore the database and to upgrade your deployment. These files are located in the operator pod. Both files should be exported as Kubernetes secrets and, along with the apiconnect-up.yml file, backed up to a secure and permanent location from where they can always be retrieved. See step 14 for more details.

## Procedure

1. Download and extract the zip file from IBM Fix Central that corresponds to the product that you purchased from [Passport Advantage®](#).

The package names are:

- IBM API Connect Enterprise Production V2018.x for IBM Cloud Private
- IBM API Connect Professional Production V2018.x for IBM Cloud Private

2. Log in to the CLI of your IBM Cloud Private cluster by entering the following command:

```
cloudctl login -a https://cluster_ip:8443 --skip-ssl-validation
```

Where *cluster\_ip* is the IP address or fully-qualified URL and port number of your IBM Cloud Private cluster environment.

3. Create a Helm TLS secret containing base64-encoded **ca.pem**, **cert.pem**, and **key.pem** artifacts.

For example, you can use the following command on a single line:

```
kubectl create secret generic helm-tls-secret \
  --from-file=cert.pem=$HOME/.helm/cert.pem \
  --from-file=ca.pem=$HOME/.helm/ca.pem \
  --from-file=key.pem=$HOME/.helm/key.pem \
  --namespace=<TARGET_NAMESPACE>
```

If you are deploying to the **default** namespace, the **--namespace** parameter is not needed. If you are deploying to any other namespace, you must specify **--namespace** or else the install will fail. The APIC operator pod reads secrets from the namespace it is deployed into.

4. Create a Docker registry secret.

This is a secret for an ICP registry where APIC build images are held.

```
kubectl create secret docker-registry <name> \
  --docker-server=registry \
  --docker-username=user \
  --docker-password=password \
  --namespace=TARGET_NAMESPACE
```

5. Log in to the cluster's Docker registry by entering the following command:

```
docker login cluster_ip:8500
```

6. Import the API Connect image archives into IBM Cloud Private by entering the following command for each image archive type that is not installed:

```
cloudctl catalog load-archive --archive image_archive_filename
```

Where *image\_archive\_filename* is one of the following file names, depending on which version of API Connect for IBM Cloud Private you purchased:

- ibm-apiconnect-ent.tgz
- ibm-apiconnect-pro.tgz

7. From the IBM Cloud Private management console, select Menu > Manage > Helm Repositories.

8. Select Sync Repositories.

The catalog is updated with the imported API Connect Helm chart.

9. Select the Catalog menu to view the list of Helm charts, and select your API Connect Helm chart.

Depending on which catalog you purchased, the Helm chart has one of the following names:

ibm-apiconnect-ent

This is the chart for the Enterprise edition.

ibm-apiconnect-pro

This is the chart for the Professional edition.

10. Select Configure to view and update the configuration settings for the Helm chart. Refer to the following table for endpoint mapping for the ICP Helm Chart:

| Endpoint created using APICUP installer | Endpoint created in ICP Helm chart with Helm CLI | Endpoint in the ICP UI       |
|-----------------------------------------|--------------------------------------------------|------------------------------|
| api-manager-ui                          | management.apiManagerUiEndpoint                  | Platform API Endpoint        |
| cloud-admin-ui                          | management.cloudAdminUiEndpoint                  | Cloud Admin UI Endpoint      |
| consumer-api                            | management.consumerApiEndpoint                   | Consumer API Endpoint        |
| platform-api                            | management.platformApiEndpoint                   | API Manager UI Endpoint      |
| portal-admin                            | portal.portalDirectorEndpoint                    | Portal Director Endpoint     |
| portal-www                              | portal.portalWebEndpoint                         | Portal Web Endpoint          |
| analytics-client                        | analytics.analyticsClientEndpoint                | Analytics Client Endpoint    |
| analytics-ingestion                     | analytics.analyticsIngestionEndpoint             | Analytics Ingestion Endpoint |
| api-gateway                             | gateway.apiGatewayEndpoint                       | API Gateway Endpoint         |
| apic-gw-service                         | gateway.gatewayServiceEndpoint                   | Gateway Service Endpoint     |

Important: Endpoints cannot change with upgrade or restore procedures.

11. Update the configuration values for the chart. Refer to the README file and the tool tips that are provided on the fields for additional information about the correct settings.

12. Perform the following actions to configure backup and restore credentials after installation or upgrade:

- a. Execute the following command:

```
kubectl edit apiconnectcluster release_name-apic-cluster --namespace release_namespace
```

- b. Ensure the following parameters are present and correct under the "settings" section of the Management subsystem specification:

The parameters are as follows:

| Parameter                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cassandra-backup-protocol</code>  | The backup protocol. Specify one of the following values: <ul style="list-style-type: none"> <li><code>sftp</code> - for secure file transfer protocol</li> <li><code>objstore</code> - for S3 compatible object storage</li> </ul> Note: <ul style="list-style-type: none"> <li>For the management subsystem, IBM Cloud and Amazon Web Services (AWS) are supported S3 object store providers.</li> <li>The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>cassandra-backup-path</code>      | The full path to the directory where the backup files will be stored. This must point to a directory on the backup server. For object storage ( <code>objstore</code> ), the path can be set to the <code>bucket</code> value or the <code>bucket/subfolder</code> value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>cassandra-backup-host</code>      | The fully qualified domain name of the backup server. Ensure that the Kubernetes nodes can access this host. If using object store, enter <code>Endpoint/Region</code> . (The <code>/</code> character between the endpoint and region are required for this setting.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>cassandra-backup-port</code>      | The port for the protocol to connect to the <code>cassandra-backup-host</code> . The backup port is not required for object storage.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>cassandra-backup-schedule</code>  | Cron like schedule for performing automatic backups. The format for the schedule is: <ul style="list-style-type: none"> <li><code>*****</code></li> <li><code>-----</code></li> <li><code>     </code></li> <li><code>     +-----</code> day of week (0 - 6) (Sunday=0)</li> <li><code>   +-----</code> month (1 - 12)</li> <li><code>  +-----</code> day of month (1 - 31)</li> <li><code> +-----</code> hour (0 - 23)</li> <li><code>+-----</code> min (0 - 59)</li> </ul> The backup schedule defaults to <code>0 0 * * *</code> . This means a backup is run every day at midnight and minute zero. The timezone for backups is UTC.  When you configure a host, if you do not specify a value for <code>cassandra-backup-schedule</code> , the default backup schedule is automatically set. Note that the default backup schedule is not set, and scheduled backups not enabled, until host configuration is completed.  Note: Cassandra <code>repair</code> cron schedule is set to <code>00 1 * * 0,2,4,6</code> . This means the repair runs at 01:00 on Sunday, Tuesday, Thursday, and Saturday. By default, the Cassandra <code>backup</code> cron schedule should not run within one hour of the repair cron schedule. Please make sure to modify the current backup configuration as needed. If backups and repairs run at the same time, backup processes can fail intermittently. |
| <code>cassandra-backup-auth-user</code> | The username for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Key ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>cassandra-backup-auth-pass</code> | The password for the server specified in <code>cassandra-backup-host</code> . If using object store, this would be the S3 Secret Access Key parameter. The password will be stored in Base64 encoded format. For example: <pre>apicup subsys set mgmt cassandra-backup-auth-pass '&lt;password&gt;'</pre> Note that you cannot use the <code>`</code> sign to assign the password to <code>cassandra-backup-auth-pass</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

For example:

```
cassandra-backup-protocol: sftp
cassandra-backup-host: mybackuphost.com
cassandra-backup-port: 22
cassandra-backup-path: /backups
cassandra-backup-schedule: 0 0 * * *
cassandra-backup-auth-user: myusername
cassandra-backup-auth-pass: mypassword
```

c. Ensure the following parameters are present and correct under the **settings** section of the Portal subsystem specification:

| Parameter                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>site-backup-host</code>      | The fully qualified domain name of the backup server, in lowercase only. Ensure that the Kubernetes nodes can access this host. If using object storage, enter <code>Endpoint/Region</code> . (The <code>/</code> character between the endpoint and region is required for the object storage setting.)                                                                                                                                                                                                                                                                                                                     |
| <code>site-backup-port</code>      | The port for the protocol to connect to the <code>site-backup-host</code> . Defaults to <code>22</code> if not explicitly set. The backup port is not required for object storage.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>site-backup-auth-user</code> | The user name for the server specified in <code>site-backup-host</code> . If using object storage, the user name is the S3 Secret Key ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>site-backup-auth-pass</code> | The password for the server specified in <code>site-backup-host</code> . If using object storage, the password is the S3 Secret Access Key parameter. The password is stored in Base64 encoded format, and must not be edited directly in the <code>apiconnect-up.yml</code> file.                                                                                                                                                                                                                                                                                                                                           |
| <code>site-backup-path</code>      | The full path to the directory where the backup files are stored. For object storage, the path can be set to the <code>bucket</code> value or the <code>bucket/subfolder</code> value.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>site-backup-protocol</code>  | The protocol that is used to communicate with your remote backup endpoint. Specify one of the following values: <ul style="list-style-type: none"> <li><code>sftp</code> - for secure file transfer protocol</li> <li><code>objstore</code> - for S3 compatible object storage</li> </ul> The default protocol is <code>sftp</code> . Note: The public certificate on the S3 storage provider must be signed by a known certificate authority that is trusted by API Connect. Use of an untrusted authority can cause the following error during backup upload: <code>x509: certificate signed by unknown authority</code> . |



| Parameter                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>site-backup-schedule</code> | <p>The schedule for how often automatic Portal backups are run. The format for the schedule is any valid cron string, as follows:</p> <pre>* * * * * - - - - -                   +----- day of week (0 - 6) (Sunday=0)       +----- month (1 - 12)     +----- day of month (1 - 31)   +----- hour (0 - 23) +----- min (0 - 59)</pre> <p>For example: <code>30 22 * * 1</code> performs backups at 10:30 pm on Mondays.<br/>The default backup schedule is <code>0 2 * * *</code> (runs every day at 2 am). The timezone for backups is UTC.</p> |

For example:

```
site-backup-host: mybackuphost.com
site-backup-port: 22
site-backup-auth-user: mybackupauthusername
site-backup-auth-pass: mybackupauthpassword
site-backup-path: /site-backups
site-backup-protocol: sftp
site-backup-schedule: "0 2 * * *"
```

d. Save the resource.

e. Wait for upgrade jobs to complete.

13. For DataPower Gateway version 2018.4.1.4 and higher, you may need to set the license version for use with the IBM License Metric Tool (ILMT). The license version must match the image that you are installing. If you are changing images, you may need to reset the license version to match your image. The options for the license version are: Developers, Production, and Non-Production. For more information about ILMT, see the [IBM License Metric Tool](#) documentation.

14. Export the certificates that are generated with Install Assist (`apicup`) by completing the following steps:

a. Get a shell to the `apiconnect-operator` pod by running the following command:

```
kubectl exec -it operator_pod_name -- /bin/bash
```

b. Generate the secrets by running the following command for each certificate:

```
kubectl create secret generic secret_name --from-file=apicup_secret_file -n namespace
```

15. Store the secrets from step 14, plus the `apiconnect-up.yml` file, in a secure and permanent location from where they can always be retrieved.

## What to do next

Continue your configuration by:

- Setting up the Cloud Manager, starting with [Accessing the cloud console user interface](#)
- To execute backup and restore commands, get a shell to the `apiconnect-operator` pod with

```
kubectl exec -it operator_pod_name -- /bin/bash
```

and then follow the instructions in the [Backing up and restoring](#) section.

Attention: Helm TLS certificates expire after 90 days. To refresh this certificate in the `apiconnect-operator` pod, delete the Helm TLS secret, follow steps 2 and 3 to regenerate the certificate and recreate the secret, and then restart the pod.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deploying with the Install Assist tool

You can deploy IBM® API Connect into any version of the IBM Cloud Private environment by using the Install Assist method.

### Before you begin

This procedure lists the steps to install using the Install Assist tool. If you want to install using the IBM Cloud Private catalog method, see [Deploying with the IBM Cloud Private catalog](#).

To deploy API Connect in the IBM Cloud Private environment by using the IBM Cloud Private Install Assist method, you must have the following items:

- The IBM Cloud Private environment installed on your server or servers. See [IBM® Cloud Private v3.1.0 documentation](#) in the IBM Cloud Private documentation for additional information about how to install IBM Cloud Private.
- The IBM Cloud Private CLI. See [Installing the IBM Cloud Private CLI](#) in the IBM Cloud Private documentation for information about installing the CLI.
- Authentication to your IBM Cloud Private Docker registry. See [Configuring authentication for the Docker CLI](#) for the required steps.
- A configured Helm CLI. See [Setting up the Helm CLI](#) for the procedure.
- Ensure you have supported software requirement versions. See [IBM API Connect Version 2018 software product compatibility requirements](#).

### About this task

IBM Cloud Private provides an environment based on Kubernetes and Docker where you can host API Connect on a local cluster. Members with access to the cluster can use the service by subscribing to it.

## Procedure

1. Obtain the appropriate installation files described in [First steps for installing API Connect: Upload files to registry](#).
2. Login to the CLI of your IBM Cloud Private cluster by entering the following command:  
`cloudctl login -a https://<cluster_ip>:8443` where `cluster_ip` is the IP address or fully-qualified URL and port of your IBM Cloud Private cluster environment.
3. Login to the cluster's Docker registry by entering the following command: `docker login cluster_ip:8500`
4. Follow the registry upload steps described in [First steps for installing API Connect: Upload files to registry](#). Substitute `<cluster_ip>:8500/<namespace>` for `<registry_host>`.
5. For IBM Cloud Private versions 2.1.0.2, and later: Create a wrapper script named `helm` in your `usr/local/bin` directory that refers to your Helm CLI and appends `--tls` to the end of the Helm commands that are run by Install Assist.  
An example of this wrapper script follows:

```
#!/bin/bash
/usr/local/bin/helm $@ --tls
```

Note: The `HELM_HOME` environment variable must be set to the location where `ca.pem`, `cert.pem`, and `key.pem` were created in step 3 of [Deploying with the IBM Cloud Private catalog](#).

6. If installing on IBM Cloud Private version 3.1.1 or later, a PodSecurityPolicy must be bound to the target namespace(s) prior to installation. Choose either a predefined PodSecurityPolicy or have your cluster administrator set up a custom PodSecurityPolicy:

- Predefined PodSecurityPolicy name: `ibm-anyuid-hostpath-psp`
- Custom PodSecurityPolicy definition:

```
apiVersion: extensions/v1beta1
kind: PodSecurityPolicy
metadata:
  annotations:
    kubernetes.io/description: "This policy allows pods to run with
      any UID and GID and any volume, including the host path.
      WARNING: This policy allows hostPath volumes.
      Use with caution."
  name: ibm-anyuid-hostpath-psp
spec:
  allowPrivilegeEscalation: true
  fsGroup:
    rule: RunAsAny
  requiredDropCapabilities:
  - MKNOD
  allowedCapabilities:
  - SETPCAP
  - AUDIT_WRITE
  - CHOWN
  - NET_RAW
  - DAC_OVERRIDE
  - FOWNER
  - FSETID
  - KILL
  - SETUID
  - SETGID
  - NET_BIND_SERVICE
  - SYS_CHROOT
  - SETFCAP
  runAsUser:
    rule: RunAsAny
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  volumes:
  - '*'
```

- Custom Cluster Role for the custom PodSecurityPolicy:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    name: ibm-anyuid-hostpath-clusterrole
rules:
- apiGroups:
  - extensions
  resourceNames:
  - ibm-anyuid-hostpath-psp
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

7. For DataPower Gateway version 2018.4.1.4 and higher, you may need to set the license version for use with the IBM License Metric Tool (ILMT). For a new installation, the license version must match the image that you are installing. If you are changing images or upgrading, you may need to reset the license version to match your image. For more information about ILMT, see the [IBM License Metric Tool](#) documentation.
  - a. To set the license, enter the following command when installing the Gateway subsystem: `apicup subsys set gwy license-version <license>` where `license` is one of `Developers`, `Production`, or `Nonproduction`.
8. Install API Connect using the instructions that are provided at [Deploying to a Kubernetes environment](#).

## What to do next

Continue your configuration by setting up the Cloud Manager, starting with [Accessing the cloud console user interface](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Migrating an IBM Cloud Private catalog install to apicup

You can migrate an IBM Cloud Private installation that was done through the ICP catalog to an Install Assist (**apicup**) installation.

### About this task

You can take migration steps and then complete an **apicup** installation. A sample migration script is provided at the end of this topic. The script shows how to do the migration tasks in Step 1 through Step 5 in the following procedure.

### Procedure

1. Create local directories for the **apicup** project and extra values.
2. Copy **apicup** files from operator pod to the local project directory.
3. Copy extra values from the operator pod to the local extra values directory.
4. Update **apiconnect-up.yml** extra values locations to reference the new extra values directory.
5. Scale apiconnect-operator StatefulSet to 0 replicas.
6. Run **apicup subsys list** to verify all subsystems are present and **apicup subsys get <subsystem name>** for any desired additional validation.
7. Follow [Deploying with the Install Assist tool](#) to upgrade.

Note: Depending on the values used for the current analytics and gateway subsystem installations, you may need to add **--no-verify true** to the **apicup subsys install** command.

8. When the subsystems have been successfully upgraded, delete *only the top-level ICP catalog helm release* with

```
helm delete --purge --no-hooks <icp catalog release name>
```

Important: You MUST specify **--no-hooks**. If you omit it, subsystems will be deleted.

Example migration script:

```
#!/bin/bash

APICONNECT_OPERATOR_POD=$(kubectl get pods | grep apiconnect-operator | awk '{print $1}')
APICONNECT_OPERATOR_STATEFULSET=$(kubectl get statefulsets | grep apiconnect-operator | awk '{print $1}')
PROJECT_DIRECTORY=$(pwd)/tmp-apiconnect-project
EXTRA_VALUES_DIRECTORY=$PROJECT_DIRECTORY/extra-values

# create project and extra values directories
mkdir $PROJECT_DIRECTORY

# create list of apicup files to copy
FILES=$(kubectl exec -it $APICONNECT_OPERATOR_POD ls | grep yml | tr -d '\r')

# copy apicup files
echo "copying apicup files to $PROJECT_DIRECTORY local project directory"
for f in $FILES; do
    kubectl cp $APICONNECT_OPERATOR_POD:/home/apic/$f $PROJECT_DIRECTORY/$f
done

# copy extra values
echo "copying extra values files to $EXTRA_VALUES_DIRECTORY"
kubectl exec $APICONNECT_OPERATOR_POD -- tar cf - "extra-values" | tar xf -
mv extra-values $PROJECT_DIRECTORY

# update location for extra values files
sed -i.bak "s#/home/apic/extra-values#$EXTRA_VALUES_DIRECTORY#g" $PROJECT_DIRECTORY/apiconnect-up.yml

# scale apiconnect-operator statefulset to 0 replicas to prevent it from making changes
kubectl scale --replicas=0 statefulset $APICONNECT_OPERATOR_STATEFULSET
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing the toolkit

You can install the toolkit that provides CLI commands, and the API Designer user interface, for IBM® API Connect.

### About this task

The toolkit is provided as executable files, so no actual installation is necessary, you just need to download the required compressed file and extract the contents.

There are two toolkit options available:

- CLI: provides a command line environment for working with IBM API Connect.
- CLI + LoopBack + Designer: provides a command line environment for working with IBM API Connect, including LoopBack® support, and the API Designer user interface.

To install the toolkit, download the compressed file that is appropriate for your chosen toolkit option and platform, then extract the contents to a chosen location on your local machine. The compressed file contains an executable file for running CLI commands and, if you choose the CLI + LoopBack + Designer option, an executable file for launching the API Designer user interface.

You can download the toolkit compressed file in either of the following ways:

- From IBM Fix Central.
- From either the Cloud Manager or API Manager user interface.

The following table identifies the name of the compressed file that you need to download, depending on your chosen toolkit option and platform:

Table 1. Toolkit file names, by option and platform

| Toolkit option            | Mac OS X                          | Linux®                              | Windows                               |
|---------------------------|-----------------------------------|-------------------------------------|---------------------------------------|
| CLI                       | toolkit-mac.zip                   | toolkit-linux.tgz                   | toolkit-windows.zip                   |
| CLI + LoopBack + Designer | toolkit-loopback-designer-mac.zip | toolkit-loopback-designer-linux.tgz | toolkit-loopback-designer-windows.zip |

## Procedure

To install and run the toolkit, complete the following steps:

1. Download the toolkit compressed file.

- To download the toolkit from IBM Fix Central, complete the following steps:
  - Open the [IBM Fix Central site](#) in your browser.
  - In the Product selector field, enter `API Connect`, then select IBM API Connect from the drop down list.
  - Select your installed 2018.x.y version from the Installed Version list, then click Continue. If you do not know your installed IBM API Connect version, contact your administrator.
  - In the Text field, enter `toolkit`, then click Continue.
  - Select the required file, as identified in [Table 1](#).
  - Note: When you download from IBM Fix Central, the release number is appended to the file name.
  - Click Continue, then follow the instructions to complete the download operation.
- To download the toolkit from either the Cloud Manager or API Manager user interface, complete the following steps:
  - Cloud Manager or API Manager user interface.
  - Select Help ( ? ) in the navigation.
  - Select Install API Connect CLI & API Designer.
  - Select CLI or CLI + LoopBack + Designer according to your preferred option.
  - Select your platform to download the toolkit compressed file.
  - Close the Install API Connect CLI & API Designer window.

2. Extract the contents of the toolkit compressed file to a folder of your choice.

The contents of the file depend on the your chosen toolkit option and platform, as follows:

Table 2. Toolkit compressed file contents, by option and platform

| Toolkit option            | Mac OS X                                                                            | Linux                      | Windows                          |
|---------------------------|-------------------------------------------------------------------------------------|----------------------------|----------------------------------|
| CLI                       | apic-slim                                                                           | apic-slim                  | apic-slim.exe                    |
| CLI + LoopBack + Designer | apic<br>api_designer-mac.zip: contains the API Designer user interface application. | apic<br>api_designer-linux | apic.exe<br>api_designer-win.exe |

The `apic-slim` or `apic-slim.exe` file is the CLI for IBM API Connect.

The `apic` or `apic.exe` file is the CLI for IBM API Connect including LoopBack support.

Tip: If you are using the CLI option, then if you rename the `apic-slim` file to `apic`, or the `apic-slim.exe` file to `apic.exe`, you can run the CLI commands exactly as documented, copy and paste sample commands from the documentation, and use any command scripts as-is if you later move to the CLI + LoopBack + Designer option.

The `api_designer-platform` file is the API Designer user interface application for the specified platform.

3. Run the CLI.

- For the Mac OS X or Linux platforms, complete the following steps:
  - Open a terminal instance and navigate to the folder where you extracted the contents of the toolkit compressed file.
  - Make the CLI file an executable file by entering the following command:

```
chmod +x download_name
```

Where `download_name` is the name of the toolkit file that you downloaded, either `apic` or `apic-slim`.

- Run CLI commands as follows:

```
./apic command_name_and_parameters
```

or

```
./apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

- For the Windows platform, complete the following steps:
  - Open a terminal window and navigate to the folder where you extracted the contents of the toolkit compressed file.
  - Run CLI commands as follows:

```
apic command_name_and_parameters
```

or

```
apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

Tip: Add the folder location of your CLI file to your PATH variable so that you can run CLI commands from anywhere in your file system.

4. Launch the API Designer user interface by running the application from the location to which you extracted the contents of the toolkit compressed file.

Note:

- To uninstall the API Designer application on a Windows platform with a non Administrator account, complete the following steps:
  - In Windows File Explorer, navigate to the USER\_HOME\AppData\Local\Programs\api-designer folder.
  - Run the **Uninstall API Designer application** application. Do **not** use the **Add or remove programs** window.
- To uninstall the API Designer application on a Windows platform with an Administrator account, you can either run the **Uninstall API Designer application** application, or you can use the **Add or remove programs** window.

## Results

---

The IBM API Connect toolkit CLI and, if selected, the API Designer user interface application are installed on your local system.

For information on using the API Designer user interface, see [Developing your APIs and applications](#).

For information on using the toolkit CLI, see [Using the developer toolkit command-line tool](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Upgrading with IBM Cloud Private Catalog

You can update an API Connect deployment with the IBM® Cloud Private Catalog.

Use the instructions for your version of API Connect

- You can upgrade API Connect Version 2018.3.7 to API Connect Version 2018.4.1 with the IBM Private Cloud Catalog. See [Upgrading Version 2018.3.7 with the IBM Private Cloud Catalog](#).
- You can upgrade from API Connect Version 2018.4.1 to a later version of API Connect with the IBM Cloud Private Catalog. See [Upgrading from Version 2018.4.1 with the IBM Cloud Private Catalog](#).
- [Upgrading from Version 2018.4.1 with the IBM Cloud Private Catalog](#)  
You can upgrade from API Connect Version 2018.4.1 to a later version of API Connect with the IBM Cloud Private Catalog.
- [Upgrading Version 2018.3.7 with the IBM Private Cloud Catalog](#)  
You can upgrade API Connect Version 2018.3.7 to API Connect Version 2018.4.1 with the IBM Private Cloud Catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Upgrading from Version 2018.4.1 with the IBM Cloud Private Catalog

You can upgrade from API Connect Version 2018.4.1 to a later version of API Connect with the IBM Cloud Private Catalog.

### Before you begin

---

Warning: If you are upgrading from IBM® API Connect Version 2018.4.1.0 or 2018.4.1.1 to a later release and you want to retain existing analytics data, you must take an analytics backup before upgrade.

Note: For installation and upgrade prerequisites, configuration instructions, and changes between releases, refer to the Helm chart README and Release Notes in the IBM Cloud Private catalog entry for API Connect.

### Procedure

---

1. Export certificate secrets. To do so, get a shell to the `apiconnect-operator` pod with:

```
kubectl exec -it <operator pod name> -- /bin/bash
```

and run the following command for each .yml file in the working directory:

```
kubectl create secret generic <secret name> --from-file=<apicup secret file> -n <namespace>
```

2. Specify each secret name for its respective subsystem during upgrade. The common certificates secret should be specified in the `global` values configuration.  
3. Follow the upgrade procedures for a Kubernetes deployment. See [Upgrading API Connect in a Kubernetes environment](#).

### What to do next

---

For DataPower Gateway version 2018.4.1.4 and later releases, you may need to set the license version for use with the IBM License Metric Tool (ILMT). When using the default gateway image, the license version must match the image that you are installing. If you are changing images or upgrading, you may need to reset the license version to match your image. See [Deploying to an IBM Cloud Private environment](#). For more information about ILMT, see the [IBM License Metric Tool](#) documentation.

Warning: An IBM API Connect deployment that has been upgraded from Version 2018.4.1.0 or 2018.4.1.1 to a later release requires a new installation of the analytics subsystem. After upgrading, run the following commands to allow a new analytics subsystem release be installed:

```
kubectl delete deployments, statefulsets, secrets, jobs, pvc -l release=analytics_subsystem_release
helm delete --purge analytics_subsystem_release
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Upgrading Version 2018.3.7 with the IBM Private Cloud Catalog

You can upgrade API Connect Version 2018.3.7 to API Connect Version 2018.4.1 with the IBM Private Cloud Catalog.

### About this task

You must use the appropriate `apicup` commands to manage certificates and migrate them to `.yaml` files.

### Procedure

1. Download Install Assist (`apicup`) for API Connect v2018.3.7
2. Create a project with `apicup` called `<release_name> -apic-cluster`, where `<release_name>` is the name of the v2018.3.7 IBM Cloud Private Helm release, and set the endpoints to match those of the release.
3. For each certificate type of the installed release (common, analytics, gateway, management, and portal), write the contents of `kubectl get secret <cert_type>-generated-certs` to your local filesystem.
4. For each certificate type of the installed release, create a folder called `<cert_type>-generated-certs`, and for each data entry in each secret obtained in step 3, write the base64-decoded contents to a file whose name matches the entry's key. Copy the folder to the project directory created in step 2.
5. Download Install Assist for API Connect v2018.4.x.
6. Within the project directory create in step 2, run an Install Assist command, such as `apicup subsys get management`. This will migrate the certificates to `.yaml` files.
7. For each file generated in the previous step run:

```
kubectl create secret generic <secret name> --from-file=<apicup secret file> -n <namespace>
```
8. Specify each secret name for its respective subsystem during upgrade. The common certificates secret should be specified in the `global` values configuration.

### What to do next

When upgrading API Connect Version 2018.3.7 to Version 2018.4.1 or later, you should also upgrade DataPower to Version 2018.4.1.x. If you upgrade DataPower from v2018.3.x to v2018.4.x, you must resync the management configuration in order to reset DataPower. See [rapic\\_gw\\_reconfig\\_dyn.html#reference\\_pxx\\_nxf\\_5fb](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Backing up and restoring an IBM Cloud Private catalog deployment

You can back up and restore the API Connect Management subsystem and Portal subsystem on IBM Cloud Private.

Note:

- Backups are intended for recovery of the Management and Portal subsystems onto the same deployment from which they were taken, or onto a new replacement installation in the same environment for disaster recovery. The same environment means the same network configuration and project directory as the original installation. Backups are not designed as a means for migrating an API Connect deployment from one environment to another.
- You must back up both the Management and Portal subsystems at the same time, to ensure synchronicity across the services.
- If you have to perform a restore, you must complete the restoration of the Management Service first, and then immediately restore the Developer Portal. The backups of the Management and Portal must be taken at the same time to ensure that the Portal sites are consistent with Management database.
- [Backing up and restoring the management database on IBM Cloud Private](#)  
You can back up and restore the API Connect management subsystem on IBM Cloud Private.
- [Backing up and restoring the Developer Portal on IBM Cloud Private](#)  
You can back up and restore the API Connect Developer Portal subsystem on IBM Cloud Private.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Backing up and restoring the management database on IBM Cloud Private

You can back up and restore the API Connect management subsystem on IBM Cloud Private.

### Before you begin

---

Ensure backup parameters are configured. See Step 12 in [Deploying with the IBM Cloud Private catalog](#).

### Procedure

---

- Backing up the management database
  1. Get a shell to the apiconnect-operator pod:

```
kubectl exec -it <operator pod name> -- /bin/bash
```
  2. Edit the `apiconnect-up.yml` file to ensure the backup parameters match that of the `apiconnectcluster` resource edited in Step 12 in [Deploying with the IBM Cloud Private catalog](#).
  3. To execute an on-demand backup for the management subsystem, run the following command:

```
apicup subsys exec management backup
```
- Restoring the management database
  1. To list management subsystem backups, run the following command:

```
apicup subsys exec management list-backups
```
  2. To restore a management subsystem backup, run the following command:

```
apicup subsys exec management restore <backupID>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Backing up and restoring the Developer Portal on IBM Cloud Private

You can back up and restore the API Connect Developer Portal subsystem on IBM Cloud Private.

### Before you begin

---

Ensure backup parameters are configured. See Step 12 in [Deploying with the IBM Cloud Private catalog](#).

### Procedure

---

- Backing up the Developer Portal
  1. Get a shell to the apiconnect-operator pod:

```
kubectl exec -it <operator pod name> -- /bin/bash
```
  2. Edit the `apiconnect-up.yml` file to ensure the backup parameters match that of the `apiconnectcluster` resource edited in Step 12 in [Deploying with the IBM Cloud Private catalog](#).
  3. To execute an on-demand backup for the portal system configuration, run the following commands:
    - To backup the Portal system configuration (and not the sites), run the following command:

```
apicup subsys exec portal backup-system
```
    - To backup a specific Portal site or all sites, run the following command:

```
apicup subsys exec portal backup-site arg
```

where `arg` is a specific site UUID or URL, or to backup all sites use the argument `installed`.
    - To backup the entire Portal service (the system and all installed sites), run the following command:

```
apicup subsys exec portal backup
```
- Restoring the Developer Portal
  1. To list the portal subsystem system and site backups, run the following command:

```
apicup subsys exec portal list-backups remote
```
  2. To restore a Developer Portal subsystem backup, select from the following commands:
    - To restore your Portal subsystem, run the following command:

```
apicup subsys exec portal restore-all run backup_from
```

where `backup_from` can be `now`, so the latest backup file that is available is used. Or you can specify a timestamp in the format of `YYYYMMDD.HHMMSS` to retrieve the nearest backup file to a specified time, searching backwards from the timestamp given. This command executes the portal restore

process, which downloads the system backup and all of the site backups from the remote server, and installs them within the Portal stack. This process will then restore the system configuration from the found backup, and restore all of the sites.

Note: From IBM API Connect Version 2018.4.1.17, the timestamp format changed to **YYYYMMDD.HHMMSS**. For Version 2018.4.1.16 and earlier, the timestamp format is **YYYY-MM-DD HH:MM:SS**.

- To restore just the Portal system configuration, run the following command:

```
apicup subsys exec portal restore-system arg
```

where **arg** can be the system backup tgz file name, to restore the system by using the specified backup file, or use the argument **latest** to restore the system by using the latest backup file on the remote server. Note that if Portal system configuration information exists on the current stack, running this command will overwrite that configuration.

- To restore just the Portal sites, run the following command:

```
apicup subsys exec portal restore-site arg
```

where **arg** can be the site backup tgz file name or URL, to restore a particular site by using the specified backup file (or the latest backup file found if using a URL). Or you can use the argument **all** to restore all of the Portal sites that have backup files on the remote server. Note that if any of the backed up sites are already installed on the current stack, then they are reinstalled by using the backup (the sites are overwritten by the backup).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## OpenShift

You can install API Connect on OpenShift.

- [Installing API Connect on OpenShift](#)  
Installing API Connect in an OpenShift environment is a Kubernetes installation with additional steps.
- [Disabling the Analytics subsystem on OpenShift](#)  
During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing API Connect on OpenShiftIBM®

Installing API Connect in an OpenShift environment is a Kubernetes installation with additional steps.

### Before you begin

Before starting the installation into an OpenShift runtime environment, consider the following information:

- During installation, OpenShift requires the **ingress-type** to be set to **route** for each subsystem, for example, **apicup subsys set SUB\_SYS ingress-type route**.
- If using an OpenShift registry, execute the **oc new-project apiconnect** command to create the apiconnect project prior to uploading the images using the **apicup registry-upload** command.
- SSH access must be disabled for the Gateway. Note that it is disabled by default.
- OpenShift routes do not redirect URLs for the api-manager-ui and cloud-admin-ui endpoints. To open the API Manager and Cloud Manager UIs in an OpenShift environment, the complete URLs for the API Manager and Cloud Manager UIs must be entered in the browser as follows:
  - API Manager user interface: **https://api\_manager\_ui\_endpoint/manager**
  - Cloud Manager user interface: **https://cloud\_admin\_ui\_endpoint/admin**
- On OpenShift, the API Connect management subsystem sets resource requests but does not set resource limits. Limits are not specified so that pods can request the necessary resources. If you choose to modify OpenShift to set resource limits, such as for CPU or memory, you must set the same value in the API Connect extra values YAML file. Ensure that the values in the YAML file match the values you set in OpenShift.  
Note: Setting of resource requests and limits on OpenShift is not supported for the API Connect Analytics subsystem.

### Procedure

1. Create namespaces or a namespace for API Connect. You can create separate namespaces for each subsystem, or place all subsystems in a single namespace. Following is an example for separate namespaces for each subsystem:

```
oc create ns apic-management
oc create ns apic-gateway
oc create ns apic-analytics
oc create ns apic-portal
```

Following is an example for one namespace containing all subsystems:



```
oc create ns apiconnect
```

Following is an example for Management, Portal and Analytics in a one namespace and Gateway in a separate namespace:

```
oc create ns apic-mpa
oc create ns apic-gateway
```

2. Create a registry secret in the namespace where the subsystems will be deployed.

```
oc apply -f <path-to-registry-secret>
```

This secret is referenced in the APICUP installer commands for each subsystem, for example:

```
apicup subsys set SUB_SYS
registry-secret=<registrySecret>
```

Refer to the installation instructions for each subsystem, for example, [Installing the Management subsystem into a Kubernetes environment](#).

3. Install and configure Tiller.

- a. Configure Tiller either globally on the cluster or scoped to a single namespace.

The following example configures Tiller globally on the cluster:

```
oc create ns tiller
export TILLER_NAMESPACE=tiller
oc process -f https://github.com/openshift/origin/raw/master/examples/helm/tiller-template.yaml -p
TILLER_NAMESPACE=${TILLER_NAMESPACE} -p HELM_VERSION=v2.10.0 | oc create -n "${TILLER_NAMESPACE}" -f -
oc create clusterrolebinding tiller-binding --clusterrole=cluster-admin --user=system:serviceaccount:tiller:tiller
```

The following example installs Tiller scoped to a single namespace, where `$(NAMESPACE)` is the namespace for API Connect or the namespace for a specific subsystem:

```
export TILLER_NAMESPACE=$(NAMESPACE)
oc process -f https://github.com/openshift/origin/raw/master/examples/helm/tiller-template.yaml -p
TILLER_NAMESPACE=${TILLER_NAMESPACE} -p HELM_VERSION=v2.10.0 | oc create -n "${TILLER_NAMESPACE}" -f -
oc create rolebinding $(NAMESPACE)-tiller-cluster-admin --clusterrole=cluster-admin --
serviceaccount=$(NAMESPACE):tiller --namespace=$(NAMESPACE)
```

Note: When Tiller is scoped to a single namespace, with OpenShift 4.3, the Tiller account might need additional permissions. Permissions must be updated when you see the following errors:

- Running `apicup subsys install mgmt --debug` returns:

```
no permissions to get CRDs
```

- Running `oc get crd` from the Tiller account returns:

```
Error from server (Forbidden):
customresourcedefinitions.apiextensions.k8s.io is forbidden:
User "system:serviceaccount:apiconnect:tiller" cannot list resource "customresourcedefinitions"
in API group "apiextensions.k8s.io" at the cluster scope
```

To resolve the problem, assign additional permissions to the Tiller account:

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: helm-clusterrole
rules:
- apiGroups:
  - ""
  - rbac.authorization.k8s.io
  - roles.rbac.authorization.k8s.io
  - authorization.k8s.io
  - apiextensions.k8s.io
  resources:
  - "clusterroles"
  - "clusterrolebindings"
  - "customresourcedefinitions"
  verbs: ["create", "get", "escalate"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["list", "get"]
```

---

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: helm-clusterrolebinding
roleRef:
  kind: ClusterRole
  apiGroup: rbac.authorization.k8s.io
  name: helm-clusterrole
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: {TILLER_APIC_NAMESPACE}
```

Replace `{TILLER_APIC_NAMESPACE}` with the namespace where both Tiller and API Connect are to be deployed.

- b. Update the deployment's Tiller YAML:

Update the YAML. Locate the following statement:

```
image: 'gcr.io/kubernetes-helm/tiller:v2.10.0'
```

And modify it to match the following example:

```
image: 'ghcr.io/helm/tiller:v2.10.0'
```

4. Assign Security Context Constraint (SCC) permissions to Service Accounts. The commands will vary depending upon the namespace configuration and permission level. Following are examples:

Attention: If you plan to install the Analytics subsystem so that data is stored locally (that is, `ingestion-only` is set to `false`, then you must install Analytics into a separate namespace where you can assign the `privileged` level of SCC access to it. See example 2 for more information.

**Example 1:** To assign anyuid access to all Service Accounts in a namespace, enter this command:

```
oc adm policy add-scc-to-group anyuid system:serviceaccounts:<namespace>
```

where `<namespace>` is the API Connect namespace, for example, `apiconnect`, that contains all four subsystems (Management, Gateway, Portal, Analytics). This command assigns anyuid permissions to all subsystems contained in the namespace. For a configuration with all four subsystems in one namespace, anyuid permissions are required because the Gateway requires anyuid.

Note: Use this approach when you install Analytics using the `ingestion-only` configuration.

**Example 2:** To assign anyuid access to the Management, Gateway, and Portal Service Accounts in one namespace, and assign privileged access to the Analytics Service Account in another namespace, enter the following commands:

```
#Assign anyuid access to the Management, Gateway, and Portal Service Accounts:
oc adm policy add-scc-to-group anyuid system:serviceaccounts:<namespace>
```

where `<namespace>` is the API Connect namespace (for example, `apiconnect`) that contains the Management, Gateway, and Portal subsystems. This command assigns anyuid permissions to all subsystems contained in the namespace, so be sure to set up a separate namespace for Analytics.

```
#Assign privileged access to the Analytics Service Account in a different namespace:
oc adm policy add-scc-to-group privileged system:serviceaccounts:<Analytics_namespace>
```

where `<Analytics_namespace>` is the namespace (for example, `analytics`) for the Analytics subsystem.

Note: Use this approach if you will install Analytics using with `ingestion-only` set to `false`. The privileged access level provides much greater access to resources in a namespace, and should only be used where necessary. Since only the Analytics subsystem requires this access level, you should assign the anyuid access level to the remaining subsystems (which requires them to be hosted in a different namespace).

**Example 3:** To assign non-root access to all Service Accounts except the Gateway, create a namespace that contains the Management, Portal and Analytics subsystems. Then create a namespace that contains only the Gateway subsystem. Enter these commands:

```
#Set nonroot permissions for all subsystems (Management, Portal, and Analytics) in the namespace
oc adm policy add-scc-to-group nonroot system:serviceaccounts:<namespace>
```

where `<namespace>` is the namespace that contains the Management, Portal and Analytics subsystems. This assigns nonroot permissions to all subsystems contained in the namespace.

```
#Set anyuid permission for the gateway subsystem
oc adm policy add-scc-to-group anyuid system:serviceaccounts:<namespace>
```

where `<namespace>` is the namespace that contains the gateway subsystem. This assigns anyuid permissions to the gateway.

5. For the DataPower® WebUI setup, depending on whether the default SCC which is installed as part of OpenShift has been modified, assign anyuid access to DAC\_OVERRIDE, SETUID, SETGID, and SYS\_CHGROOT. See the following examples.

**Example 1:** To assign anyuid access to DAC\_OVERRIDE for the gateway subsystem, enter this command:

```
oc adm policy add-scc-to-group anyuid system:dac_override:<namespace>
```

where `<namespace>` is the namespace that contains the gateway subsystem.

**Example 2:** To assign anyuid access to SETUID for the gateway subsystem, enter this command:

```
oc adm policy add-scc-to-group anyuid system:setuid:<namespace>
```

where `<namespace>` is the namespace that contains the gateway subsystem.

**Example 3:** To assign anyuid access to SETGID for the gateway subsystem, enter this command:

```
oc adm policy add-scc-to-group anyuid system:setgid:<namespace>
```

where `<namespace>` is the namespace that contains the gateway subsystem.

**Example 4:** To assign anyuid access to SYS\_CHGROOT for the gateway subsystem, enter this command:

```
oc adm policy add-scc-to-group anyuid system:sys_chgroot:<namespace>
```

where `<namespace>` is the namespace that contains the gateway subsystem.

Note: If you encounter problems assigning Security Context Constraints (SCC) in order to enable the WebUI for DataPower, see the recommendation to create a new SCC: [https://www.ibm.com/mysupport/s/question/0D50z00006RZGPB/apic-2018-gateway-service-error-when-enabling-datapower-web-ui?language=en\\_US](https://www.ibm.com/mysupport/s/question/0D50z00006RZGPB/apic-2018-gateway-service-error-when-enabling-datapower-web-ui?language=en_US).

6. Install the subsystems following the instructions for installing API Connect on Kubernetes. Set the `ingress-type` to `route` for each subsystem, for example, `apicup subsys set SUB_SYS ingress-type route`. See [Installing and upgrading on Kubernetes](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Disabling the Analytics subsystem on OpenShift

During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

## About this task

Disabling the Analytics subsystem stops the collection and storage of data as well as making the subsystem unavailable. When you disable Analytics, there will be no data in either the API Manager Dashboards or third-party offloads.

## Procedure

1. Back up your Analytics data as explained in [Backing up and restoring the analytics database](#).

2. Disable shard allocation for storage.

Disabling shard allocation is optional, but is recommended because it prevents new shards from being created for replication when a node is unavailable, and helps avoid corruption issues in full system restarts. Disable shard allocation for storage by running the following command:

```
oc -n namespace exec -it storage-master|shared_pod -- curl_es _cluster/settings -XPUT -d '{"persistent":{"cluster.routing.allocation.enable":"none"}}'
```

where:

- `namespace` is the namespace where the Analytics subsystem is installed.
- `storage-master|shared_pod` is the name of any storage-master or storage-shared pod. You can get the name of your storage pods by running the following command and substituting in the namespace where Analytics is installed:



```
oc -n namespace get po
```

When you disable shard allocation, the response looks like the following example:

```
{"acknowledged":true,"persistent":{"cluster":{"routing":{"allocation":{"enable":"none"}}},"transient":{}}
```

3. Unassociate the Analytics service from all gateway services.




- a. In Cloud Manager, click  Topology.
- b. In the section for the Availability Zone that contains the Analytics service, locate the Gateway service that the Analytics service is associated with.
- c. Click  and select Unassociate analytics service.

All analytics collection will be disabled and there will be no data in either the API Manager Dashboards or third-party offloads. For more information, see [Associating an analytics service with a gateway service](#).

4. Unregister the Analytics service from Cloud Manager:



- a. In Cloud Manager, click  Topology.
- b. In the section for the Availability Zone that contains the Analytics service, locate the Analytics service and click Delete.

For more information, see [Registering an analytics service](#).

5. Perform a synced flush of storage.

Flushing the storage is optional, but is recommended because it helps to avoid losing the data that was not yet written to disk. This step flushes all current index operations synchronously and attempts to write everything that is in flux to disk. Perform a synced flush of storage by running the following command:

```
oc -n namespace exec -it storage-master|shared_pod -- curl_es _flush/synced -XPOST
```

When you flush storage, the response looks like the following example:

```
{"shards":{"total":33,"successful":13,"failed":0},"export-status":{"total":1,"successful":1,"failed":0},".apic-config":{"total":1,"successful":1,"failed":0},".kibana-6":{"total":1,"successful":1,"failed":0},"apic-api-2020.06.19-000002":{"total":15,"successful":5,"failed":0},"apic-api-2020.06.18-1":{"total":15,"successful":5,"failed":0}}
```

The operation often fails, and it is safe to rerun it multiple times until it passes and there are all "failed" counts are zero. If it continues to fail after running multiple attempts over the span of a couple minutes, you can proceed to the next step.

6. Run the following two commands to configure analytics for ingestion-only:

```
$ apicup subsys set <analytics-subsystem-name> ingestion-only=true
```

```
$ apicup subsys install <analytics-subsystem-name>
```

7. Scale the analytics-ingestion deployment down to 0 replicas by completing the following steps.

- a. Get a list of all deployments so that you can find the name of the analytics-ingestion deployment:

```
$ oc get deployments -n <namespace>
```

- b. Scale the analytics-ingestion deployment down to 0 replicas:

```
$ oc scale deployment/<analytics-ingestion-deployment-name> --replicas=0 -n <namespace>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## IBM Cloud Pak for Integration

IBM API Connect provides the API management capability in IBM Cloud Pak for Integration.

## About this task

For information on deploying or upgrading API Connect as a component of Cloud Pak for Integration, see the *API management capability deployment* section in the [Cloud Pak for Integration documentation](#).

- [Backing up and restoring on Cloud Pak for Integration](#)  
Back up and restore API Management data in IBM Cloud Pak for Integration.
- [Disabling the Analytics subsystem on Cloud Pak for Integration](#)  
During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Backing up and restoring on Cloud Pak for Integration

Back up and restore API Management data in IBM Cloud Pak for Integration.

### About this task

If you deployed a version of the API Management capability in IBM Cloud Pak for Integration that uses API Connect V2018 FP13-ifix1, you can back up and restore your data.

Restriction: The following instructions apply only to API Connect V2018 FP13-ifix1.

- [CP4I: Backing up data in V2018 FP13-ifix1](#)
- [CP4I: Recovering the v2018 FP13-ifix1 deployment](#)  
Note: Although this topic discusses recovering the deployment after a disaster, it includes information on restoring backed up data (see steps 5 and later).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Disabling the Analytics subsystem on Cloud Pak for Integration

During maintenance of an API Connect cluster, you can shut down the Analytics subsystem by disabling the client, ingestion, and storage microservices so that those processes are not running and there is no way to reach them.

### About this task

Disabling the Analytics subsystem stops the collection and storage of data as well as making the subsystem unavailable. When you disable Analytics, there will be no data in either the API Manager Dashboards or third-party offloads.

## Procedure

1. Back up your Analytics data as explained in [Backing up and restoring the analytics database](#).
2. Disable shard allocation for storage.  
Disabling shard allocation is optional, but is recommended because it prevents new shards from being created for replication when a node is unavailable, and helps avoid corruption issues in full system restarts. Disable shard allocation for storage by running the following command:

```
oc -n namespace exec -it storage-master|shared_pod -- curl_es _cluster/settings -XPUT -d '{"persistent": {"cluster.routing.allocation.enable": "none"}}'
```

where:

- `namespace` is the namespace where the Analytics subsystem is installed.
- `storage-master|shared_pod` is the name of any storage-master or storage-shared pod. You can get the name of your storage pods by running the following command and substituting in the namespace where Analytics is installed:



```
oc -n namespace get po
```

When you disable shard allocation, the response looks like the following example:

```
{"acknowledged": true, "persistent": {"cluster": {"routing": {"allocation": {"enable": "none"}}}, "transient": {}}
```

3. Unassociate the Analytics service from all gateway services.




- a. In Cloud Manager, click  Topology.
- b. In the section for the Availability Zone that contains the Analytics service, locate the Gateway service that the Analytics service is associated with.
- c. Click  and select Unassociate analytics service.

All analytics collection will be disabled and there will be no data in either the API Manager Dashboards or third-party offloads. For more information, see [Associating an analytics service with a gateway service](#).

- Unregister the Analytics service from Cloud Manager:



- In Cloud Manager, click  Topology.
- In the section for the Availability Zone that contains the Analytics service, locate the Analytics service and click Delete.

For more information, see [Registering an analytics service](#).

- Perform a synced flush of storage.

Flushing the storage is optional, but is recommended because it helps to avoid losing the data that was not yet written to disk. This step flushes all current index operations synchronously and attempts to write everything that is in flux to disk. Perform a synced flush of storage by running the following command:

```
oc -n namespace exec -it storage-master|shared_pod -- curl -es _flush/synced -XPOST
```

When you flush storage, the response looks like the following example:

```
{ "_shards": {"total": 33, "successful": 13, "failed": 0}, ".export-status": {"total": 1, "successful": 1, "failed": 0}, ".apic-config": {"total": 1, "successful": 1, "failed": 0}, ".kibana-6": {"total": 1, "successful": 1, "failed": 0}, "apic-api-2020.06.19-000002": {"total": 15, "successful": 5, "failed": 0}, "apic-api-2020.06.18-1": {"total": 15, "successful": 5, "failed": 0}}
```

The operation often fails, and it is safe to rerun it multiple times until it passes and there are all "failed" counts are zero. If it continues to fail after running multiple attempts over the span of a couple minutes, you can proceed to the next step.

- Run the following two commands to configure analytics for ingestion-only:

```
$ apicup subsys set <analytics-subsystem-name> ingestion-only=true
```

```
$ apicup subsys install <analytics-subsystem-name>
```

- Scale the analytics-ingestion deployment down to 0 replicas by completing the following steps.

- Get a list of all deployments so that you can find the name of the analytics-ingestion deployment:

```
$ oc get deployments -n <namespace>
```

- Scale the analytics-ingestion deployment down to 0 replicas:

```
$ oc scale deployment/<analytics-ingestion-deployment-name> --replicas=0 -n <namespace>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Migrating a Version 5 deployment

The current migration path from v5 is to API Connect v10. You can get more information about migrating to v10 here:

[https://www.ibm.com/support/knowledgecenter/SSMNEJ\\_v10/com.ibm.apic.install.doc/migrating.html](https://www.ibm.com/support/knowledgecenter/SSMNEJ_v10/com.ibm.apic.install.doc/migrating.html).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using the API Connect operations command line interface

The IBM® API Connect v2018 **apicops** command line interface is targeted at Operations teams. It contains commands to check the health of the system and some commands to fix specific problems that might be encountered.

The **apicops** command line interface is in active development and the latest version is available to download from [Github.com](#). All suggestions, feedback, and bug reports are welcome - feel free to raise an issue in this repository. Please limit the issues to those specific to **apicops** itself - issues about API Connect cannot be accepted, they must be raised by using the normal IBM Support route.

Note: Support for **apicops** is for IBM API Connect v2018.4.1.6 and higher, only.

---

## Installing

Download the binary for your operating system from the latest release on [github.com](#) and rename it to be **apicops**. For more information, see [Releases - APIC Operations CLI](#).

Note: Linux and Mac variants require that you run **chmod +x** on the downloaded file before you can run it.

---

## Requirements

To run **apicops**, you need to have kubectl, or something that implements the same CLI as kubectl, such as oc (OpenShift client), installed locally. If you aren't using kubectl, then set the environment variable **APICOPS\_K8SCIENT** to the name of the Kubernetes client binary, such as oc.

If you are using the oc, then v4.1.x or higher is required, even if you are using v3.x of the OpenShift cluster.

Then, set the `KUBECONFIG` environment variable to point to your kubeconfig file and `apicops` picks it up from there.

```
$ export KUBECONFIG=/home/user/my.kubeconfig
$ apicops
```

If you are running inside an API Connect OVA file, then run `apicops` as root, `sudo -i`, and it automatically picks up the kubeconfig.

## Setting the target namespace

By default, the targeted namespace for the deployment is `default`. If your deployment uses an alternative namespace, then you need to set this value in the relevant context of your kubeconfig file. You can either edit your kubeconfig file, and add the namespace property with your value. Or, you can set the namespace for the current context by using the following command, where `<namespace>` is the value you want to use.

```
kubectl config set "contexts." `kubectl config current-context` ".namespace" <namespace>
```

You can view all contexts and their configured namespaces with the command:

```
kubectl config get-contexts
```

## Usage

```
$ apicops COMMAND
running command...
$ apicops (-v|--version|version)
apicops/0.1.46 linux-x64 node-v10.16.3
$ apicops --help [COMMAND]
USAGE
  $ apicops COMMAND
  ...
```

- [Identify Services State](#)

API Connect uses tasks to do actions such as synchronizing content between API Manager and the Gateways and Portals. When you are diagnosing some problems, it can be useful to determine what the state is of those tasks.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Identify Services State

API Connect uses tasks to do actions such as synchronizing content between API Manager and the Gateways and Portals. When you are diagnosing some problems, it can be useful to determine what the state is of those tasks.

Determining the state of the task can be done by using the `apicops`

`services:identify-state` command, which identifies the state of any gateway and portal services and returns any associated task IDs that are incomplete.

**USAGE**

```
$ apicops services:identify-state
```

**OPTIONS**

```
-e, --embellish  Output a table per service instead of single lines. In JSON mode beautify the JSON
-j, --json      Output as raw JSON instead of lines/tables
```

**ALIASES**

```
$ apicops iss
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring and managing your server environment

You configure and manage the servers that comprise your IBM® API Connect on-premises cloud by using the Cloud Manager user interface.

The Cloud Manager user interface is the part of IBM API Connect that enables a Cloud Administrator to define and manage the API Connect on-premises cloud.

You can use the Cloud Manager to define the API Connect cloud by performing the following tasks:

- Create Provider organizations and invite users to serve as the owner
- Create and manage user roles and role defaults
- Create availability zones for services
- Register the relevant servers that will provide the gateway, analytics, and portal services
- Associate an analytics service with a gateway to enable reports for API Events
- Configure resources for user authentication, TLS security, and OAuth providers and make the resources visible to all or selected provider organizations

- Connect to an existing SMTP mail server and edit templates for system-generated emails
- Set the default gateway service for catalogs
- [Activating your Cloud Manager user account](#)  
If you have been invited to be an administrator of your API Connect cloud, you must activate your account by using the activation link that was sent by another administrator. You can then access the Cloud Manager user interface.
- [Accessing the Cloud Manager user interface](#)  
How to navigate to and log in to the Cloud Manager user interface.
- [Defining your topology](#)  
To define your API Connect on-premises cloud, you define availability zones and register services within those zones to securely create, promote, and track APIs. The installation procedure provides initial configuration of an Availability Zone containing the Management service.
- [Managing authentication and security](#)  
Secure Cloud Manager and API Manager as well as your Catalogs with a user registry. Secure your APIs with OAuth. Create TLS profiles to ensure that information you share among web servers will not be stolen or tampered with.
- [Configuring the cloud settings](#)  
Before you can add a provider organization to your cloud, you must specify your cloud settings to configure an email server for notifications as well as user registry options and catalog defaults.
- [Administering provider organizations](#)  
Manage the provider organizations who have accounts to publish APIs in your API Connect cloud.
- [Administering members and roles](#)  
Cloud Administrators can add members and assign them roles to enable them to work in Cloud Manager. The Cloud and Topology Administrators can also create roles and role defaults. You can also delete users to prevent them from accessing Cloud Manager.
- [Monitoring the cloud](#)  
API Connect generates events to monitor the status of your cloud.
- [Changing your Cloud Manager password and profile information](#)  
You can change your Cloud Manager password and update your profile information.
- [Resolving login problems by increasing HTTP header size](#)  
You can resolve login problems for the Cloud Manager UI by increasing the maximum HTTP client header size.
- [Extending the Gateway server behavior](#)  
To support your enterprise requirements, you can extend the Gateway servers within IBM API Connect to provide extra enforcement behavior.
- [Cloud Manager Tutorials](#)  
Tutorials for using the Cloud Manager user interface in IBM API Connect.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Activating your Cloud Manager user account

If you have been invited to be an administrator of your API Connect cloud, you must activate your account by using the activation link that was sent by another administrator. You can then access the Cloud Manager user interface.

### Before you begin

A cloud administrator invited you to join API Connect .

### Procedure

1. Complete the following steps to activate your Cloud Manager account:

If the Identity Provider uses LDAP

An invitation email with an activation link is sent. Click the activation link, or paste it in a browser, to log in directly with your LDAP user credentials. Upon authentication, the API Connect user record is updated from the backend identity provider.

If the Identity Provider uses a local registry

An invitation email with an activation link is sent. Click the activation link or paste it in a browser. The activation link takes you to a sign up page where you enter your first name, last name and password. Passwords must have a minimum of 8 characters and contain characters from at least three of the four following categories:

- Uppercase letters
- Lowercase letters
- Numbers
- Special characters (for example: ! # \$ %)

Note:

- The email address that you enter on the sign up page must match the email address to which the invitation email was sent, otherwise the account activation fails.
- If you previously had an account that was removed, and you are being invited again, you must re-activate your account by using the Sign In option on the page, **not** by completing the registration form and using the Sign Up option; attempting to re-register will fail.

If the Identity Provider uses an authentication URL

An invitation email with an activation link is sent. Click the activation link or paste it in a browser. The activation link takes you to a sign up page where you enter your credentials, which then become your user name and password. Upon authentication, the API Connect user record is updated from the backend identity provider.

If multiple user registries are available for selection on the sign up page, then make sure that the correct registry for your Cloud Manager account is selected. You might need to ask your administrator which user registry is appropriate for your account.

For information on configuring user registries and making them available for Cloud Manager login, see [User registries overview](#) and [Selecting user registries for Cloud Manager and API Manager](#).

2. Click Sign up to complete your registration, then click Sign in to open the Cloud Manager login page.

---

## Results

You have created your API Connect administrator account.

---

## What to do next

Log in to the Cloud Manager user interface and, depending on your role, start to manage the cloud topology, and work with provider organizations and cloud resources. To access the Cloud Manager login page in the future, use the following URL:

`https://host/admin`

where *host* is the fully qualified host name or IP address of the Management server.

---

## Related concepts

- [Managing authentication and security](#)
- [Configuring the cloud settings](#)

---

## Related tasks

- [Accessing the Cloud Manager user interface](#)
- [Defining your topology](#)
- [Administering provider organizations](#)
- [Administering members and roles](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Accessing the Cloud Manager user interface

How to navigate to and log in to the Cloud Manager user interface.

---

## Before you begin

To open the Cloud Manager user interface, enter the URL for the <cloud-admin-ui> endpoint entered during installation, followed by `/admin`. The <cloud-admin-ui> endpoint is configured in the Management subsystem using the following command: `apicup subsys set mgmt cloud-admin-ui <endpoint>`.

The first time that you access the Cloud Manager user interface, you enter `admin` for the user name and `7iron-hide` for the password. You will be prompted to change the Cloud Administrator password and email address.

---

## Procedure

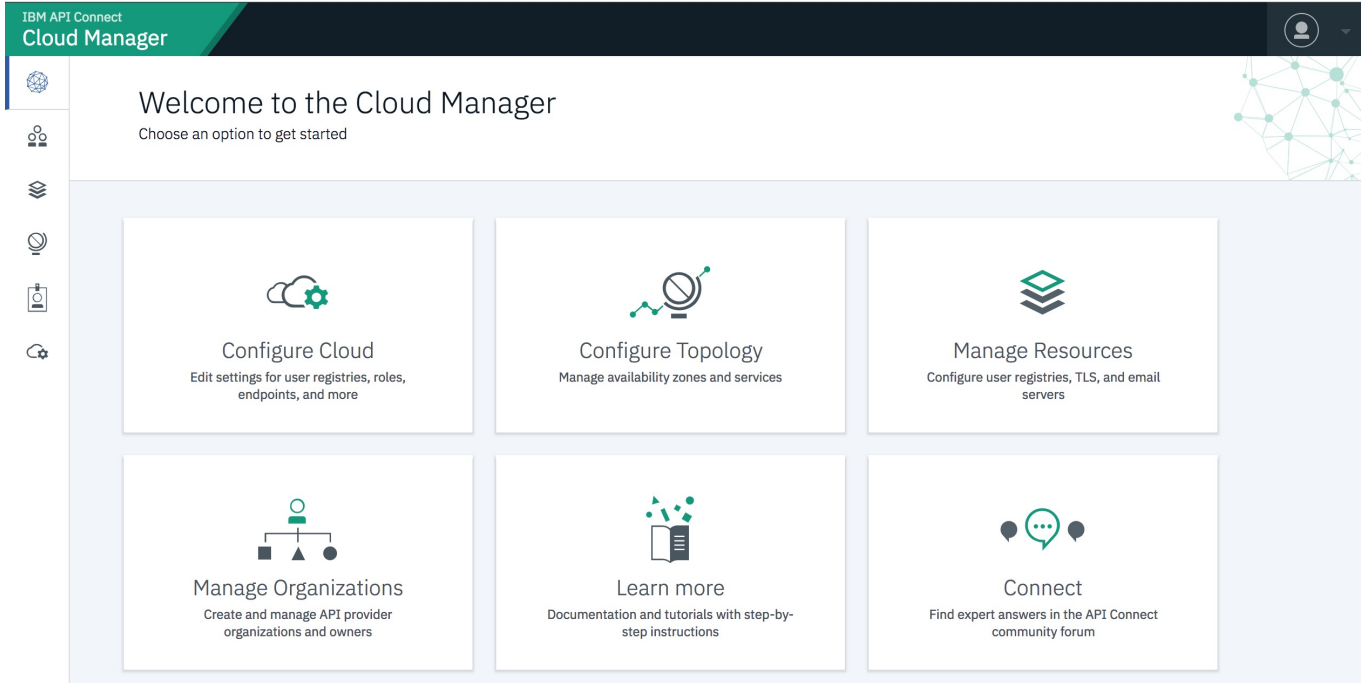
1. Open a browser and enter the URL for the cloud-admin-ui endpoint followed by `/admin`. Note that this is secure connection using HTTPS, for example:  
`https://<cloud-admin-ui>/admin`.
2. Enter the Cloud Administrator user name and password. If you are logging in for the first time, enter `admin` for the user name and `7iron-hide` for the password. Otherwise, enter `admin` for the user name and `<your password>` for the password.
3. If your cloud contains multiple user registries, as configured on the Settings\_>User Registries screen, then choose the user registry that contains your login credentials. You may need to ask your administrator which user registry is appropriate for your account.
4. Click Sign In.  
Important: The first time that you access the cloud console, you must, for security reasons, enter a new password and your email address. If you forget your password and request a password reset, the notification email is sent to this email address. The email is sent by the email server configured in the **Notifications** section of the cloud **Settings**.

---

## Results

The Cloud Manager user interface opens to the home page, as shown in the following screen capture:





## What to do next

Configure the email server so you can reset your password if needed and send other notifications. See [Tutorial: Configuring the Cloud](#).

Configure your topology by registering services. See [Tutorial: Configuring the Cloud](#).

Create a Provider Organization. See [Tutorial: Creating a Provider Organization](#).

## Related tasks

- [Changing your Cloud Manager password and profile information](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

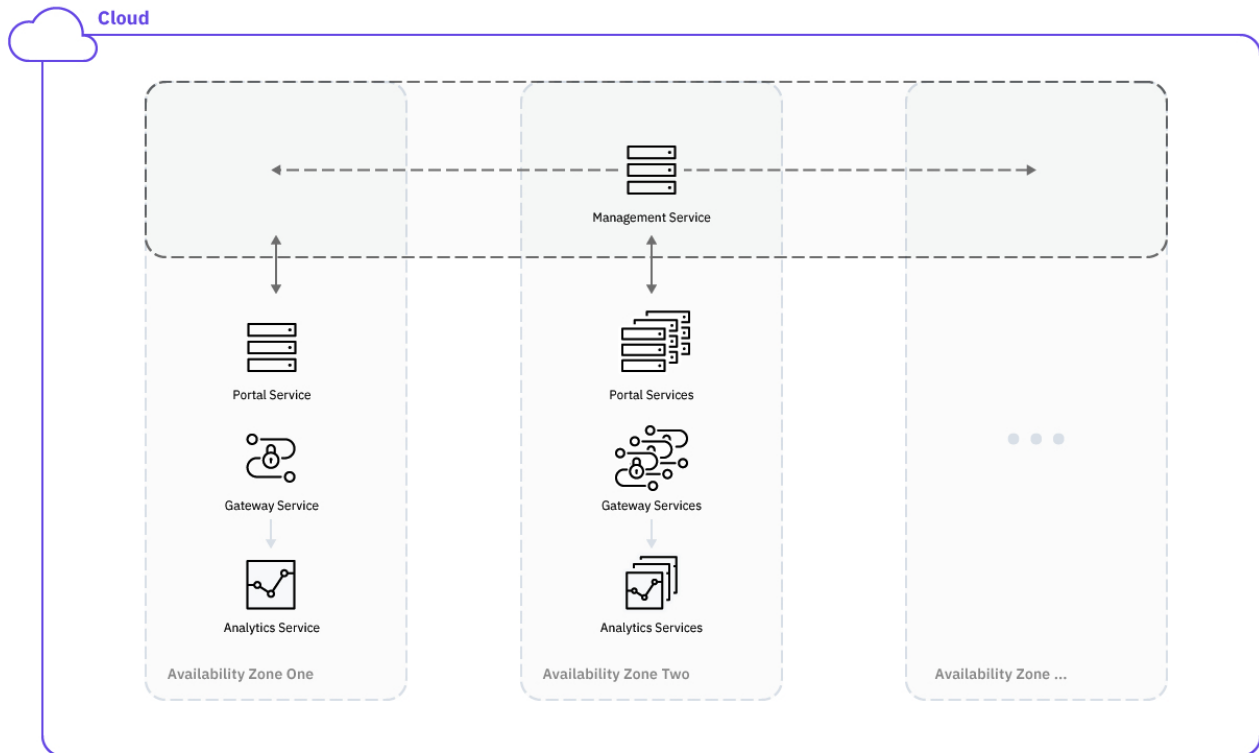
## Defining your topology

To define your API Connect on-premises cloud, you define availability zones and register services within those zones to securely create, promote, and track APIs. The installation procedure provides initial configuration of an Availability Zone containing the Management service.

## About this task

The Cloud Manager topology consists of Availability Zones that contain the API Connect services (management, gateway, analytics and portal services). Availability Zones can contain one or more gateway services, analytics service, and portal service, but there is one management service that spans all Availability Zones. When configuring your Availability Zones and Services, the recommended practice is that the gateway, analytics, and portal services do not communicate across Availability Zones. Only the management service can span across Availability Zones.

The following diagram illustrates the topology for services and Availability Zones in an API Connect network:



One of the following roles is required to register and manage services:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

Following are the tasks involved in defining the API Connect topology:

- Create one or more Availability Zones.
- Register one or more Gateway, Analytics, and Portal services in each Availability Zone
- Associate an Analytics service with each Gateway service
- Set the visibility for the services

There are two types of deployment scenarios:

1. Appliance-based deployment using .ova files on Virtual Machines
2. Container-based deployment using Docker and Kubernetes clusters

The configuration of the Management service will differ depending upon the deployment scenario. For both, a Management service is added to the Default Availability Zone by the installation utility.

Once the cloud is defined and an email server has been configured, you invite other users to access API Manager to create APIs. You expose these APIs to the development community through the Developer Portal user interface.

To define the API Connect cloud, complete the following tasks:

- **Configuring the Management service**  
A Management service is configured in the Default Availability Zone by the installation script. For an appliance-based deployment on VMs, additional Management services may be configured using the .ova file. For a container-based deployment, only one Management service is configured for the cloud.
- **Creating an Availability Zone**  
An Availability Zone is a logical or physical set of data centers containing one or more API Connect services. Availability Zones provide redundancy and failover in the event of network issues. The Default Availability Zone is created during installation; it includes a Management service.
- **Registering a gateway service**  
A gateway service is required to handle incoming traffic for APIs.
- **Registering an analytics service**  
Configure at least one analytics service in your API Connect on-premises cloud. The analytics service is always associated with a gateway service, from which it collects API event data.
- **Registering a portal service**  
Define one or more portal services in your API Connect on-premises cloud.
- **Associating an analytics service with a gateway service**  
To collect data about your API usage and other statistics, you must associate an analytics service with each gateway service. You can associate multiple gateway services with one analytics service, but each gateway can be associated with only one analytics service.
- **Setting visibility for a service**  
The visibility setting determines which provider organizations can access a service.

## Related concepts

---

- [Configuring and managing your server environment](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring the Management service

A Management service is configured in the Default Availability Zone by the installation script. For an appliance-based deployment on VMs, additional Management services may be configured using the .ova file. For a container-based deployment, only one Management service is configured for the cloud.

## Before you begin

---

You must complete the following tasks:

- Run the Install Assist installation program to configure the Default Availability Zone that contains the Management service. See [Installing API Connect into a Kubernetes environment](#) and [Deploying the Management OVA file](#).

## About this task

---

The Management service is added to your cloud by the installation script. It will be added to the Default Availability Zone. You cannot add or configure Management services using the Cloud Manager user interface.

The Management service is handled differently depending on the deployment option:

- Appliance-based deployment: API Connect Services are deployed on VMs using an .ova configuration file. Multiple Management services may be deployed, but each Availability Zone can have only one Management service.
- Container-based deployment: Services are deployed in Docker containers and managed in Kubernetes clusters. One Management service is deployed for the entire cloud using the installation script.

## Results

---

One or more Management services are configured.

## What to do next

---

Complete the following task:

- [Registering a gateway service](#)

## Related tasks

---

- [Creating an Availability Zone](#)
- [Registering a gateway service](#)
- [Registering an analytics service](#)
- [Registering a portal service](#)
- [Associating an analytics service with a gateway service](#)
- [Setting visibility for a service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating an Availability Zone

An Availability Zone is a logical or physical set of data centers containing one or more API Connect services. Availability Zones provide redundancy and failover in the event of network issues. The Default Availability Zone is created during installation; it includes a Management service.

## Before you begin

---

You must complete the following task:

- Run Install Assist to configure the Default Availability Zone that contains the Management service. See [Installing API Connect into a Kubernetes environment](#) or [Deploying the Management OVA file](#).

## About this task

Availability zones organize API Connect operations based upon your business needs. Availability zones are sets of logical or physical data centers containing one or more API Connect services. Multiple availability zones in your cloud provide redundancy and fail over in the event of network issues. Availability zones can be organized by region, for global separation, or for physical or logical separation of data centers.

The Default Availability Zone is created during the installation process. It contains the Management service that was configured by Install Assist. You register one or more gateway, analytics, and portal services in the availability zones to configure your API Connect cloud topology.

Note: For appliance-based deployments, additional management services may be added (one per availability zone) using the installation script. However, for container-based deployments using Docker and Kubernetes, only one management service is allowed. No additional management services can be added if you are using container-based deployment.

When you create an availability zone, you should provide a descriptive title; a name is generated automatically for internal identification. For example:

Table 1. Example availability zones


| Title                          | Name                           |
|--------------------------------|--------------------------------|
| US Availability Zone           | us-availability-zone           |
| Hampshire Availability Zone    | hampshire-availability-zone    |
| Newport Availability Zone      | newport-availability-zone      |
| London South Availability Zone | london-south-availability-zone |

One of the following roles is required to add and manage Availability Zones:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

## Procedure

Follow these steps to add one or more additional Availability Zones to your on-premises cloud:

1. In the Cloud Manager, click  Topology.
2. You will see the current Availability Zones configured in your cloud. The Default Availability Zone is created by the installation script, and includes the Management service. To add another Availability Zone, choose Create Availability Zone.
3. Enter the following values:

| Field              | Description                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------|
| Title (required)   | Enter a descriptive title for the availability zone. This title will display on the screen. |
| Name (required)    | This field is auto-populated by the system and used as the internal field name.             |
| Summary (optional) | Enter a brief description.                                                                  |

4. Click Create to complete the operation.

## Results

The availability zone will be added on the Services page. You can now add gateway, analytics, and portal services to the availability zone.

## Related tasks

- [Registering a gateway service](#)
- [Registering an analytics service](#)
- [Registering a portal service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Registering a gateway service

A gateway service is required to handle incoming traffic for APIs.

## Before you begin

Before registering a Gateway service in Cloud Manager, the DataPower® API Connect Gateway Service has to either be installed as a subsystem in your Kubernetes cluster or enabled on the DataPower appliance. For a Kubernetes environment, see [Installing the Gateway subsystem into a Kubernetes environment](#). For appliances, see [Configuring API Connect Gateway Service](#) for more information.

Also complete the following task:

- [Configuring the Management service](#)

## About this task

A Gateway service represents a cluster of gateway servers that host published APIs and provide the API endpoints used by client applications. Gateways execute API proxy invocations to backend systems and enforce API policies including client identification, security and rate limiting.


One of the following roles is required to register and manage a gateway services:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

Note: You can also register, and manage, gateway services by using the developer toolkit CLI; for details, see [apic gateway-services](#).

## Procedure

Complete the following steps to configure a Gateway service for your cloud:

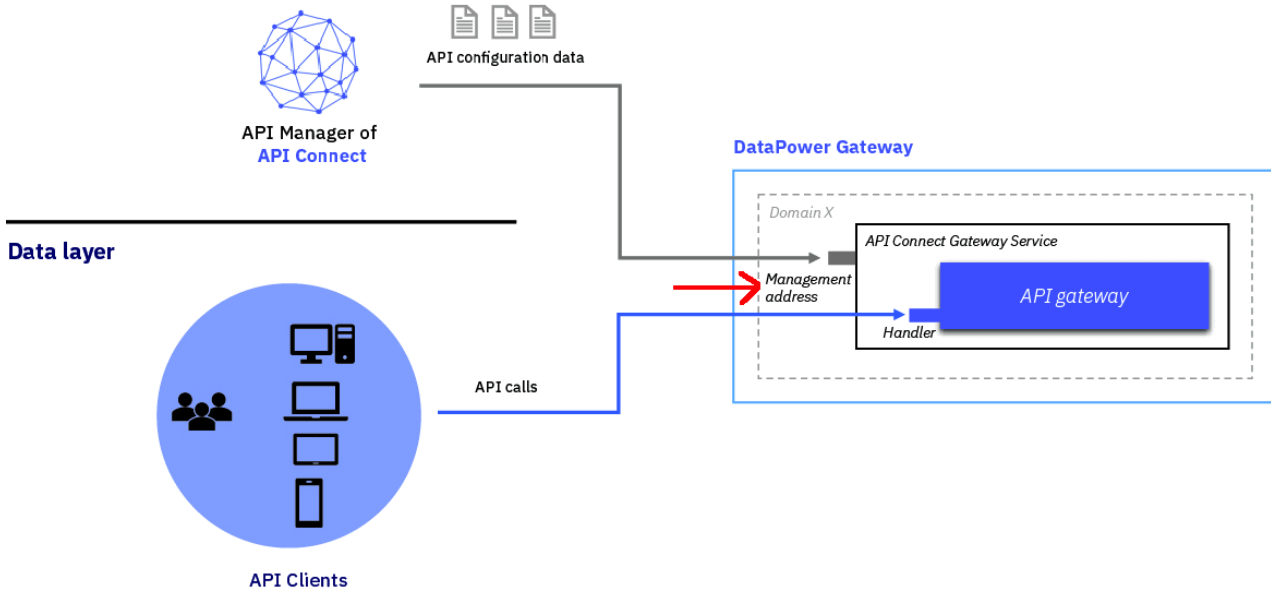
1. In the Cloud Manager, click  Topology.
2. From the Availability Zone that will contain the Gateway service, select Register Service.
3. On the Configure Service page, select DataPower Gateway as the service type. Select the Gateway type that you want to create, either DataPower Gateway (v5 compatible) or DataPower API Gateway. For a description of the gateway types, see [API Connect gateway types](#)
4. Enter the values to configure the Gateway service. You will need to obtain the endpoints from your deployment configuration. For a Kubernetes environment, the endpoints are configured by the following values in the apicup installation script. For an appliance, the endpoint is configured in DataPower.

| Field                                                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)                                                           | Enter a descriptive title for the gateway service. This title will be displayed on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Name (required)                                                            | This field is auto-populated by the system and used as the internal field name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Summary (optional)                                                         | Enter a brief description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Management Endpoint:</b><br>Endpoint (required)                         | Enter the API Connect Gateway Service endpoint. <ul style="list-style-type: none"> <li>• For a Kubernetes environment, the Management Endpoint is the endpoint entered for the command <code>set gwy apic-gw-service</code>. See <a href="#">Installing the Gateway subsystem into a Kubernetes environment</a> for more information.</li> <li>• For an appliance, the Management Endpoint is the <i>Management address</i> to the API Connect Gateway Service shown in the <a href="#">gateway service connection diagram</a>. For one gateway, this takes the form <code>http://&lt;ip-address-for-gateway&gt;:3000</code>. For multiple gateways, it would be the address:port of the load balancer</li> </ul> |
| <b>Management Endpoint:</b><br>TLS Client Profile (optional)               | Specify the TLS Client profile to use when contacting the gateway through the management endpoint.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>API Invocation Endpoint and SNI:</b> API Endpoint Base (required)       | Enter the base portion of the URL that maps to the base portion of the URL for incoming API traffic. It is a public FQDN with additional paths that are specific to your API calls. For example: <code>https://api.mycompany.com</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>API Invocation Endpoint and SNI:</b> Server Name Indication - Host Name | For supporting Server Name Indication (SNI) at the API Endpoint Base. The default hostname of '*' is required to allow all hosts. Enter other host names as needed. Wild card format is supported. The SNI capability enables you to serve multiple TLS secure host names through the same Gateway service, using the same IP address and port, without requiring them to use the same TLS profile.<br>Note: To allow requests from clients that don't support SNI, you <b>must</b> include a host name value of '*'.                                                                                                                                                                                             |
| <b>API Invocation Endpoint and SNI:</b> TLS Server Profile                 | The TLS server profile that supports the given hostname for SNI.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| OAuth Shared Secret (optional)                                             | For sites using native OAuth providers, enter the shared secret that will be used by all API calls going through the gateway.<br>Note: The specified shared secret must be 64 characters (64 bytes) in length, prefixed with <code>0x</code> , and must consist only of hexadecimal characters. For example: <code>0xa354282f227c10250511ae9c9e8c7ed9f4f1bd0d7c04cb6d5bd178f8c62296e3</code>                                                                                                                                                                                                                                                                                                                      |

The following diagram illustrates the gateway service connection:

---

## Management layer



5. When you are finished, click Save.

---

## Results

The Gateway service is added to the appropriate Availability Zone for your cloud.

---

## What to do next

If you want to change the TLS Profile from Cloud Manager, for SNI Mapping of API Invocation endpoint, it is automatically changed in the associated Gateway. You do not need to do a removal or re-registration of the Gateway service.

Add additional gateway services. Add one or more analytics services. Add one or more portal services. Associate the gateway service with an analytics service. Set the visibility for the gateway.

---

## Related tasks

- [Registering a portal service](#)
- [Registering an analytics service](#)
- [Associating an analytics service with a gateway service](#)
- [Setting visibility for a service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Registering an analytics service

Configure at least one analytics service in your API Connect on-premises cloud. The analytics service is always associated with a gateway service, from which it collects API event data.

---

## Before you begin

You must complete the following tasks:

- [Creating an Availability Zone](#)
- [Registering a gateway service](#)

---


## About this task

The Analytics service collects API event data from the Gateway service. In Cloud Manager, you configure the Analytics service and then associate it with one or more Gateways. You can have multiple Gateways associated with a single Analytics service, but each Gateway can be associated with only one Analytics service. Using the Analytics service, you can filter, sort, and aggregate the API event data and view the results on a dashboard. One of the following roles is required to register and manage analytic services:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

## Procedure

Complete the following steps to configure the Analytics services for your cloud:

1. In the Cloud Manager, click  Topology.
2. In the Availability Zone that will contain the Analytics service, select Register Services > Analytics.
3. Enter the values to configure the Analytics service:

| Field                            | Description                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)                 | Enter a descriptive title for the Analytics service. This title will display on the screen.                                                                                                                                                                                                                                                        |
| Name (required)                  | This field is auto-populated by the system and used as the internal field name.                                                                                                                                                                                                                                                                    |
| Summary (optional)               | Enter a brief description.                                                                                                                                                                                                                                                                                                                         |
| Endpoint (required)              | Enter the fully-qualified domain name. If configured during installation using the Install Assist utility, it is the <code>apicup subsys set analytics analytics-client</code> value.<br>If you are using the ingestion-only configuration for Analytics, the client endpoint is not used. Instead, provide the TLS endpoint URL for this setting. |
| TLS Client Profile (required)    | Select the TLS Client Profile that will be used to communicate with the analytics service. The Client Profile applies to the Endpoint.                                                                                                                                                                                                             |
| Advanced Analytics Configuration | Configure advanced parameters in Kibana.                                                                                                                                                                                                                                                                                                           |

4. Click Save to complete the operation.

## Results

The Analytics service is added to your cloud settings.

## What to do next

Associate the Analytics service one or more Gateway services. For more information, see [Associating an analytics service with a gateway service](#).

## Related tasks

- [Registering a gateway service](#)
- [Registering a portal service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Registering a portal service

Define one or more portal services in your API Connect on-premises cloud.

## Before you begin

You must complete the following tasks:

- [Creating an Availability Zone](#)
- [Registering a gateway service](#)
- [Configuring an email server for notifications](#)
- [Setting up notifications](#)
- [Setting up notifications](#)

Note:

When you create or register a Developer Portal service, the Portal subsystem checks that the Portal web endpoint is accessible. However sometimes, for example due to the complexity of public and private networks, the endpoint cannot be reached. The following example shows the errors that you might see in the `portal-ww` pod, admin container logs, if the endpoint cannot be reached:

```
An error occurred contacting the provided portal web endpoint: example.com
The provided Portal web endpoint example.com returned HTTP status code 504
```

In this instance, you can disable the Portal web endpoint check so that the Developer Portal service can be created successfully.  
To disable the endpoint check, complete one of the following updates depending on your platform:

On Kubernetes

Add the following section to the Portal custom resource (CR) template:

```
spec:
  template:
  - containers:
    - env:
      - name: PORTAL_SKIP_WEB_ENDPOINT_VALIDATION
        value: "true"
      name: admin
    name: www
```

On VMware

In your `apicup` project, create a file called `ptl-extra-values.yaml` (or edit the file if one already exists), and add the following section:

```
spec:
  template:
  - containers:
    - env:
      - name: PORTAL_SKIP_WEB_ENDPOINT_VALIDATION
        value: "true"
      name: admin
    name: www
```

Run the following commands:

```
apicup subsys set <ptl_subsys> extra-values-file <path-to-ptl-extra-values-yaml-file>
apicup subsys install <ptl_subsys>
```

## About this task


Each Availability Zone contains one or more Portal services. The Portal service provides a developer portal used by application developers to discover APIs and onboard consumers. An email server must be configured and set as the email server for the cloud before registering a portal service. One of the following roles is required to register and manage Portal services:

- Administrator
- Topology Administrator
- Owner
- A custom role with the `Topology:Manage` permission

Important: It's not recommended to have more than 100 sites per Developer Portal service. Note that it's not necessary to have a Portal site for every Catalog, for example Catalogs that are only for API Developers don't need a Portal site, as the APIs can be tested by using credentials from the API Manager. If more than 100 sites are required, you should configure additional Developer Portal services.

## Procedure

Complete the following steps to configure the Portal services for your cloud:

1. In the Cloud Manager, click  Topology.
2. In the Availability Zone that will contain the Portal service, select Register Services > Portal.
3. Enter the values to configure the Portal service.

| Field                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)                                          | Enter a descriptive title for the portal service. This title will display on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Name (required)                                           | This field is auto-populated by the system and used as the internal field name.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Summary (optional)                                        | Enter a brief description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Management Endpoint:</b> Endpoint (required)           | Enter the IP address, fully-qualified host name, service, or ingress name. Used for communication with API Manager. If configured during installation using the Install Assist utility, it is the <code>apicup subsys set portal portal-admin</code> value.                                                                                                                                                                                                                                                            |
| <b>Management Endpoint:</b> TLS Client Profile (optional) | Select the TLS Client Profile that will be used to communicate with the portal service. The profile applies to the Management Endpoint.                                                                                                                                                                                                                                                                                                                                                                                |
| Portal Website URL (required)                             | The URL that will be used for public access to the portal. If configured during installation using the Install Assist utility, it is the <code>apicup subsys set portal portal-www &lt;portal&gt;.&lt;hostname&gt;.&lt;domainname&gt;</code> value. Multiple <code>portal-www</code> endpoints may be configured, as described here in these topics: <a href="#">Defining multiple portal endpoints for a Kubernetes environment</a> and <a href="#">Defining multiple portal endpoints for a VMware environment</a> . |

4. When you are finished, click Save.

## Results

The Portal service is configured in your cloud and can be used to publish APIs.

## Related tasks

- [Registering a gateway service](#)
- [Registering an analytics service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Associating an analytics service with a gateway service

To collect data about your API usage and other statistics, you must associate an analytics service with each gateway service. You can associate multiple gateway services with one analytics service, but each gateway can be associated with only one analytics service.

---

### Before you begin

You must complete the following tasks:

- Run Install Assist to configure the Default Availability Zone that contains the Management service.
- Register a gateway service and an analytics service.

---

### About this task

Each gateway service is associated with an analytics service. Each gateway service can be associated with only one analytics service (a one-one relationship), however, each analytics service can be associated with one or more gateway services (a one-many relationship). The associated analytics service collects API event data from the gateway service.

In order to associate an analytics service with a gateway, you must first configure at least one analytics service and one gateway service.


One of the following roles is required to associate an analytics service with a gateway service:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

Follow these steps to associate an analytics service with a gateway service:

---

### Procedure

1. In the Cloud Manager, click  Topology.
2. In the section for the Availability Zone you want to work on, locate the Gateway service that requires an Analytics service in the SERVICE column.
3. Locate the ASSOCIATED ANALYTICS SERVICE column.
4. If an Analytics service has been configured, the Associate Analytics service link is enabled for each Gateway service. Click Associate Analytics Service in the same row as the Gateway service that requires an Analytics service.
5. In Associate Analytics Service, you will see the name of the Gateway service and a list of Analytics services that have been configured in your cloud.
6. Select the Analytics service to associate with the Gateway service by adding a checkmark, then click Associate.
7. To remove the associated Analytics service, select Unassociate analytics service from the actions menu. Note that when there is no Analytics service associated with a Gateway service, all analytics collection will be disabled and there will be no data in either the API Manager Dashboards or third-party offloads.

---

### Results

The Analytics service is associated with a Gateway service and will collect API event data for that Gateway.

---

### Related tasks

- [Registering a gateway service](#)
- [Registering an analytics service](#)

---

### Related information

- [Installing API Connect into a Kubernetes environment](#)
- [Deploying the Management OVA file](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Setting visibility for a service

The visibility setting determines which provider organizations can access a service.

---

### Before you begin

You must complete the following tasks:

- Run Install Assist to configure the Default Availability Zone that contains the Management service
- Create additional availability zones if needed, see [Creating an Availability Zone](#)
- Register at least one gateway service, see [Registering a gateway service](#)
- Register other services

## About this task

---

The visibility setting controls which provider organizations can use a service. The default visibility setting is Public.


One of the following roles is required to set the visibility for services:

- Administrator
- Topology Administrator
- Owner
- A custom role with the **Topology:Manage** permission

## Procedure

---

Follow these steps to set the visibility for the services in your on-premises cloud:

1. In the Cloud Manager, click  Topology.
2. From the list of Services, choose Set visibility from the actions menu next to the name of the service that requires the visibility setting.
3. Select the visibility setting for the service. The options are:
  - Private - the service is not visible and cannot be used by any provider organization
  - Public - the service is visible and can be used by all provider organizations
  - Custom - the service is visible only to the provider organizations designated by you
4. For Custom visibility, select the provider organizations that will be able to use the service.
5. Click Make visible to complete the operation.

## Results

---

For Private, the service cannot be used by any provider organizations. For Public, the service can be used by all provider organizations. For Custom, the service can be used by the provider organizations that you designate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing authentication and security

---

Secure Cloud Manager and API Manager as well as your Catalogs with a user registry. Secure your APIs with OAuth. Create TLS profiles to ensure that information you share among web servers will not be stolen or tampered with.

As Cloud Administrator or Topology Administrator (or with a custom role that contains the `Settings:Manage` permission), you can configure the following authentication and security mechanisms:

- User registries to authenticate users of Cloud Manager and API Manager and of your Catalogs and APIs.
- OAuth providers to provide protection for APIs.
- TLS profiles to secure transmission of data through the Gateway to external web sites and among web servers.

The following topics describe how to configure user registries, OAuth providers, and TLS profiles:

- [User registries overview](#)  
API Connect supports several types of user registries for authenticating users. The credentials for all users of Cloud Manager, API Manager, and the Developer Portal must be stored in a user registry.
- [TLS profiles overview](#)  
API Connect supports TLS Profiles for securing data transmission over HTTPS.
- [OAuth Provider overview](#)  
API Connect supports OAuth Specification 2.0, for both Native and Third party implementations.

## Related concepts

---

- [User registries overview](#)
- [OAuth Provider overview](#)
- [TLS profiles overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## User registries overview

API Connect supports several types of user registries for authenticating users. The credentials for all users of Cloud Manager, API Manager, and the Developer Portal must be stored in a user registry.

---

## Introduction to user registries

A user registry holds unique user account credentials, primarily usernames and passwords, which are accessed during authentication for logging into Cloud Manager, API Manager, the Developer Portal, and also for calling APIs (if the API is configured to use Basic Authentication in the Security Definition). The Cloud Administrator configures user registries to ensure that all users are successful at logging in.

User registries also provide security when calling APIs. For information on configuring Security Definitions for APIs, see [Creating a basic authentication security definition](#).

User registries are configured as Resources. Once the user registry has been configured as a resource in Resources > User registries, the next step is indicate the active user registries for your cloud in Cloud Settings > User registries. The user registry must be made active in your cloud in order to make it available for authentication. When a user logs into Cloud Manager or API Manager, the specified user registry is queried for credentials to confirm the user's identity.

In Cloud Manager and API Manager, a registry cannot be changed after a user is invited to be the owner of a provider organization, even if the invitation is not yet accepted.

User registries in API Connect serve the following primary functions:

- To authenticate a user at login time based upon username and password.
- To store basic profile information such as first name, last name, and email address.
- To provide secure access to Catalogs.
- To provide Basic Authentication for APIs when called by an application.

In order to log in to Cloud Manager, API Manager, the Developer Portal, or access a catalog, a user must have valid credentials (username and password) stored in a user registry that is configured in API Connect.

The Security Definition for an API can be configured to require a username and password, called the Basic Authentication method. With Basic Authentication, the username and password are included in the HTTP authorization header, and the credentials are verified through user registry.

User registries that are configured in Cloud Manager have the following characteristics:

- They are available to Cloud Manager, API Manager, the Developer Portal, and for Basic Authentication for APIs.
- They are available to provider organizations, as determined by the visibility setting.
- They can be edited and deleted only in Cloud Manager.

User registries that are configured in API Manager have the following characteristics:

- They are available to the Developer Portal (for Catalogs) and for Basic Authentication for APIs.
- They are available only to the Provider Organization that created them.

---

## User registries supported by API Connect

API Connect integrates with several types of user registries to accommodate all security solutions. You can use your corporate LDAP registry, an Authentication URL, or an LUR to provide secure access to Cloud Manager, API Manager, the Developer Portal, and APIs. API Connect includes two internal databases that serve as local user registries, or LURs. The Providers LUR supports credentials for Provider organizations and the Admin LUR supports credentials for the Administrator organization. Multiple registries may be configured.

The following user registry types can serve as a resource in API Connect:

- Local user registry (LUR) - An internal database of usernames and passwords stored on the local server. Contains the Admin user account which may not be modified or deleted.
- LDAP - A user registry definition that points to an existing corporate LDAP directory. May be set up as case-sensitive.
- Authentication URL - Accesses a URL that points to a service for validating user credentials.
- OpenID Connect (OIDC).
- [Configuring an Authentication URL user registry](#)  
An Authentication URL user registry provides a simple mechanism for authenticating users by referencing a custom identity provider.
- [Configuring an LDAP user registry in the Cloud Manager](#)  
You can use the Cloud Manager UI to configure an LDAP user registry as a shared resource to provide user authentication for the Cloud Manager, the API Manager, and the Developer Portal. APIs can also be secured with an LDAP user registry.
- [Configuring a Local User Registry](#)  
A Local User Registry (LUR) can be configured to provide user authentication for both Cloud Manager and API Manager.
- [Configuring an OIDC user registry](#)  
Configure a shared OIDC user registry for user onboarding and authentication when multi-factor authentication (MFA) is required.
- [Setting visibility for a user registry](#)  
The visibility setting determines which provider organizations can access a resource, for example, user registries.
- [Deleting a user registry](#)  
As the Cloud Manager administrator, you can delete user registries.
- [Removing a user from a user registry](#)  
As an API Connect user, you can remove yourself from an API Connect user registry. As an administrative user, you can remove other users from an API Connect user registry.

---

## Related tasks

- [Configuring an Authentication URL user registry](#)
- [Configuring a Local User Registry](#)

- [Configuring an LDAP user registry in the Cloud Manager](#)
- [Setting visibility for a user registry](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring an Authentication URL user registry

An Authentication URL user registry provides a simple mechanism for authenticating users by referencing a custom identity provider.

### About this task

This topic describes how to configure a new Authentication URL user registry as a Resource in your cloud. After the user registry is configured, you must select it for use in your cloud in Settings > User Registries. See [Selecting user registries for Cloud Manager and API Manager](#).

One of the following roles is required to configure user registries.:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

Note: When a user is presented with the form for completing their API Connect user registration, which fields are pre-populated depends on which fields are returned in the response from the Authentication URL identity provider. If any of the following fields are returned, they will be pre-populated in the registration form:

- `username`
- `email`
- `first_name`
- `last_name`

If the `username` field is not returned, the registration form displays the user name that was provided by the user. The pre-population capability requires that the response from the Authentication URL identity provider satisfies the following conditions:


- The `Content-Type` must be `application/json`.
- The response body format must be JSON.

A sample response is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

```
{
  "username": "myuser",
  "email": "myuser@example.com",
  "first_name": "My",
  "last_name": "User"
}
```

### Procedure

1. In the Cloud Manager, click .
2. Select User Registries to see the list of current user registries in your cloud.
3. Click Create in the User Registries section.
4. Select Authentication URL User Registry and enter the following parameters:

| Field                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)              | Enter a descriptive name to use on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Name (required)               | The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .                                                                                                                                                                                                                                                                                                                                                                         |
| Summary (optional)            | Enter a brief description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Display Name (required)       | The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> .<br>Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> . |
| URL (required)                | Enter the URL for the authentication service. When establishing authentication, API Connect makes a GET call to the URL. The call includes the user name and password it has collected from the user in its authorization header. Either <code>200 OK</code> or <code>401 Unauthorized</code> will be returned.                                                                                                                                                                                                                                     |
| TLS Client Profile (optional) | Select a TLS Client Profile to allow secure authentication with a specific web server.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Case sensitive | <p>To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend Authentication URL server:</p> <ul style="list-style-type: none"> <li>• <b>Only</b> select Case sensitive if your backend server supports case-sensitivity.</li> <li>• <b>Do not</b> select Case sensitive if your backend server does not support case-sensitivity.</li> </ul> <p>Note:</p> <p>The Developer Portal does not support case sensitive usernames.</p> <p>Note: After at least one user has been onboarded into the registry, you cannot change this setting.</p> |

5. Click Save.

## Results

The Authentication URL user registry is saved and is available to be selected as a user registry for Cloud Manager or API Manager by selecting it in Settings > User Registries. It can also be used for Basic Authentication in the Security Definition for an API.

## Related concepts

- [Managing authentication and security](#)
- [User registries overview](#)

## Related tasks

- [Configuring a Local User Registry](#)
- [Configuring an LDAP user registry in the Cloud Manager](#)
- [Setting visibility for a user registry](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Configuring an LDAP user registry in the Cloud Manager

You can use the Cloud Manager UI to configure an LDAP user registry as a shared resource to provide user authentication for the Cloud Manager, the API Manager, and the Developer Portal. APIs can also be secured with an LDAP user registry.

## Before you begin

To configure an LDAP user registry as a shared resource in the Cloud Manager, the LDAP directory must be created for use with your API Connect ecosystem.

LDAP registries can be used to secure APIs, to authenticate users to the Cloud Manager and the API Manager, or for securing a Catalog to authenticate Developer Portal users.

Important: If you are using an LDAP registry to secure APIs, note the following limitations:

- Authentication methods Compose DN and Compose UPN are not supported with the DataPower® API Gateway.
- The STARTTLS protocol, which upgrades an insecure protocol to a secure one by applying TLS security, is not supported with an LDAP user registry

One of the following roles is required to configure an LDAP user registry:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task

You can create an LDAP user registry that is specific to a provider organization, or one that can be shared and available to all of the provider organizations in your API Connect environment. An organization-specific LDAP user registry can be used for authenticating Developer Portal users in a specific provider organization. While a shared LDAP user registry can be used across the Cloud Manager, the API Manager, and the Developer Portal components in your environment.

This topic describes how to configure a shared LDAP user registry that is available to all of the provider organizations in your API Connect environment. If you want to create an organization-specific registry, see [Creating an LDAP user registry in API Manager](#) for more information. Note also that the visibility of a user registry is set to **shared** by default. However, you can change the visibility setting to make the registry private, or visible only to specific provider organizations. For more information, see [Setting visibility for a user registry](#).

Note:


- If you configure your LDAP user registry to be writable (by selecting the User Managed checkbox on the registry), you can use the Developer Portal UI for onboarding and authenticating new Developer Portal users, as well as those users that already exist in the LDAP database. A writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.
- You can also create and manage LDAP user registries by using the developer toolkit CLI (see [Using the CLI to configure a shared LDAP user registry](#)), and by using the API Connect REST APIs (see the [API Connect REST API documentation](#)).

You create an LDAP user registry by configuring a set of properties in the Cloud Manager UI. If you want to enable writable LDAP, you must complete the Attribute Mapping section by selecting the User Managed checkbox, and providing the mapping of your source LDAP attribute names to the target API Connect values. You can also change a registry to be read-only again by clearing the User Managed checkbox. After configuring the user registry, you must set it as active in Settings > User Registries. To make the registry available to the Developer Portal, you must define the registry for consumer onboarding in the associated Catalog. To secure APIs with an LDAP registry, you must configure security definitions.

For general information about authenticating with LDAP, see [LDAP authentication](#).

## Procedure

Follow these steps to configure a new LDAP user registry as a shared resource in the Cloud Manager UI.

1. In the Cloud Manager, click  Resources.
2. Click Create in the User Registries section.
3. Select LDAP User Registry for the user registry type, and enter the following information:

| Field                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                                  | Enter a descriptive name to display on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Name                                   | The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Display Name (required)                | The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> .<br><br>Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> .                                                               |
| Summary (optional)                     | Enter a brief description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Address                                | Enter the IP address or host name of the LDAP server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Port                                   | Enter the Port number that API Connect can use to communicate with the LDAP registry. For example, 389.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Select a TLS Client Profile (optional) | Select the TLS Client Profile the LDAP server requires.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Select an LDAP protocol version        | Select the version number for the LDAP protocol that you are using.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Case-sensitivity                       | To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend LDAP server: <ul style="list-style-type: none"> <li>• <b>Only</b> select Case sensitive if your backend LDAP server supports case-sensitivity.</li> <li>• Do <b>not</b> select Case sensitive if your backend LDAP server does not support case-sensitivity.</li> </ul> Note:<br>The Developer Portal does not support case sensitive usernames.<br><br>Note: After at least one user has been onboarded into the registry, you cannot change this setting. |

4. Click Next and enter the authentication information, which will vary depending on the selected Authentication Method. The choices are:
  - Compose DN - Select this format if you can compose the user LDAP Distinguished Name (DN) from the user name. For example, `uid=<username>,ou=People,dc=company,dc=com` is a DN format that can be composed from the user name. If you are unsure whether Compose (DN) is the correct option, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, Compose DN is not supported with the DataPower API Gateway.
  - Compose UPN - Select this format if your LDAP directory supports binding with User Principal Names such as `john@acme.com`. The Microsoft Active Directory is an example of an LDAP directory that supports Compose UPN authentication. If you are unsure whether your LDAP directory supports binding with UPNs, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, Compose UPN is not supported with the DataPower API Gateway.
  - Search DN - Select this format if you cannot compose the user LDAP Distinguished Name from the user name; for example, if the base DNs of the users are different. This format might require an administrator DN and password to search for users in the LDAP directory. If your LDAP directory permits anonymous binds, you can omit the admin DN and password. If you are unsure if your LDAP directory permits anonymous binds, contact your LDAP administrator.

For all of the authentication methods:

If you are creating an LDAP registry to authenticate users of an API, you can specify an LDAP authorization group to restrict API access. To be able to call an API that is secured by the LDAP registry, a user must successfully authenticate with their LDAP user ID and password **and** they must be a member of the specified authorization group. The authorization group can be a Static Group or Dynamic Group. A static group is one in which the individual members of the group are explicitly listed. A dynamic group is one which is defined according to the set of attributes that the group members share in common.

5. For authentication method Compose DN, enter the following:

| Field                               | Description                                                                                                                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bind Method                         | Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind. |
| Admin DN                            | For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> .              |
| Admin Password                      | For Authenticated Bind, enter the user password for the Admin DN.                                                                                                                       |
| Prefix                              | Specify the prefix to the DN. For example (uid=).                                                                                                                                       |
| Suffix                              | Specify the suffix to the DN. For example ).                                                                                                                                            |
| Base DN (optional)                  | Enter a base DN in the Base DN field, or click Get Base DN to populate the field with a retrieved base DN.                                                                              |
| Use group authentication (optional) | Static or Dynamic. For Static Group, enter the Group Based DN, Prefix, and Suffix. For Dynamic Group, enter the Filter condition for the group.                                         |

6. For authentication method Compose UPN, enter the following:

| Field       | Description                                                                                                                                                                             |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bind Method | Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind. |

| Field                               | Description                                                                                                                                                                |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Admin DN                            | For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> . |
| Admin Password                      | For Authenticated Bind, enter the user password for the Admin DN.                                                                                                          |
| Suffix                              | Enter the domain part of the user principal name. For example, <code>@acme.com</code> .                                                                                    |
| Use group authentication (optional) | Enter the Filter condition for the group.                                                                                                                                  |

7. For authentication method Search DN, enter the following:

| Field                               | Description                                                                                                                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bind Method                         | Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind. |
| Admin DN                            | For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> .              |
| Admin Password                      | For Authenticated Bind, enter the user password for the Admin DN.                                                                                                                       |
| Prefix                              | Specify the prefix to the DN. For example <code>(uid=</code> .                                                                                                                          |
| Suffix                              | Specify the suffix to the DN. For example <code>)</code> .                                                                                                                              |
| Base DN (optional)                  | Enter a base DN in the Base DN field, or click Get Base DN to populate the field with a retrieved base DN.                                                                              |
| Use group authentication (optional) | Static or Dynamic. For Static Group, enter the Group Based DN, Prefix, and Suffix. For Dynamic Group, enter the Filter condition for the group.                                         |

8. Optional: Click Test configuration to test the settings for your LDAP user registry. Enter valid credentials to ensure that you can access the LDAP database.

9. Optional: If you want to make your LDAP user registry writable, select the User Managed checkbox in the Attribute Mapping section, and provide the mapping of your source LDAP attribute names to the target API Connect values. Click Add to add each name/value pair, specified as follows:

- LDAP ATTRIBUTE NAME - is the name of the source LDAP attribute.
- API CONNECT VALUE - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.

The default user profile properties that API Connect requires during user registration are `username`, `first_name`, `last_name`, `email`, and `password`, as shown in the following example:

| LDAP ATTRIBUTE NAME       | API CONNECT VALUE                                      |
|---------------------------|--------------------------------------------------------|
| <code>dn</code>           | <code>uid=[username],ou=users,dc=company,dc=com</code> |
| <code>cn</code>           | <code>[first_name] [last_name]</code>                  |
| <code>sn</code>           | <code>[last_name]</code>                               |
| <code>mail</code>         | <code>[email]</code>                                   |
| <code>userPassword</code> | <code>[password]</code>                                |

You must ensure that you enter the correct attribute mapping values for your LDAP configuration, to enable API Connect to access the LDAP database. Note that a writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.

10. Click Create.

Your new LDAP registry is shown in the list of User Registries on the Resources page.

## What to do next

To make the LDAP registry available for user authentication in the Cloud Manager and the API Manager, you must set it as active in the Settings > User Registries section. See [Selecting user registries for Cloud Manager and API Manager](#) for more information.

If you want to make the LDAP registry available for authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. In the API Manager UI, click Manage followed by the relevant Catalog, and then click Settings > Onboarding. In the Catalog User Registries section, click Edit, select the user registry, and click Save. For more information, see [Creating and configuring Catalogs](#).

If you want to use the LDAP user registry to secure APIs, see the following information:

- To use for basic authentication in the security definition for an API, see [Creating a basic authentication security definition](#).
- To use for authentication in the User Security configuration for a native OAuth provider, see [Configuring user security for a native OAuth provider](#).
- [Using the CLI to configure a shared LDAP user registry](#)  
You can use the developer toolkit CLI to configure an LDAP user registry to provide user authentication for the Cloud Manager, the API Manager, and the Developer Portal. APIs can also be secured with an LDAP user registry.

## Related concepts

- [Managing authentication and security](#)
- [User registries overview](#)

## Related tasks

- [Configuring a Local User Registry](#)
- [Configuring an Authentication URL user registry](#)
- [Setting visibility for a user registry](#)

## Related information

- [LDAP authentication](#)
- [Securing your API Connect Cloud with LDAP \(series of developer articles\)](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using the CLI to configure a shared LDAP user registry

You can use the developer toolkit CLI to configure an LDAP user registry to provide user authentication for the Cloud Manager, the API Manager, and the Developer Portal. APIs can also be secured with an LDAP user registry.

---

### Before you begin

To configure an LDAP user registry as a resource in the Cloud Manager, the LDAP directory must be created and available to use with your API Connect ecosystem.

LDAP registries can be used to secure APIs, or for authenticating users to the Cloud Manager, the API Manager, and the Developer Portal.

Important: If you are using an LDAP registry to secure APIs, note the following limitations:

- Authentication methods `compose_dn` and `compose_upn` is not supported with the DataPower® API Gateway.
- The STARTTLS protocol, which upgrades an insecure protocol to a secure one by applying TLS security, is not supported with an LDAP user registry

One of the following roles is required to configure an LDAP user registry:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

---

### About this task

You can create an LDAP user registry that is specific to a provider organization, or one that can be shared and available to all of the provider organizations in your API Connect environment. An organization-specific LDAP user registry can be used for onboarding and authenticating Developer Portal users, while a shared LDAP user registry can be used for authenticating Cloud Manager, API Manager, and Developer Portal users.

This topic describes how to configure a shared LDAP user registry. If you want to create an organization-specific registry, see [Using the CLI to create an organization-specific LDAP user registry](#) for more information. Note also that the visibility of a user registry is set to `shared` by default. However, you can change the visibility setting to make the registry private, or visible only to specific provider organizations. For more information, see [Setting visibility for a user registry](#).

Note:

- If the LDAP user registry is configured as writable (by enabling the `user-managed` property on the registry), you can use the Developer Portal UI for onboarding and authenticating new Developer Portal users, as well as those users that already exist in the LDAP database. A writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.
- You can also create LDAP user registries by using the Cloud Manager UI (see [Configuring an LDAP user registry in the Cloud Manager](#)), and by using the API Connect REST APIs (see the [API Connect REST API documentation](#)).

You configure an LDAP user registry by first defining the registry details in a configuration file. You then use a developer toolkit CLI command to create the registry, passing the configuration file as a parameter. Finally, to make the registry available for user authentication, you need to configure the registry in the management server, or in the appropriate Catalog, or both, depending on your requirements. To secure APIs with an LDAP registry, you must configure security definitions. You can use the following instructions to create a writable or a read-only LDAP user registry.

For general information about authenticating with LDAP, see [LDAP authentication](#).

---

### Logging in to the management server CLI

Before you can define the LDAP user registry configuration, you must log in to your management server from the developer toolkit CLI as a member of the cloud administration organization. Use the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm admin/identity_provider
```

For full details of the login command, see [Logging in to a management server](#).

For more information about how to use the CLI, see [Installing the toolkit](#), and [Overview of the command-line tool](#).

---

### Defining your LDAP configuration

You define the configuration of your LDAP user registry in an `ldap_config_file.yaml` file, as shown in the following example. Note that the actual contents of your YAML file will vary depending on the authentication method of your LDAP server, and this is explained in the following tables.

```
name: registry_name
title: "display_title"
integration url: LDAP_integration_url
user_managed: true_or_false
user_registry_managed: false
case_sensitive: true_or_false
identity_providers:
- name: provider_name
  title: provider_title
endpoint:
  endpoint: "ldap_server_url_and_port"
configuration:
  authentication method: authentication_method
  authenticated_bind: "true_or_false"
  admin_dn: "admin_dn"
```



```

admin_password: admin_password
search_dn_base: "search_dn_base"
search_dn_scope: search_dn_scope
search_dn_filter_prefix: prefix
search_dn_filter_suffix: suffix
attribute_mapping:
  dn: "distinguished_name"
  cn: "common_name"
  sn: "last_name"
  mail: "email_address"
  userPassword: "password"

```

The registry properties that are common to each authentication method are described in the following table:

| Property                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>                  | The name of the registry. This name is used in CLI commands.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>title</b>                 | A descriptive name to display in a graphical user interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>integration_url</b>       | The LDAP integration URL in your API Connect configuration. You can determine the LDAP integration URL by using the following CLI command:<br><code>apic integrations:list --server mgmt_endpoint_url --subcollection user-registry</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>user_managed</b>          | Determines whether your user registry is writable or not. Must be set to <b>true</b> for writable LDAP. You can change this setting to <b>false</b> if you don't want the registry to be writable; see the Switching your LDAP registry between writable and read-only section at the end of this topic for details. Note that a writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.                                                                                                                                                                                                                                                                                                                |
| <b>user_registry_managed</b> | Must be set to <b>false</b> for LDAP. Determines whether API Connect manages your user registry. Only LUR registries are managed by API Connect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>case_sensitive</b>        | Determines whether your user registry is case-sensitive. Valid values are: <ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul> <p>To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend LDAP server:</p> <ul style="list-style-type: none"> <li>• <b>Only</b> set <b>case_sensitive</b> to <b>true</b> if your backend LDAP server supports case-sensitivity.</li> <li>• Set <b>case_sensitive</b> to <b>false</b> if your backend LDAP server does not support case-sensitivity.</li> </ul> <p>Note: After at least one user has been onboarded into the registry, you cannot change this setting.</p> |
| <b>identity_providers</b>    | An array containing the details of your LDAP server, where: <ul style="list-style-type: none"> <li>• <b>name</b> - is the name of the LDAP server and is the name that is used in CLI commands</li> <li>• <b>title</b> - is the display name of the LDAP server</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>endpoint</b>              | The endpoint of your LDAP server, made up of the url and port, for example:<br><code>"ldap://server.com:389"</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>tls_profile</b>           | Optionally set the TLS Client Profile that the LDAP server requires.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>protocol_version</b>      | Optionally set the version number for the LDAP protocol that you are using. Valid values are: <ul style="list-style-type: none"> <li>• 2</li> <li>• 3</li> </ul> <p>Defaults to 3 if not explicitly set.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

The properties in the configuration section will vary depending on the selected authentication method. The three authentication methods are:

- **compose\_dn** - Set this format if you can compose the user LDAP Distinguished Name (DN) from the user name. For example, `uid=<username>,ou=People,dc=company,dc=com` is a DN format that can be composed from the user name. If you are unsure whether Compose (DN) is the correct option, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, **compose\_dn** is not supported with the DataPower API Gateway.
- **compose\_upn** - Set this format if your LDAP directory supports binding with User Principal Names such as `john@acme.com`. The Microsoft Active Directory is an example of an LDAP directory that supports Compose UPN authentication. If you are unsure whether your LDAP directory supports binding with UPNs, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, **compose\_upn** is not supported with the DataPower API Gateway.
- **search\_dn** - Select this format if you cannot compose the user LDAP Distinguished Name from the user name; for example, if the base DN of the users are different. This format might require an administrator DN and password to search for users in the LDAP directory. If your LDAP directory permits anonymous binds, you can omit the admin DN and password. If you are unsure if your LDAP directory permits anonymous binds, contact your LDAP administrator.

For authentication method **compose\_dn**, set the following configuration properties:

| Properties                   | Description                                                                                                                                                                                                                                                                                                                    |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>authentication_method</b> | <b>compose_dn</b>                                                                                                                                                                                                                                                                                                              |
| <b>authenticated_bind</b>    | The bind method. Valid values are: <ul style="list-style-type: none"> <li>• <b>"true"</b> - authenticated bind</li> <li>• <b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p> |
| <b>admin_dn</b>              | If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example:<br><code>"cn=admin,dc=company,dc=com"</code>                                                                                                             |
| <b>admin_password</b>        | If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the user password for the <b>admin_dn</b> .                                                                                                                                                                                                                       |
| <b>search_dn_base</b>        | Optionally set a base DN, for example:<br><code>"dc=company,dc=com"</code>                                                                                                                                                                                                                                                     |
| <b>bind_prefix</b>           | Set the prefix to the DN, for example:<br><code>(uid=</code>                                                                                                                                                                                                                                                                   |

| Properties               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bind_suffix</b>       | Set the suffix to the DN, for example:<br>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>attribute_mapping</b> | <p>If <b>user_managed</b> is set to <b>true</b>, provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows:</p> <pre>ldap_registry_attribute_name: "apic_ldap_attribute_value"</pre> <p>Where:</p> <ul style="list-style-type: none"> <li><code>ldap_registry_attribute_name</code> - is the name of the source LDAP attribute</li> <li><code>apic_ldap_attribute_value</code> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.</li> </ul> <p>The user profile properties that API Connect requires during user registration are <b>username</b>, <b>first_name</b>, <b>last_name</b>, <b>email</b>, and <b>password</b>. The following extract shows an example of an attribute mapping:</p> <pre>attribute_mapping: dn: "uid=[username],ou=users,dc=company,dc=com" cn: "[first_name] [last_name]" sn: "[last_name]" mail: "[email]" userPassword: "[password]"</pre> |

For authentication method **compose\_upn**, set the following configuration properties:

| Properties                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>authentication_method</b> | <b>compose_upn</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>authenticated_bind</b>    | <p>The bind method. Valid values are:</p> <ul style="list-style-type: none"> <li><b>"true"</b> - authenticated bind</li> <li><b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>admin_dn</b>              | <p>If <b>authenticated_bind</b> is set to <b>"true"</b>, enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example:</p> <pre>"cn=admin,dc=company,dc=com"</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>admin_password</b>        | If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the user password for the <b>admin_dn</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>bind_suffix</b>           | <p>Enter the domain part of the user principal name. For example:</p> <pre>@acme.com</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>attribute_mapping</b>     | <p>If <b>user_managed</b> is set to <b>true</b>, provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows:</p> <pre>ldap_registry_attribute_name: "apic_ldap_attribute_value"</pre> <p>Where:</p> <ul style="list-style-type: none"> <li><code>ldap_registry_attribute_name</code> - is the name of the source LDAP attribute</li> <li><code>apic_ldap_attribute_value</code> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.</li> </ul> <p>The user profile properties that API Connect requires during user registration are <b>username</b>, <b>first_name</b>, <b>last_name</b>, <b>email</b>, and <b>password</b>. The following extract shows an example of an attribute mapping:</p> <pre>attribute_mapping: dn: "uid=[username],ou=users,dc=company,dc=com" cn: "[first_name] [last_name]" sn: "[last_name]" mail: "[email]" userPassword: "[password]"</pre> |

For authentication method **search\_dn**, set the following configuration properties:

| Property                     | Description                                                                                                                                                                                                                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>authentication_method</b> | <b>search_dn</b>                                                                                                                                                                                                                                                                                                                  |
| <b>authenticated_bind</b>    | <p>The bind method. Valid values are:</p> <ul style="list-style-type: none"> <li><b>"true"</b> - authenticated bind</li> <li><b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p> |
| <b>admin_dn</b>              | <p>If <b>authenticated_bind</b> is set to <b>"true"</b>, enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example:</p> <pre>"cn=admin,dc=company,dc=com"</pre>                                                                                                               |
| <b>admin_password</b>        | If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the user password for the <b>admin_dn</b> .                                                                                                                                                                                                                          |
| <b>search_dn_base</b>        | <p>Optionally set a base DN, for example:</p> <pre>"dc=company,dc=com"</pre>                                                                                                                                                                                                                                                      |

| Property                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>search_dn_scope</code>         | Optionally set the search DN scope. The scope determines which part of the directory information tree is examined. Possible values are: <ul style="list-style-type: none"> <li><code>base</code></li> <li><code>one</code></li> <li><code>sub</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>search_dn_filter_prefix</code> | Set the prefix to the DN, for example:<br>(uid=                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>search_dn_filter_suffix</code> | Set the suffix to the DN, for example:<br>)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>attribute_mapping</code>       | <p>If <code>user_managed</code> is set to <code>true</code>, provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows:</p> <pre>ldap_registry_attribute_name: "apic_ldap_attribute_value"</pre> <p>Where:</p> <ul style="list-style-type: none"> <li><code>ldap_registry_attribute_name</code> - is the name of the source LDAP attribute</li> <li><code>apic_ldap_attribute_value</code> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.</li> </ul> <p>The user profile properties that API Connect requires during user registration are <code>username</code>, <code>first_name</code>, <code>last_name</code>, <code>email</code>, and <code>password</code>. The following extract shows an example of an attribute mapping:</p> <pre>attribute_mapping: dn: "uid=[username],ou=users,dc=company,dc=com" cn: "[first_name] [last_name]" sn: "[last_name]" mail: "[email]" userPassword: "[password]"</pre> |

Save your `ldap_config_file.yaml` so it can be accessed by the `user-registries:create` command in the following section. See the [Example](#) section for an example configuration file.

## Creating your LDAP user registry

To create your shared LDAP user registry, run the following CLI command:

```
apic user-registries:create --server mgmt_endpoint_url --org admin ldap_config_file.yaml
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL.
- `--org admin` means that the registry will be created in the admin organization, which is required for a user registry to be shared.
- `ldap_config_file` is the name of the YAML file that defines the configuration of your LDAP user registry.

On completion of the registry creation, the command displays the following summary details:

```
registry_name registry_url
```

The `registry_name` is derived from the `name` property in the configuration YAML file. The `registry_url` is the URL with which the registry resource can be accessed. Your shared LDAP user registry is now created; see the following sections for instructions on how to make the registry available to users.

## Configuring your LDAP registry for Cloud Manager or API Manager login

If you want to make your LDAP registry available for authenticating Cloud Manager and API Manager users, you must configure it on the management server.

- Determine the URL of your LDAP user registry by using the following command (or you can copy and paste from the summary of the registry creation):

```
apic user-registries:list --server mgmt_endpoint_url --org admin
```

- Determine what the current user registries are, because you will need to confirm these as well as add your new LDAP registry:

```
apic user-registry-settings:get --server mgmt_endpoint_url --output -
```

This command outputs a list of all the current user registries in your environment, similar to the following example:

```
type: user_registry_setting
api_version: 2.0.0
name: user-registry-setting
admin_user_registry_default_url: >-
  https://server.com/api/user-registries/xxxxx/xxxxx-1234
admin_user_registry_urls:
- >-
  https://server.com/api/user-registries/xxxxx/xxxxx-1234
provider_user_registry_default_url: >-
  https://https://server.com/api/user-registries/xxxxx/xxxxx-5678
provider_user_registry_urls:
- >-
  https://https://server.com/api/user-registries/xxxxx/xxxxx-5678
created_at: '2019-09-30T12:22:19.467Z'
updated_at: '2019-10-17T10:05:37.867Z'
url: 'https://server.com/api/cloud/settings/user-registries'
```

- Enter the following command to update your user registries (the terminating hyphen character means that the command takes input from the command line):

```
apic user-registry-settings:update --server mgmt_endpoint_url -
```

The following message is output:

```
Reading USER_REGISTRY_SETTING_FILE arg from stdin
```

4. If you want to make your LDAP registry available for authenticating Cloud Manager users, enter the following data, followed by a new line:

```
admin_user_registry_urls:
- >-
  current_admin_user_registry_urls
- new_ldap_user_registry_url
```

where:

- `current_admin_user_registry_urls` are the current admin user registry URLs, as listed in Step 2 in the `admin_user_registry_urls` section. Note that you must include all of the user registry URLs that you want to remain, listing each URL on a new line.
- `new_ldap_user_registry_url` is the URL of your new LDAP user registry, as determined in Step 1.

5. If you want to make your LDAP registry available for authenticating API Manager users, enter the following data, followed by a new line:

```
provider_user_registry_urls:
- >-
  current_provider_user_registry_urls
- new_ldap_user_registry_url
```

where:

- `current_provider_user_registry_urls` are the current provider organization user registry URLs, as listed in Step 2 in the `provider_user_registry_urls` section. Note that you must include all of the user registry URLs that you want to remain, listing each URL on a new line.
- `new_ldap_user_registry_url` is the URL of your new LDAP user registry, as determined in Step 1.

6. Press **CTRL D** to terminate the input. A confirmation message is output, for example:

```
user-registry-setting https://server.com/api/cloud/settings/user-registries
```

where `server.com` is your management server endpoint.

## Configuring your LDAP registry in a Catalog

If you want to make your LDAP registry available for authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. Complete the following steps:

1. Determine the URL of your LDAP user registry by using the following command (or you can copy and paste from the summary of the registry creation):

```
apic user-registries:list --server mgmt_endpoint_url --org admin
```

2. Log in to the management server as a member of a provider organization by entering the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

3. Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic configured-catalog-user-registries:create --server mgmt_endpoint_url --org organization_name --catalog catalog_name -
```

where `catalog_name` is the value of the `name` property of the required Catalog. The command returns:

```
Reading CONFIGURED_CATALOG_USER_REGISTRY_FILE arg from stdin
```

4. Enter the following data, followed by a new line:

```
user_registry_url: ldap_registry_url
```

where `ldap_registry_url` is the URL of your LDAP registry, obtained in step 1.

5. Press **CTRL D** to terminate the input.

## Switching your LDAP registry between writable and read-only

After an LDAP user registry has been created, it can be switched between writable and read-only by updating the `user_managed` property in the registry configuration. Complete the following steps.

1. Determine the name or ID of the LDAP user registry that you want to update, by running the following command (or you can use the summary from the registry creation):

```
apic user-registries:list --server mgmt_endpoint_url --org admin
```

The command returns a list of all the user registries on that server, shown by name followed by their registry URL. The registry ID is located at the end of the URL, for example `https://company.com/api/user-registries/x-x-x-x-x-x/registry_id`.

2. Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic user-registries:update --server mgmt_endpoint_url --org admin registry_name_or_id -
```

where `registry_name_or_id` is the name or ID of the LDAP user registry that you want to update (as determined in the previous step). The command returns:

```
Reading USER_REGISTRY_FILE arg from stdin
```

3. Enter the following data, followed by a new line:

```
user_managed: true_or_false
```

where `true` makes the registry writable, and `false` makes the registry read-only.

4. Press **CTRL D** to terminate the input.

Note that if you are changing your registry from read-only to writable, you must also set the `attribute_mapping` configuration, as described in the previous registry property tables.

## Using an LDAP user registry to secure APIs

---

If you want to use the LDAP user registry to secure APIs, see the following information:

- To use for basic authentication in the security definition for an API, see [Creating a basic authentication security definition](#).
- To use for authentication in the User Security configuration for a native OAuth provider, see [Configuring user security for a native OAuth provider](#).

For details of all the `apic user-registries` and `apic configured-catalog-user-registries` commands, see [apic user-registries](#) and [apic configured-catalog-user-registries](#).

## Example

---

The following example shows a configuration file that uses the Search DN authentication method for setting up writable LDAP:

```
name: sdn-ldap
title: "SDN LDAP User Registry"
integration_url: https://mycompany.com/api/cloud/integrations/user-registry/xxx-xxx-xxx
user_managed: true
user_registry_managed: false
case_sensitive: false
identity_providers:
  - name: ldap
    title: "SDN LDAP Identity Provider"
endpoint:
  endpoint: "ldap://mycompany.com:389"
configuration:
  authentication_method: search_dn
  authenticated_bind: "true"
  admin_dn: "cn=admin,dc=company,dc=com"
  admin_password: xxxx
  search_dn_base: "dc=company,dc=com"
  search_dn_scope: sub
  search_dn_filter_prefix: (uid=
  search_dn_filter_suffix: )
  attribute_mapping:
    dn: "uid=[username],ou=users,dc=company,dc=com"
    cn: "[first_name] [last_name]"
    sn: "[last_name]"
    mail: "[email]"
    userPassword: "[password]"
```

## Related information

---

- [Command-line tool reference for the developer toolkit](#)
- [Securing your API Connect Cloud with LDAP \(series of developer articles\)](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring a Local User Registry

---

A Local User Registry (LUR) can be configured to provide user authentication for both Cloud Manager and API Manager.

### About this task

---

Local User Registries (LURs) are the default user registries included in API Connect. LURs are local databases included with API Connect. Two default LURs are installed and configured during installation of API Connect. They cannot be deleted. The default Admin user account is stored in the Provider LUR.

You can create and configure a new LUR. After you create it, you must set it as active in Settings > User Registries. See [Selecting user registries for Cloud Manager and API Manager](#).

One of the following roles is required to configure user registries:

- Administrator
- Owner
- Topology Administrator
- Custom role with the Settings: Manage permissions

## Procedure

---

Follow these steps to configure a new LUR:

1. In the Cloud Manager, click  .

- Click Create in the User Registries section.
- Select Local User Registry as the type for the user registry and enter the following information:

| Field                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)        | Enter a descriptive name for use on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Name (required)         | The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .                                                                                                                                                                                                                                                                                                                                                                         |
| Display Name (required) | The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> .<br>Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> . |
| Summary (optional)      | Enter a brief description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Case sensitive          | Select this setting if user names are case-sensitive.<br>Note:<br>The Developer Portal does not support case sensitive usernames.<br>Note: After at least one user has been onboarded into the registry, you cannot change this setting.                                                                                                                                                                                                                                                                                                            |

- Click Save.

## Results

The user registry is saved and is available to be configured for user authentication in Cloud Manager or API Manager by selecting it in Settings > User Registries. See [Selecting user registries for Cloud Manager and API Manager](#).

## Related concepts

- [Managing authentication and security](#)
- [User registries overview](#)

## Related tasks

- [Configuring an Authentication URL user registry](#)
- [Configuring an LDAP user registry in the Cloud Manager](#)
- [Setting visibility for a user registry](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring an OIDC user registry

Configure a shared OIDC user registry for user onboarding and authentication when multi-factor authentication (MFA) is required.

You can create an OIDC user registry that is specific to a provider organization, or that is shared and available to all the provider organizations in your API Connect environment. An organization-specific OIDC user registry is used for onboarding and authenticating Developer Portal users, while a shared OIDC user registry can be used for onboarding and authenticating Cloud Manager, API Manager, and Developer Portal users.

This topic describes how to create a shared registry. For information on how to create an organization-specific registry, see [Creating an OIDC user registry](#).


API Connect provides two methods for creating an OIDC user registry in Cloud Manager, as described in the following sections:

- [Using the UI to configure an OIDC user registry](#)
- [Using the CLI to configure an OIDC user registry](#)

Note: Refresh tokens are not supported for a user in an OIDC user registry when accessing the Cloud Manager or API Manager user interfaces.

## Using the UI to configure an OIDC user registry

Use the Cloud Manager application's user interface to configure a shared OIDC user registry when multi-factor authentication (MFA) is required.

- In the Cloud Manager navigation pane, click  Resources.
- Click User Registries.
- Click Create and select OIDC User Registry.
- On the Create OIDC User Registry page, use the fields in each of the following sections to configure the registry settings, and then click Create. Many of the registry settings are preconfigured to simplify the configuration steps.

### Provider Type

Use the settings in Table 1 to define the provider type.

Table 1. Provider Type settings

| Field | Description |
|-------|-------------|
|-------|-------------|

| Field         | Description                                                                                                                                                                                                                                                                                                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Provider Type | OIDC provider. Select one of the following supported OIDC providers: <ul style="list-style-type: none"> <li>• Facebook</li> <li>• GitHub</li> <li>• Google</li> <li>• LinkedIn</li> <li>• Slack</li> <li>• Twitter</li> <li>• Windows Live</li> <li>• Standard OIDC (default value allows you to specify another provider)</li> </ul> |
| Title         | Provide a descriptive name for display purposes.                                                                                                                                                                                                                                                                                      |
| Name          | Automatically generated. This name is used in CLI commands to reference the registry. For details of the CLI commands for managing user registries, see the <a href="#">apic user-registries</a> topic in the Command Line tool reference section of this documentation.                                                              |
| Summary       | Provide a brief description of the new registry.                                                                                                                                                                                                                                                                                      |

Provider Endpoint

Automatically generated for most supported providers. In the Authorization Endpoint field, type the URL of the provider's authorization endpoint.

Token Endpoint

Fill in the settings as described in Table 2.

Table 2. Token Endpoint settings

| Field | Description                                                                                                                       |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|
| URL   | Preconfigured for most of the supported OIDC providers. Type the URL of the provider's token endpoint.                            |
| TLS   | Select the TLS Client Profile for the token endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile. |

UserInfo Endpoint

Fill in the settings as described in Table 3.

Table 3. UserInfo Endpoint settings

| Field | Description                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------------------|
| URL   | Preconfigured for most of the supported OIDC providers. Type the URL of the provider's userinfo endpoint.                            |
| TLS   | Select the TLS Client Profile for the userinfo endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile. |

JWKS Endpoint

Fill in the settings as described in Table 4.

Table 4. JWKS Endpoint settings

| Field | Description                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------------------|
| URL   | Type the URL of the read-only endpoint that contains the public keys' information in JWKS format.                                    |
| TLS   | Select the TLS Client Profile for the userinfo endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile. |

Client Information

Fill in the settings as described in Table 5.

Table 5. Client Information settings

| Field                        | Description                                                                                                                                                                                                                                                     |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Client ID                    | Provide the client ID of the application that is registered with the selected OIDC provider.                                                                                                                                                                    |
| Client Secret                | Provide the client secret of the application that is registered with the selected OIDC provider.                                                                                                                                                                |
| Response Type                | Preconfigured for most of the supported OIDC providers. Specify the data type of the response that will be received from the OIDC provider.                                                                                                                     |
| Scopes                       | Preconfigured for most of the supported OIDC providers. Specify the access scope for the OIDC provider.                                                                                                                                                         |
| Client Authentication Method | Preconfigured for most of the supported OIDC providers. Select the authentication method to be used with the OIDC provider. Options are: <ul style="list-style-type: none"> <li>• Http basic authentication schema</li> <li>• Data encoded form body</li> </ul> |

Additional Support

Optional. Select the additional security parameters described in Table 6.

Table 6. Additional security options

| Security parameter                 | Description                                                                                                            |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| NONCE                              | Enable the NONCE extension to prevent compromised requests from being used again (replayed).                           |
| Proof Key for Code Exchange (PKCE) | Enable the PKCE extension to allow public clients to mitigate the threat of having the authorization code intercepted. |

Advanced Features

Optional. Select the advanced features described in Table 7.

Table 7. Advanced features

| Feature                          | Description                                                                                                                     |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Auto onboard                     | Allow users to execute calls to APIs without logging in first, provided they present a valid token issued by the OIDC provider. |
| Always use the userinfo endpoint | Configures the OIDC user registry to always fetch user data from the userinfo endpoint, if populated.                           |
| Return third-party access token  | Include the third-party OIDC access token in the response.                                                                      |
| Return third-party id_token      | Include the third-party OIDC id_token in the response.                                                                          |

User Mapping

Fill in the settings as described in Table 8.

Note:

The User Mapping fields are preconfigured for most of the supported OIDC providers to minimize potential errors; use care when changing the settings. For the Standard OIDC option, contact your OIDC provider to obtain the details of the fields.


Table 8. User mapping settings

| Field | Description |
|-------|-------------|
|-------|-------------|

| Field      | Description                                                                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Username   | The name of the field in the response token that contains the user's user name.<br>Note: The username field must be unique for this OIDC registry, because it identifies the user in the system and cannot be changed. |
| Email      | The name of the field in the response token that contains the user's email address.                                                                                                                                    |
| First name | The name of the field in the response token that contains the user's given name.                                                                                                                                       |
| Last name  | The name of the field in the response token that contains the user's surname.                                                                                                                                          |

## Enabling the OIDC user registry

Complete the following steps to enable the new user registry for Cloud Manager, API Manager, or both.

1. On the navigation pane, click  Settings.
2. Click User Registries, > Edit.
3. Click Edit in the section that corresponds to the application for which you are enabling the new user registry.
4. Select the new OIDC user registry.
5. When you have finished enabling the registry, click Save.

## Using the CLI to configure an OIDC user registry

Use the developer toolkit CLI to configure a shared OIDC user registry when multi-factor authentication (MFA) is required.

You configure an OIDC user registry by first defining the registry details in a configuration file. You then use a CLI command to create the registry, passing the configuration file as parameter. To make the registry available to the Developer Portal, you must enable the registry in the associated Catalog.

Note:

- An OIDC registry, in common with a Local User Registry, cannot be used to secure APIs on the gateway.
- Because the interaction with the third party OIDC provider is handled by the Management server, the Management server is the application from the point of view of the third party OIDC provider. Your OIDC redirection endpoint, which is used by authorization server to send the token to the Management server, must be accessible to the OIDC provider through your firewall. When you register your application with the third party OIDC provider, you are required to supply the associated OIDC redirect URI, `https://consumer.mycompany.com/consumer-api/oauth2/redirect` for example. However, this information is not available until you have created your OIDC user registry in API Connect. You must therefore first register your application without this information, then update it later, as detailed in the instructions on this page.

### Logging in to the Management server

Before you can create an OIDC user registry, you must log in to your management server from the CLI. Use the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm admin/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

### Defining your OIDC registry configuration

You define the configuration of your OIDC user registry in a YAML file. As a minimum, the YAML file must have the following content:

```
title: registry_title
integration_url: oidc_integration_url
case_sensitive: case_sensitivity_setting
configuration:
  client_id: 'app_client_id'
  client_secret: 'my-client-secret'
  provider_type: oidc_provider_type
```

where:

- `registry_title` is your chosen descriptive title for the user registry.
- `oidc_integration_url` is the OIDC integration URL in your API Connect configuration. You can determine the OIDC integration URL by using the following CLI command:

```
apic integrations:list --server mgmt_endpoint_url --subcollection user-registry
```

- `case_sensitivity_setting` determines whether your user registry is case-sensitive. Valid values are:
  - `true`
  - `false`

To ensure proper handling of user name capitalization, you **must** ensure that your case-sensitivity setting here matches the setting on the backend OIDC provider:

- **Only** set `case_sensitive` to `true` if the backend OIDC provider supports case-sensitivity.
- Set `case_sensitive` to `false` if the backend OIDC provider does not support case-sensitivity.

Note: After at least one user has been onboarded into the registry, you cannot change this setting.

- `app_client_id` is the client ID of the application that is registered with the OIDC server, and must be in string format.
- `my-client-secret` is the client secret of the application that is registered with the OIDC server, and must be in string format.
- `oidc_provider_type` is the type of OIDC provider; specify one of the following values:
  - `facebook`
  - `github`
  - `google`
  - `linkedin`
  - `slack`
  - `twitter`
  - `windows_live`
  - `standard`

Use the `standard` provider type for any OIDC provider that is compliant with the OIDC standard.

Note: If the provider type is `standard`, you must include the following additional properties in the `configuration` section of your YAML file:

```
authorization_endpoint: 'oidc_auth_endpoint'
token_endpoint:
```



```
endpoint: 'oidc_token_endpoint'
```

where:

- `oidc_auth_endpoint` is the authorization endpoint on the OIDC server, and must be in string format.
- `oidc_token_endpoint` is the token endpoint on the OIDC server, and must be in string format.

## Default OIDC configurations

For each OIDC provider type, API Connect assumes a default configuration, but you can override the default configuration properties in your YAML file. The default configurations are as follows:

- Facebook

```
authorization_endpoint: 'https://www.facebook.com/v3.1/dialog/oauth'
token_endpoint:
  endpoint: 'https://graph.facebook.com/v3.1/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://graph.facebook.com/me'
scope: email public_profile
field_mapping:
  username: email
  email: email
  last_name: last_name
  first_name: first_name
```

- Github

```
authorization_endpoint: 'https://github.com/login/oauth/authorize'
token_endpoint:
  endpoint: 'https://github.com/login/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://api.github.com/user'
scope: 'read:user user:email'
field_mapping:
  username: login
  email: email
  last_name: name
  first_name: name
```

- Google

```
authorization_endpoint: 'https://accounts.google.com/o/oauth2/v2/auth'
token_endpoint:
  endpoint: 'https://www.googleapis.com/oauth2/v4/token'
scope: openid profile email
field_mapping:
  username: email
  email: email
  last_name: family_name
  first_name: given_name
```

- LinkedIn

```
authorization_endpoint: 'https://www.linkedin.com/oauth/v2/authorization'
token_endpoint:
  endpoint: 'https://www.linkedin.com/oauth/v2/accessToken'
userinfo_endpoint:
  endpoint: 'https://api.linkedin.com/v1/people/~:(id,first-name,last-name,picture-url,public-profile-url,email-address)?
format=json'
scope: r_basicprofile r_emailaddress
field_mapping:
  username: emailAddress
  email: emailAddress
  last_name: lastName
  first_name: firstName
credential_location: form_body
```

- Slack

```
authorization_endpoint: 'https://slack.com/oauth/authorize'
token_endpoint:
  endpoint: 'https://slack.com/api/oauth.access'
userinfo_endpoint:
  endpoint: 'https://slack.com/api/users.identity'
scope: identity.basic identity.email
field_mapping:
  username: user.email
  email: user.email
  last_name: user.name
  first_name: user.name
```

- Twitter:

```
request_endpoint: https://api.twitter.com/oauth/request_token'
authorization_endpoint: https://api.twitter.com/oauth/authenticate'
token_endpoint:
  endpoint: 'https://api.twitter.com/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://api.twitter.com/1.1/account/verify_credentials.json'
oauth_signature_method: 'HMAC-SHA1'
field_mapping:
  email: email
  first_name: name
  last_name: name
  username: screen_name
```

- WindowsLive

```
authorization_endpoint: 'https://login.microsoftonline.com/common/oauth2/v2.0/authorize'
token_endpoint:
  endpoint: 'https://login.microsoftonline.com/common/oauth2/v2.0/token'
scope: openid offline_access profile email
field_mapping:
  username: preferred_username
  email: email
  last_name: last_name
  first_name: first_name
```

- Standard

```
response_type: code
scope: openid
field_mapping:
  username: sub
  email: email
  last_name: family_name
  first_name: given_name
credential_location: auth_header
```

Although this is the default configuration for the **standard** provider type, you should contact your OIDC provider to obtain the details of the fields that you need to define.

## Creating your OIDC user registry

To create your OIDC user registry, use the following CLI command:

```
apic user-registries:create --server mgmt_endpoint_url --org admin oidc_config_file
```

where:

- *mgmt\_endpoint\_url* is the platform API endpoint URL.
- *organization\_name* is the value of the **name** property of your provider organization.
- *oidc\_config\_file* is the name of the YAML file that defines the configuration of your OIDC user registry.

On completion of the registry creation, the command displays the following summary details:

```
registry_name registry_url
```

By default, the *registry\_name* is derived from the **title** property in the configuration YAML file but you can override this by including a **name** property in the file. The *registry\_url* is an internal URL that API Connect assigns to the registry.

After you have created your OIDC user registry, you must update your application registration with the third party OIDC provider to include the OIDC redirect URI; you can obtain this information by using the following command, which displays the details of the registry in the command window:

```
apic user-registries:get --server mgmt_endpoint_url --org organization_name registry_name --output -
```

The required **oidc\_redirect\_uri** value is in the **consumer:** section; for example:

```
consumer:
  oidc_redirect_uri: https://consumer.mycompany.com/consumer-api/oauth2/redirect
```

## Enabling your OIDC registry in a Catalog

If you want to make your OIDC registry available for onboarding and authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. Complete the following steps:

1. Determine the URL of your OIDC user registry by using the following command:

```
apic user-registries:list --server mgmt_endpoint_url --org admin
```

2. Log in to the Management server as a member of a provider organization; enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the **apic login** command, see [Logging in to a management server](#).

3. Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic configured-catalog-user-registries:create --server mgmt_endpoint_url --org organization_name --catalog catalog_name -
```

where *catalog\_name* is the value of the **name** property of the required Catalog. The command returns

```
Reading CONFIGURED_CATALOG_USER_REGISTRY_FILE arg from stdin
```

4. Enter the following data, followed by a new line:

```
user_registry_url: oidc_registry_url
```

where *oidc\_registry\_url* is the URL of your OIDC registry, obtained in step [1](#).

5. Press **CTRL D** to terminate the input.

For details of all the **apic user-registries** and **apic configured-catalog-user-registries** commands, see [apic user-registries](#) and [apic configured-catalog-user-registries](#).

You can also complete the operations described in this topic by using the API Connect REST APIs; see the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting visibility for a user registry

The visibility setting determines which provider organizations can access a resource, for example, user registries.

### Before you begin

---

You must complete the following tasks:

- [Creating an Availability Zone](#)
- [Registering a gateway service](#)

### About this task


---

The user registry visibility setting controls which provider organizations are able to use that registry for authenticating Developer Portal users, or for securing APIs. For details on how to specify the user registries that can be used for authenticating users of the Cloud Manager and API Manager user interfaces, see [Selecting user registries for Cloud Manager and API Manager](#).

### Procedure

---

Follow these steps to set the visibility for the resources (user registries) in your on-premises cloud:

1. In the Cloud Manager, click  Resources.
2. Select User Registries as the resource to view.
3. From the list of user registries, choose Edit visibility from the actions menu next to the name of the user registry you are working with.
4. Select the visibility setting for the user registry. The options are:
  - Private - the user registry is not visible and cannot be used by any provider organization
  - Public - the user registry is visible and can be used by all provider organizations
  - Custom - the user registry is visible only to the provider organizations designated by you
5. For Custom visibility, select the provider organizations that you want to have access to the user registry.
6. Click Make visible to complete the operation.

### Results

---

For Private, the user registry cannot be used by any provider organizations. For Public, the user registry can be used by all provider organizations. For Custom, the user registry can be used by the provider organizations that you designate.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting a user registry

As the Cloud Manager administrator, you can delete user registries.

### About this task

---


Complete the following steps to delete a user registry. All users within the user registry will be deleted. The pre-configured user registries (API Manager Local User Registry and Cloud Manager Local User Registry) may not be deleted. Any user registry marked as the Default in Settings > User Registries also may not be deleted.

One of the following roles is required to delete a user registry:

- Administrator
- Owner

### Procedure

---

1. In the Cloud Manager, click .
2. Select User Registries to see the list of current user registries in your cloud.
3. Choose Delete from the overflow menu that is adjacent to the user registry you want to delete.
4. Optionally, you can delete multiple user registries by marking the checkboxes and selecting Delete Selected User Registries from the overflow menu for the list.
5. Confirm the deletion.

### Results

---

The user registry is deleted and removed from the User Registries list in Cloud Manager. All user accounts contained in the user registry are deleted.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing a user from a user registry

As an API Connect user, you can remove yourself from an API Connect user registry. As an administrative user, you can remove other users from an API Connect user registry.

For details on how to remove a user from a user registry, see the following subtopics:

- [Removing yourself from a user registry](#)  
As an API Connect user, you can remove yourself from the API Connect user registry in which you were registered, by using the developer toolkit CLI.
- [Removing another user from a user registry](#)  
You can remove a user from an API Connect user registry by using the developer toolkit CLI.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing yourself from a user registry

As an API Connect user, you can remove yourself from the API Connect user registry in which you were registered, by using the developer toolkit CLI.

---

### Before you begin

You cannot remove yourself from a user registry if you are a member of a provider organization. Before removing yourself from a user registry, ensure that you have been deleted from any provider organizations. For more information, see [Removing a user from an organization](#).

---

### Procedure

1. Log in to the management server as the user that you want to remove.  
The format of the login command depends on whether you are a user in the Cloud Manager admin organization, or you are a user in a provider organization.

If you are a Cloud Manager admin user, enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm admin/identity_provider
```

If you are a provider organization user, enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

Note: If the same user registry is used for both the Cloud Manager and API Manager user interfaces, and you have access to both, when you remove yourself from the user registry you will lose access to both user interfaces regardless of which organization you log in to.

2. Remove yourself from the user registry. Enter the following command:

```
apic me:delete --server mgmt_endpoint_url
```

For example:

```
apic me:delete --server platform-api.myserver.com.com
```

The command confirms successful removal by returning the details of the deleted user; for example:

```
user1 [state: enabled] https://platform-api.myserver.com.com/api/user-registries/32830897-1d23-4fac-acf5-0193d0b2c1b5/4438937a-6ad0-4eaa-9163-820888ac6245/users/040adb11-e9a4-4d93-9c2e-62a974da0689
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing another user from a user registry

You can remove a user from an API Connect user registry by using the developer toolkit CLI.

---

### Before you begin

To complete this task, you must have Administrator access to your API Connect cloud.

You cannot remove a user from a user registry if that user is a member of a provider organization. Before removing a user from a user registry, ensure that they have been deleted from any provider organizations. For more information, see [Removing a user from an organization](#).

## Procedure

1. Log in to the management server as a member of the Cloud Manager admin organization. Enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm admin/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

Note: If the same user registry is used for both the Cloud Manager and API Manager user interfaces, and the user has access to both, when you remove them from the user registry they will lose access to both even though you are logging in to the admin organization.

2. Identify the name of the user registry from which you want to remove the user. Enter the following command:

```
apic user-registries:list --server mgmt_endpoint_url --org admin
```

For example:

```
apic user-registries:list --server platform-api.myserver.com.com --org admin
```

The command returns a list of all user registries, with the registry name displayed first; for example:

```
api-manager-lur      https://platform-api.myserver.com.com/api/user-registries/3283e897-1d23-4fac-acf5-0193d0b2c1b5/4438937a-6ad0-4eaa-9163-820888ac6245
cloud-manager-lur   https://platform-api.myserver.com.com/api/user-registries/3283e897-1d23-4fac-acf5-0193d0b2c1b5/3adbf524-cd74-4051-99c0-89ce5ffcc9c0
my-ldap             https://platform-api.myserver.com.com/api/user-registries/3283e897-1d23-4fac-acf5-0193d0b2c1b5/29eae413-cd74-4051-99c0-89ce5ffcc9c0
```

3. Identify the name of the user, from the required user registry, that you want to remove. Enter the following command:

```
apic users:list --server mgmt_endpoint_url --org admin --user-registry user_registry_name
```

For example:

```
apic users:list --server platform-api.myserver.com.com --org admin --user-registry my-ldap
```

The command returns a list of all users in the user registry, with the user name displayed first; for example:

```
user1 [state: enabled] https://platform-api.myserver.com.com/api/user-registries/32830897-1d23-4fac-acf5-0193d0b2c1b5/4438937a-6ad0-4eaa-9163-820888ac6245/users/040adb11-e9a4-4d93-9c2e-62a974da0689
user2 [state: enabled] https://platform-api.myserver.com.com/api/user-registries/32830897-1d23-4fac-acf5-0193d0b2c1b5/4438937a-6ad0-4eaa-9163-820888ac6245/users/12f1aa82-d9f0-4670-99f1-7500d1fb6583
```

4. Remove the required user from the user registry. Enter the following command:

```
apic users:delete user_name --server mgmt_endpoint_url --org admin --user-registry user_registry_name
```

For example:

```
apic users:delete user1 --server platform-api.myserver.com.com --org admin --user-registry my-ldap
```

The command confirms successful removal by returning the details of the deleted user; for example:

```
user1 [state: enabled] https://platform-api.myserver.com.com/api/user-registries/32830897-1d23-4fac-acf5-0193d0b2c1b5/4438937a-6ad0-4eaa-9163-820888ac6245/users/040adb11-e9a4-4d93-9c2e-62a974da0689
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## TLS profiles overview

API Connect supports TLS Profiles for securing data transmission over HTTPS.

### Introduction to TLS profiles

Important: API Connect includes several default TLS profiles to help you get started working with the application. The default profiles should not be used in a production environment. It is important to create your own profiles to ensure a secure network.

API Connect may need to transmit data across an untrusted network, for example, when accessing the Gateway, email server, or LDAP server. TLS provides secure network layer transportation of data between two parties.

There are two types of TLS Profiles: a TLS Server Profile and a TLS Client Profile. A TLS Server Profile is used by the Gateway to configure its endpoint for use during API execution. A TLS Client profile is used whenever the system needs to communicate with another endpoint over TLS.

The components of a TLS Profile are:

- TLS Protocol version indicates the versions of the Transport Layer Security Protocol required for the profile. TLS Protocol versions 1.0, 1.1, and 1.2 are supported.
- Optional support for mutual authentication and renegotiation for Server Profiles.
- Optional support for weak server connections and Server Name Indication for Client Profiles.

- Cipher suites to secure HTTPs communication within the API Connect ecosystem.
- Keystores containing public and private key pairs.
- Truststores containing public keys for trusted third party services, such as Google, Facebook, or Verisign.
- [Viewing TLS Profiles, Keystores, and Truststores](#)  
The current list of TLS profiles, Keystores, and Truststores can be viewed on the main TLS screen.
- [Creating a TLS Server Profile](#)  
In Cloud Manager, you can configure the profile that is used on the gateway when it acts as a TLS server.
- [Creating a TLS Client Profile](#)  
In Cloud Manager, TLS profiles are configured as a Resource to provide secure transmission of data over HTTPs.
- [Creating a Keystore](#)  
Each keystore contain a matched pair of a public certificates and its private keys. These artifacts provide identity information during a TLS handshake.
- [Creating a Truststore](#)  
A truststore contains a list of certificates. The certificates are used to verify the peer during a TLS handshake.
- [Viewing certificate details and adding certificates to a keystore or truststore](#)  
You can view details for the certificates in an existing keystore or truststore and add additional certificates.
- [Generating a self-signed certificate using OpenSSL](#)  
OpenSSL is an open source implementation of the SSL and TLS protocols. It provides the transport layer security over the normal communications layer, allowing it to be intertwined with many network applications and services.
- [Generating a PKCS#12 file for Certificate Authority](#)  
PKCS#12 (P12) files define an archive file format for storing cryptographic objects as a single file. API Connect supports the P12 file format for uploading a keystore and truststore. The keystore should contain both a private and public key along with intermediate CA certificates.
- [Binding a TLS server profile to a gateway service](#)  
You can bind the SSL certificate of a TLS profile to a gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.
- [Updating the PKCS#12 certificate for a TLS server profile](#)  
A server certificate bound to a gateway service can be invalidated if the host name in the digital certificate of the server does not match the URL specified by the client, or because it has expired. When this happens, you must update the TLS profile with a new CA certificate or PKCS#12 (P12) file.

## Related tasks

---

- [Creating a TLS Server Profile](#)
- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Viewing TLS Profiles, Keystores, and Truststores

The current list of TLS profiles, Keystores, and Truststores can be viewed on the main TLS screen.

## Before you begin

---

Important: API Connect includes several default TLS profiles to help you get started working with the application. The default profiles should not be used in a production environment. It is important to create your own profiles to ensure a secure network.

One of the following roles is required to view TLS Server Profiles:

- Administrator
- Owner
- Topology Administrator
- Viewer
- Custom role with the `Settings: Manage or Settings: view` permissions

## About this task

---

The TLS list screen provides a list of TLS Profiles, Keystores and Truststores configured for your cloud. Users with view permissions can use this screen as a reference for the TLS Profiles.


Users with manage permissions can create new Profiles, Keystores, and Truststores; edit existing ones; and delete items that are no longer needed. Users can view the Keystores and Truststores assigned to the profiles.




Users can view and set the visibility for Client Profiles. For more information on visibility, see [Setting visibility for a TLS Client Profile](#)

## Procedure

---

Following are the tasks you can perform using the list screen as a base:

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Complete your management task:

- To create a new item, click Create next to the table for the type of item you want to create.
- To edit an existing item, either click the Title of the item, or click the options menu  and then click Edit.
- To set the visibility of a TLS Client Profile, click the options menu  for the profile, and then click Edit Visibility.
- To delete an existing item, click the options menu  for the profile, and then click Delete.

## Related concepts

- [TLS profiles overview](#)

## Related tasks

- [Creating a TLS Server Profile](#)
- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Creating a TLS Server Profile

In Cloud Manager, you can configure the profile that is used on the gateway when it acts as a TLS server.

## Before you begin

Important: API Connect includes several default TLS profiles to help you get started. The default profiles should not be used in a production environment. It is important to create your own profiles to secure your network.

One of the following roles is required to configure TLS Server Profiles:


- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: manage permissions`

## About this task

The Server profile is for the gateway when it is acting as the TLS server.

## Procedure

Perform the following steps to create a TLS Server profile:

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Click Create in the TLS Server Profile table.
4. Enter the fields to configure the TLS Server Profile:

| Field                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)                 | Enter a Title for the profile. The title is displayed on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Name (required)                  | The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage TLS Server Profiles, see <a href="#">apic tls-server-profiles</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Version (required)               | Assign a version number for the profile. Using version numbers allows you to create multiple server profiles with the same name and different configurations, for example, <i>MyProfile 1.0</i> and <i>MyProfile 1.1</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Summary (optional)               | Enter a description of the profile.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Protocols (required)             | Select one or more supported TLS protocol versions. The default is 1.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Mutual Authentication (required) | Determines the level of two-way authentication for the server profile. In two-way authentication, the server responds to a client by sending a request for the client certificate. <ul style="list-style-type: none"> <li>• None (default) No support for mutual authentication.</li> <li>• Request Enable this option to request client authentication during the TLS handshake. When the application sends the request, the gateway requests that the application sends the certificate. If the client does not send the certificate, the certificate is not checked on the gateway.</li> <li>• Require Enable this option to require client authentication during the TLS handshake. When the application sends the request, the gateway requests that the application sends the certificate. If the client does not send the certificate, the TLS handshake fails and the request is blocked.</li> </ul> |
| Limit Renegotiation (optional)   | Client-initiated renegotiation allows the connection to be retried. The default is to prevent renegotiation. Remove the checkmark to allow renegotiation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Field                 | Description                                                                                                                                                                                                                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keystore (required)   | A keystore is a repository containing a public and private key pair. The Server Profile requires a keystore in order to securely identify the system. When an application sends an API request, the keystore is used to verify a matching certificate.                                                                    |
| Truststore (optional) | A truststore is a repository containing certificates. The certificates are used to verify the peer during a TLS handshake. If, in addition to a keystore, a truststore is specified, the certificate is further checked for validity by ensuring that is signed by the root certificate, which must be in the truststore. |
| Ciphers (required)    | Select the ciphers for the profile.                                                                                                                                                                                                                                                                                       |

5. Click Save.

## Related concepts

- [TLS profiles overview](#)

## Related tasks

- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a TLS Client Profile

In Cloud Manager, TLS profiles are configured as a Resource to provide secure transmission of data over HTTPs.

### Before you begin

One of the following roles is required to configure TLS Profiles:


- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

### About this task

API Connect uses both TLS Server and TLS Client profiles. A TLS Server profile is presented when a communication request is received. The Server profile validates the request against the Keystore and protocol version to determine whether the connection is secure. The Client profile is presented to initiate communication with another system. Client Profiles may be made visible for use by provider organizations in policies in API Manager. See [Setting visibility for a TLS Client Profile](#).

### Procedure

Perform the following steps to create a TLS Client profile:

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Click Create in the TLS Client Profile table.
4. Enter the fields to configure the TLS Client Profile:

| Field                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)             | Enter a Title for the profile. The title is displayed on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Name (required)              | The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage a TLS Client Profile, see <a href="#">apic tls-client-profiles</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Version (required)           | Assign a version number for the profile. Using version numbers allows you to create multiple server profiles with the same name and different configurations, for example, <i>MyProfile 1.0</i> and <i>MyProfile 1.1</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Summary (optional)           | Enter a description of the profile.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Protocols (required)         | Select one or more supported TLS protocol versions. The default is 1.2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Server Connection (optional) | Specify whether to support weak or insecure credentials. <ul style="list-style-type: none"> <li>• Allow insecure server connections - Insecure server connections may result from self-signed certificates, expired or corrupted certificates, or certificates from an unknown or untrusted source. Check this box to allow the connection to proceed with an insecure connection. The default is to not allow insecure server connections.</li> <li>• Support Server Name Indication (SNI) - Check this box to enable SNI. SNI allows support for multiple certificates presented on the same IP address using different host names. The client profile sends the name of a virtual domain as part of the TLS negotiation. The default is to enable SNI.</li> </ul> |



| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keystore (optional)   | A Keystore is a repository containing public and private key pairs. Select the keystore where you will store the certificates for the profile. Default keystores are provided, and you can also create your own.                                                                                                                                                            |
| Truststore (optional) | A Truststore is a repository containing verified public keys, which are usually obtained from a third-party certificate authority. Truststores provide secure identification for peer systems. A truststore is usually used when mutual authentication is enabled. Select a truststore for the profile. Default truststores are provided, and you can also create your own. |
| Ciphers (required)    | Cipher suites are encryption/decryption algorithms used to secure HTTPs communication within the API Connect ecosystem. All ciphers are enabled by default. You can unselect one or more ciphers if they are not needed for the profile.                                                                                                                                    |

5. Click Save.

- [Setting visibility for a TLS Client Profile](#)

The visibility setting determines which provider organizations can access a resource.

- [Defining elliptic curve cryptographic schemes for a TLS client profile](#)

You define the elliptic curve cryptographic schemes for a TLS client profile by using the developer toolkit CLI.

## Related concepts

- [TLS profiles overview](#)

## Related tasks

- [Setting visibility for a TLS Client Profile](#)
- [Creating a TLS Server Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting visibility for a TLS Client Profile

The visibility setting determines which provider organizations can access a resource.

### Before you begin

You must complete the following task:



- [Creating a TLS Client Profile](#)

### About this task

The visibility setting gives you control over the resources that are available to provider organizations.

### Procedure

Follow these steps to set the visibility for the resources in your on-premises cloud:

1. In the Cloud Manager, click  Resources.
2. Select TLS as the resource to view.
3. From the table of TLS Client Profiles, choose Edit visibility from the  options menu next to the name of the profile you are managing.
4. Select the visibility setting for the profile. The options are:
  - Private - the profile is visible only in Cloud Manager and cannot be used by any provider organization
  - Public - the profile is visible and can be used by all provider organizations
  - Custom - the profile is visible only to the provider organizations designated by you
5. For Custom visibility, select the provider organizations that you want to have access to the profile.
6. Click Save to complete the operation.

### Results

For Private, the profile cannot be used by any provider organizations. For Public, the profile can be used by all provider organizations. For Custom, the profile can be used by the provider organizations that you designate. If visible, the profiles will be available in API Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Defining elliptic curve cryptographic schemes for a TLS client profile

You define the elliptic curve cryptographic schemes for a TLS client profile by using the developer toolkit CLI.

## About this task

To define elliptic curve cryptographic schemes for a TLS client profile, you include `elliptic_curve_auto_negotiation` and `elliptic_curve` properties in a YAML file definition for the TLS client profile. The `elliptic_curve` property lists the required elliptic curve cryptographic schemes. For example:

```
elliptic_curve_auto_negotiation: false
elliptic_curve:
  - secp521r1
  - secp384r1
  - prime256v1
```

You then use the developer toolkit CLI to create the TLS client profile in API Connect.

When `elliptic_curve_auto_negotiation` is set to `true`, the system negotiates the Elliptic-curve Diffie-Hellman (ECDH) key agreement automatically with its peer, and any `elliptic_curve` property settings are ignored.

The following example shows a complete YAML file for a TLS client profile:

```
type: tls_client_profile
name: my-tls-client-profile
version: 1.0.0
title: My TLS client profile
protocols:
  - tls_v1.2
ciphers:
  - ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
elliptic_curve_auto_negotiation: false
elliptic_curve:
  - sect163k1
insecure_server_connections: false
server_name_indication: true
```

Note:

- The `elliptic_curve_auto_negotiation` option is not supported by any of the API Connect gateway types. If the TLS client profile is targeted for an API Connect gateway; this setting is ignored by the gateway.
- The elliptic curve cryptographic schemes shown in each row of the following table are equivalent. However, API Connect recognizes only one or the other depending on how the TLS profile is used, as indicated in the table.

Table 1.

| API enforcement on the gateway | API Connect server access security |
|--------------------------------|------------------------------------|
| secp192r1                      | prime192v1                         |
| secp256r1                      | prime256v1                         |

Therefore, if you want to use either of these schemes and are unsure whether you are targeting the TLS client profile to the API Connect gateway for API enforcement, whether you are using it to secure user access to the API Connect servers, or whether it will be used for both purposes, specify both equivalent schemes; API Connect will simply ignore the non-relevant scheme. For example:

```
elliptic_curve:
  .
  .
  .
  - secp256r1
  - prime256v1
  .
  .
  .
```

## Procedure

To create a TLS client profile with elliptic curve cryptographic schemes defined, complete the following steps:

- Create a YAML file definition for your TLS client profile, with the required `elliptic_curve` property.
- Log in to the management server from the developer toolkit CLI. Log in either as a member of the cloud administration organization or as a member of a provider organization, depending on where you want to create the TLS client profile. For details, see [Logging in to a management server](#).
- Create the TLS client profile by using the following command:

```
apic tls-client-profiles:create --server mgmt_endpoint_url --org organization_name tls_client_profile_yaml_file
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL, and is the same as that which was used when you logged in at step 2.
- `organization_name` is either `admin`, for the cloud administration organization, or the name of your provider organization, and is the same as that which was used when you logged in at step 2.
- `tls_client_profile_yaml_file` is the name of the YAML file that contains the definition for your TLS client profile.

Note: When you install IBM® API Connect, the API Connect gateway has a pre-supplied default TLS client profile that is used for API enforcement if you do not configure a TLS client profile; you cannot configure this default TLS client profile on the gateway.

For reference details of all the `apic tls-client-profiles` commands, see [apic tls-client-profiles](#).

You can also complete the operations described in this topic by using the API Connect REST APIs; see the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Keystore

Each keystore contains a matched pair of a public certificate and its private keys. These artifacts provide identity information during a TLS handshake.

### Before you begin

Cloud Manager and API Manager both support and use TLS certificates, but they do not themselves produce strong encryption keys or manage your encryption keys. Encryption keys are generated and managed according to your own procedures. For more information, see [Generating a PKCS#12 file for Certificate Authority](#) and [Generating a self-signed certificate using OpenSSL](#).

One of the following roles is required to configure Keystores:


- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

### About this task

API Connect includes pre-configured Keystores which may be used for testing purposes. For production environments, we suggest creating a new, secure Keystore.

### Procedure

Perform the following steps to create a TLS Client profile:

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Click Create in the Keystore table.

| Field                                                   | Description                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)                                        | Enter a Title for the Keystore. The title is displayed on the screen.                                                                                                                                                                                                                       |
| Name (required)                                         | The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage keystores, see <a href="#">apic keystores</a> .                                                                           |
| Summary (optional)                                      | Enter a brief description.                                                                                                                                                                                                                                                                  |
| Private Key & Public Key:<br>Step 1: Upload private key | Upload the file containing the private key certificate. If necessary, you can click Browse to locate the file. If the file contains both the private and public keys, upload it in Step 1. Private and public keys are always uploaded in pairs, either in a single file or separate files. |
| Private key password (optional)                         | Enter the password for the private key if it has a password.                                                                                                                                                                                                                                |
| Private Key & Public Key:<br>Step 2: Upload public key  | If the public key is contained in a separate file, upload it in Step 2. Private and Public keys are always uploaded in pairs, either in a single file or separate files.                                                                                                                    |

4. Click Save.

Note: After they have been uploaded, private keys cannot be downloaded from API Connect.

### Related concepts

- [TLS profiles overview](#)

### Related tasks

- [Creating a TLS Client Profile](#)
- [Creating a Truststore](#)
- [Generating a PKCS#12 file for Certificate Authority](#)
- [Generating a self-signed certificate using OpenSSL](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Truststore

A truststore contains a list of certificates. The certificates are used to verify the peer during a TLS handshake.

## Before you begin

---

One of the following roles is required to configure Truststores:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task


---

Cloud Manager and API Manager both support and use TLS certificates, but they do not themselves produce strong encryption keys or manage your encryption keys. Encryption keys are generated and managed according to your own procedures. For more information, see [Generating a PKCS#12 file for Certificate Authority](#) and [Generating a self-signed certificate using OpenSSL](#).

API Connect includes pre-configured Truststores which may be used for testing purposes. For production environments, we suggest creating a new, secure Truststore.

## Procedure

---

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Click Create in the Truststore table.

| Field              |                                                                                                                                                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title (required)   | Enter a Title for the Truststore. The title is displayed on the screen.                                                                                                                                               |
| Name (required)    | The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage truststores, see <a href="#">apic truststores</a> . |
| Summary (optional) | Enter a brief description.                                                                                                                                                                                            |
| Public Keys        | Upload the file containing the public key certificate. If necessary you can click Browse to locate the file.                                                                                                          |

4. Click Save.

## Related concepts

---

- [TLS profiles overview](#)

## Related tasks

---

- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Viewing certificate details and adding certificates to a keystore or truststore

You can view details for the certificates in an existing keystore or truststore and add additional certificates.

## Before you begin

---

One of the following roles is required to edit Keystores and Truststores:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task


---

API Connect includes pre-configured Keystores and Truststores which you can use for testing purposes. For production environments, we suggest creating a new, secure Keystore or Truststore. From an existing Keystore or Truststore, you can edit the title, view the details of the certificates, and add new certificates.

## Procedure

---

To work with existing keystores and truststores:

1. In the Cloud Manager, click  Resources.
2. Select TLS.
3. Click the name of the keystore or truststore.

4. Edit the name and summary if needed.
5. The Subject, Finger Print, and Expiration date is shown for each certificate in the keystore or truststore. Click > to view the certificate details.
6. Add additional private and/or public keys if needed.
7. Click Save.

Note: After they have been uploaded, private keys cannot be downloaded from API Connect.

## Related concepts

---

- [TLS profiles overview](#)

## Related tasks

---

- [Creating a TLS Server Profile](#)
- [Creating a TLS Client Profile](#)
- [Creating a Truststore](#)
- [Creating a Keystore](#)
- [Generating a PKCS#12 file for Certificate Authority](#)
- [Generating a self-signed certificate using OpenSSL](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Generating a self-signed certificate using OpenSSL

OpenSSL is an open source implementation of the SSL and TLS protocols. It provides the transport layer security over the normal communications layer, allowing it to be intertwined with many network applications and services.

## Before you begin

---

One of the following roles is required to complete this task:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task

---

This topic tells you how to generate a self-signed SSL certificate request using the OpenSSL toolkit to enable HTTPS connections.

## Procedure

---

To generate a self-signed SSL certificate using the OpenSSL, complete the following steps:

1. Write down the Common Name (CN) for your SSL Certificate. The CN is the fully qualified name for the system that uses the certificate. For static DNS, use the hostname or IP address set in your Gateway Cluster (for example, `192.16.183.131` or `dp1.acme.com`).
2. Run the following OpenSSL command to generate your private key and public certificate. Answer the questions and enter the Common Name when prompted.

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem
```

3. Review the created certificate:

```
openssl x509 -text -noout -in certificate.pem
```


4. Combine your key and certificate in a PKCS#12 (P12) bundle:

```
openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12
```

5. Validate your P2 file.

```
openssl pkcs12 -in certificate.p12 -noout -info
```

Once the certificate file is created, it can be uploaded to a keystore.

6. In the Cloud Manager, click  Resources.
7. Select TLS.
8. Click Create in the Keystore table.
9. Create a Keystore and upload the certificate file following the instructions at [Creating a Keystore](#).

Note:

- API Connect supports only the P12 (PKCS12) format file for the present certificate.
- Your P12 file must contain the private key, the public certificate from the Certificate Authority, and all intermediate certificates used for signing.
- Your P12 file can contain a maximum of 10 intermediate certificates.

10. Click Save.

## Related concepts

- [TLS profiles overview](#)

## Related tasks

- [Creating a TLS Server Profile](#)
- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Generating a PKCS#12 file for Certificate Authority

PKCS#12 (P12) files define an archive file format for storing cryptographic objects as a single file. API Connect supports the P12 file format for uploading a keystore and truststore. The keystore should contain both a private and public key along with intermediate CA certificates.

### Before you begin

One of the following roles is required to add a key to a keystore or truststore:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

Before you can generate a P12 file, you must have a private key (for example: `key.pem`), a signed certificate by a Certificate Authority (for example `certificate.pem`) and one or more certificates from the CA authority.

Note: If your certificate file contains more than one certificate, you must manually split the file and create a single file for each entry. Each entry must be bound by the following markers:

```
-----BEGIN CERTIFICATE-----  
-----END CERTIFICATE-----
```

### Procedure

1. If you have intermediate certificates from your CA, concatenate them into a single `.pem` file to build your `caChain`. Be sure to enter a new line following each certificate's data.


```
cat ca1.pem ca2.pem ca3.pem > caChain.pem  
cat caChain.pem  
-----BEGIN CERTIFICATE-----  
MIIEPjCCA46gAwIBAgIQEod26KZabjd+BQMG1Dw16jANBgkqhkiG9w0BAQUFADCB  
...  
lQX7CkTJn61AJUsyEa8H/gjVQnHp4VOLFR/dKgeVcCRvZF7Tt5AuiyHY  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
MIIEPDCCAYsGawIBAgIQSEus8arH1xND0aJ0NumXJTANBgkqhkiG9w0BAQUFADBv  
...  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
MIIEjCCAyx6gAwIBAgIBATANBgkqhkiG9w0BAQUFADBvMQswCQYDVQQGEWJTRTEU  
...  
-----END CERTIFICATE-----
```

2. Create the P12 file including the private key, the signed certificate and the CA file you created in step 1, if applicable. Omit the `-CAfile` option if you don't have CA certificates to include.

The following command uses OpenSSL, an open source implementation of the SSL and TLS protocols.

```
openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12 -CAfile caChain.pem -chain
```

Once the certificate file is created, it can be uploaded to a keystore.

3. In the Cloud Manager, click  Resources.

4. Select TLS.

5. Click Create in the Keystore table.

6. Create a Keystore and upload the certificate file following the instructions at [Creating a Keystore](#).

Note:

- API Connect supports only the P12 (PKCS12) format file for the present certificate.
- Your P12 file must contain the private key, the public certificate from the Certificate Authority, and all intermediate certificates used for signing.
- Your P12 file can contain a maximum of 10 intermediate certificates.

7. Click Save.

## Related concepts

---

- [TLS profiles overview](#)

## Related tasks

---

- [Creating a TLS Server Profile](#)
- [Creating a TLS Client Profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)
- [Generating a self-signed certificate using OpenSSL](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Binding a TLS server profile to a gateway service

You can bind the SSL certificate of a TLS profile to a gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.

### Before you begin

---

One of the following roles is required:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Topology: Manage permissions`

For more information on which user roles have access, see [Adding members to the admin organization](#).


### About this task

---

Perform the following steps to bind a TLS profile to a gateway service.

### Procedure

---

1. In the Cloud Manager user interface, click  Topology, then select the gateway service that you want to work with.
2. Scroll down to the API Invocation Endpoint section, and under Server Name Indication (SNI) select the required TLS server profile.  
You can also bind a TLS server profile during the initial registration of a gateway service; see [Registering a gateway service](#).
3. Click Save when done.

### Results

---

The TLS profile binds to the gateway service.

### What to do next

---

If you want to change the TLS Profile from Cloud Manager, for SNI Mapping of API Invocation endpoint, it is automatically changed in the associated Gateway. You do not need to do a removal or re-registration of the Gateway service.

### Related tasks

---

- [Creating a TLS Server Profile](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Updating the PKCS#12 certificate for a TLS server profile

A server certificate bound to a gateway service can be invalidated if the host name in the digital certificate of the server does not match the URL specified by the client, or because it has expired. When this happens, you must update the TLS profile with a new CA certificate or PKCS#12 (P12) file.

## Before you begin

---

One of the following roles is required:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task

---


If the expiration date of a certificate or a P12 file is approaching, or if a certificate is invalidated, use the steps in this topic to update a TLS profile bound to a gateway service. CA certificate and P12 file expiration dates are displayed in the details of the containing keystore; see step 3.

You update the certificate for a TLS server profile by replacing the certificate in the keystore that is associated with the TLS server profile.

Complete the following steps to update a TLS profile that has an invalidated or expired certificate or P12 file. After you have uploaded the new certificate, you must remove and re-add the associated gateway service.

## Procedure

---

1. In the Cloud Manager user interface, click  Resources, then click TLS.
2. To identify the keystore that is associated with the TLS server profile, complete the following steps:
  - a. In the TLS Server Profile section, select the required profile.
  - b. In the Keystore/Truststore section, note the selected keystore, then click Cancel to close the TLS server profile details page.
3. In the Keystore section, select the required keystore.  
The Certificates section displays the expiration date of a certificate. You can expand a certificate to see further details.
4. Click Browse and select the required P12 file.  
Note:
  - API Connect supports only the P12 (PKCS12) format file for the present certificate.
  - Your P12 file must contain the private key, the public certificate from the Certificate Authority, and all intermediate certificates used for signing.
  - Your P12 file can contain a maximum of 10 intermediate certificates.
5. Click Save when done.

## Related tasks

---

- [Creating a TLS Server Profile](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## OAuth Provider overview

API Connect supports OAuth Specification 2.0, for both Native and Third party implementations.

## Introduction to OAuth

---

OAuth is a token-based authorization protocol that allows third-party websites or applications to access user data without requiring the user to share personal information. In API Connect, you can secure an API with OAuth.

In Cloud Manager, you configure both Native and Third party OAuth providers that can be made visible to selected Provider organizations. The OAuth Provider configuration is based on the OAuth 2.0 Specification, which is available at <https://tools.ietf.org/html/rfc6749>. Knowledge of the OAuth 2.0 specification is required to implement an OAuth Provider in API Connect.

One of the following roles is required to configure OAuth Providers:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage permissions`

Note: In a multi-node cluster, OAuth operations will fail if quorum is lost. Quorum requires that the number of active nodes is greater than 50% of the total number of nodes in the cluster.

- [Configuring a native OAuth provider](#)  
Native OAuth providers are configured and managed by you within your cloud.
- [Configuring a third-party OAuth provider](#)  
Enter the secure endpoints to provide OAuth authentication from a third party.
- [Setting the visibility for OAuth providers](#)  
The visibility setting determines which provider organizations can use an OAuth provider to secure APIs.
- [OAuth concepts for API Connect](#)




**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring a native OAuth provider

Native OAuth providers are configured and managed by you within your cloud.

### About this task

A native OAuth provider object provides settings for OAuth processing operations such as generating and validating OAuth tokens. An OAuth provider object is referenced by an OAuth security definition to protect an API. When a native OAuth provider is used, the OAuth operations are performed natively by API Connect.

Every OAuth provider object has a backing API. Your configuration here automatically updates the OpenAPI document of the API. You can edit the OpenAPI document directly by navigating to the  Resources > OAuth Providers page, selecting your OAuth provider, then clicking API Editor.

Note: Take care when modifying the code directly on the Source tab of the API Editor because validation is limited. For example:

- If you change the name of auto generated assembly actions in the source code, the assembly will be prevented from updating dynamically when the OAuth provider settings are modified.
- You must ensure that the OAuth provider name matches the value specified in the `oauth-provider-settings-ref` field in each OAuth assembly action.

When a published API references an OAuth provider object, the backing API is automatically made available in the gateway.

One of the following roles is required to configure a native OAuth provider:


- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

Note:

- The OAuth provider logs Analytics data for failure cases, but does not log successful cases. Activity log policies that call for logging of Analytics data upon success do not apply for the OAuth provider.
- You must ensure the OAuth Provider is configured in the Sandbox Catalog before using the OAuth Provider in a non-Sandbox Catalog.


### Procedure

OAuth provider configuration uses a series of screens. The first set of screens controls a basic OAuth provider configuration. Perform the following steps to configure a native OAuth Provider for your cloud:

1. In the Cloud Manager, click  Resources.
2. Click OAuth Providers > Add > Native OAuth provider.
  - a. Complete the following parameters for the first screen, then click Next.

| Field                  | Description                                                                                                                                                                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                  | Enter a title for the native OAuth provider.                                                                                                                                                                                                                                                                                                        |
| Name                   | This field is auto-populated by the system.                                                                                                                                                                                                                                                                                                         |
| Description (optional) | Enter a brief description.                                                                                                                                                                                                                                                                                                                          |
| Base path (optional)   | The base path is the URL segment of the API that is shared by all operations in the API. It does not include the host name or any additional segments for paths or operations. The base path must be unique for a given catalog. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty. |
| Gateway Type           | Select the gateway type, either DataPower® Gateway (v5 compatible) or DataPower API Gateway. For information about types of gateways, see <a href="#">API Connect gateway types</a> .<br><br>OAuth Providers apply to one gateway type.                                                                                                             |

- b. In the next screen, enter the following additional configuration parameters, then click Next.

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authorize Path         | <code>/oauth2/authorize/</code> is the standard OAuth endpoint to login to account                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Token Path             | <code>/oauth2/token/</code> is the standard OAuth endpoint to exchange code for access token.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Supported grant types  | <ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type "Client Credentials." Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type "Authorization Code."</li> <li>• Resource owner - Password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> <li>•  Resource owner - JWT - A verified signed JSON Web Token is exchanged directly for an access token.</li> </ul> |
| Supported client types | <ul style="list-style-type: none"> <li>• Confidential - Client can maintain secure credentials on a secure server</li> <li>• Public - Client credentials are not secure. (Public is not available for DataPower API Gateway at this time.)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

- c. Enter the scopes in the next screen. A scope becomes an option in the request and response for an access token. Click Add to add additional fields for scopes. Click Next when done.

| Field             | Description     |
|-------------------|-----------------|
| sample_scope_1    | Scope for token |
| additional scopes | Scope for token |

- d. Enter the parameters for User Security in the next screen. Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization. User Security is not required for the Application grant type. Click Next when done.

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identity Extraction | <p>Determines how the user credential is extracted:</p> <ul style="list-style-type: none"> <li>Basic Authentication - HTTP basic authentication (requires no additional configuration)</li> <li>Default HTML Form - Use default login form for user name and password</li> <li><input type="checkbox"/> API Gateway only Context variable - Specify which variable contains the user name and password. API Connect OAuth context variables as listed here <a href="#">API Connect context variables</a><br/>Note: DataPower API Gateway only. Not available for DataPower Gateway (v5 compatible).</li> <li>Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML login form for user security</a>.</li> <li>Redirect - Enter an endpoint to redirect to a third-party identity provider. For more information, see <a href="#">Authenticating and authorizing through a redirect URL</a>.</li> <li><input type="checkbox"/> API Gateway only Disabled - do not collect the user credential</li> </ul> <p>Note: If you use either the Default HTML Form or Redirect identity extraction methods, the response from the redirect endpoint <b>must</b> maintain the order of the query parameters before the <b>state_nonce</b> query parameter, otherwise the authorization fails.</p> |
| Authentication      | <p>Authenticate application users with a user registry. Select an LDAP or Authentication URL user registry or create the SampleAuthURL User Registry. For a DataPower API Gateway, you have the option to disable authentication with a user registry.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Authorization       | <p>The following methods for extracting the user credential are available:</p> <ul style="list-style-type: none"> <li>Authenticated - Authorize authenticated users automatically.</li> <li>Default HTML Form - Use default HTML form to authorize.</li> <li>Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML authorization form for user security</a>.</li> <li><input type="checkbox"/> API Gateway only Disabled - Disable authorization.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

- Review the Summary for the native OAuth provider configuration.
- Click Back to make changes.
- Click Finish to save the basic configurations and to proceed to the Advanced Parameters for a native OAuth Provider.

## Results

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

- [Configuring basic settings for a native OAuth provider](#)  
You can update the identification details and basic configuration settings for a native OAuth provider.
- [Configuring scopes for a native OAuth provider](#)  
Access tokens contain authorization for specific scopes.
- [Configuring user security for a native OAuth provider](#)  
Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization.
- [Configuring tokens for a native OAuth provider](#)  
Set time to live for access tokens and refresh tokens, and a time period for maximum consent for all tokens.
- [Configuring token management and revocation for a native OAuth provider](#)  
Select whether to use a native gateway (DataPower) or third party endpoint for token revocation.
- [Configuring introspection for a native OAuth provider](#)  
Define an introspection path to allow the metadata for an access token to be examined.
- [Configuring metadata for a native OAuth provider](#)  
Use Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.
- [Configuring the OIDC parameters for a native OAuth provider](#)  
Open ID Connect (OIDC) provides an additional authentication protocol based on OAuth 2.0. OIDC provides user information encoded in a JSON Web Token, or JWT.
- [Editing a native OAuth provider by using the API Editor](#)  
You can edit the source and assembly policies for the Native OAuth Provider using the API editor.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring basic settings for a native OAuth provider

You can update the identification details and basic configuration settings for a native OAuth provider.


### About this task

One of the following roles is required to configure the basic settings for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator

- Custom role with the `Settings:Manage` permissions

You can select the basis settings pages for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the basic settings for an existing native OAuth provider. If you want to update the basic settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, > OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

1. To modify the identification details, click Info in the sidebar menu, then update the following fields as required:

| Field                  | Description                                                                                                                                                                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                  | Enter a title for the native OAuth provider.                                                                                                                                                                                                                                                                                                        |
| Name                   | This field is auto-populated by the system.                                                                                                                                                                                                                                                                                                         |
| Description (optional) | Enter a brief description.                                                                                                                                                                                                                                                                                                                          |
| Base path (optional)   | The base path is the URL segment of the API that is shared by all operations in the API. It does not include the host name or any additional segments for paths or operations. The base path must be unique for a given catalog. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty. |

2. To modify the basic configuration settings, click Configuration in the sidebar menu, then update the following fields as required:

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authorize Path         | <code>/oauth2/authorize/</code> is the standard OAuth endpoint to login to account                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Token Path             | <code>/oauth2/token/</code> is the standard OAuth endpoint to exchange code for access token.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Supported grant types  | <ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type "Client Credentials." Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type "Authorization Code."</li> <li>• Resource owner - Password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> <li>• <span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Resource owner - JWT - A verified signed JSON Web Token is exchanged directly for an access token.</li> </ul> <p>Note: If you plan to configure OpenID Connect (OIDC) for a native OAuth provider, you must include at least one of the following grant types: Implicit, Access code.</p> |
| Supported client types | <ul style="list-style-type: none"> <li>• Confidential - Client can maintain secure credentials on a secure server</li> <li>• Public - Client credentials are not secure. (Public is not available for DataPower® API Gateway at this time.)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

DataPower Gateway (v5 compatible) only Note: If the gateway type is DataPower Gateway (v5 compatible) and, when the native OAuth provider was created, only the Application grant type was selected, you cannot add further grant types until you configure the user security settings. In particular, you must specify the user registry for authenticating application users. To configure the user security settings, complete the following steps:

- a. Click User Security in the sidebar menu, then click Edit.
- b. Update the user security settings as required; for more details, see [Configuring user security for a native OAuth provider](#).
- c. Click Save when done.

3. Click Save when done.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring scopes for a native OAuth provider

Access tokens contain authorization for specific scopes.

### About this task

Client applications can request only the scopes or a subset of the scopes that you define here. The scopes are included in the access tokens that are generated from the provider. When an OAuth protected API is invoked, the gateway checks the scopes carried in the access tokens against the list of allowed scopes in the security definition for the API to determine whether to grant access.

In addition, you can enforce advanced scope checks. The advanced scope check URLs are invoked after application authentication or after user authentication based on which URLs are configured. The final scope permission that is granted by the access token is the result of all scope checks.


Per IETF RFC 6749, the value of the scope parameter is a list of space-delimited, case-sensitive strings. For more information, see [The OAuth 2.0 Authorization Framework](#).

One of the following roles is required to configure scopes for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

You can select the scope settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the scope settings for an existing native OAuth provider. If you want to update the scope settings for an existing native OAuth provider, complete the

following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

Perform the following steps to configure the scopes for the access token:

1. Click **Scopes** in the sidebar menu. The currently configured scopes are listed. Review and update the scopes as required.

| Field             | Description     |
|-------------------|-----------------|
| sample_scope_1    | Scope for token |
| additional scopes | Scope for token |

In the **Default scopes** section, select the default scopes to be used if the API request doesn't contain any scopes.

If the user authorization method is set to **Default HTML Form** in the **User Security** settings, all scopes specified here are added automatically to the authorization consent form.

2. **Advanced scope check before token generation.** This setting specifies the scope check endpoint where additional scope verification is performed in addition to the basic scopes. The advanced scope check URLs are invoked after application authentication or after owner authentication based on which URLs are configured. The scopes are included in the token and will overwrite any previous scopes.

| Field                   | Description                                                                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application scope check | Allow extra verification by running a scope check from an endpoint. Enter the endpoint and an optional TLS Profile to use for an application scope check. |
| Owner scope check       | Further refine the scope with an additional check. Enter the endpoint and an optional TLS Profile to use for an owner scope check.                        |

For more information about scope, see [Scope](#)

3. **Advanced scope check after token generation.** This setting specifies an additional scope check at the API consumer level to verify compliance with the scope requirements of the API.

| Field                 | Description                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Enabled               | Select the check box to enable the advanced scope check after token validation. Enter an optional default validator endpoint. |
| Use endpoint from API | Select the check box to use the endpoint from the API, or clear the check box to override the endpoint from the API.          |

For more information about scope, see [Scope](#)

4. Click **Save when done**.

## Results

Depending upon the visibility setting, the OAuth Provider with the specified scopes can be used to secure the APIs in catalog.

## Related information

- [Scope](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring user security for a native OAuth provider

Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization.


### About this task

User security authenticates the user. It is required for the **Implicit Access Code (Authorization code)**, and **Resource owner password grant** types. It is not used for an **Application grant** type.

One of the following roles is required to configure user security for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions




You can select the user security settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the user security settings for an existing native OAuth provider. If you want to update the user security settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

Perform the following steps to configure the user security settings for the OAuth Provider:

1. Click User Security in the sidebar menu.
2. Specify the following parameters for User Security. Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization. User Security is not required for the Client Credentials (Application) grant type. Click Next when done.

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identity Extraction | <p>Determines how the user credential is extracted:</p> <ul style="list-style-type: none"> <li>• Basic Authentication - HTTP basic authentication (requires no additional configuration)</li> <li>• Default HTML Form - Use default login form for user name and password</li> <li>•  Context variable - Specify which variable contains the user name and password. API Connect OAuth context variables as listed here <a href="#">API Connect context variables</a></li> <li>• Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML login form for user security</a>.</li> <li>• Redirect - Enter an endpoint to redirect to a third-party identity provider. For more information, see <a href="#">Authenticating and authorizing through a redirect URL</a>.</li> <li>•  Disabled - do not collect the user credential</li> </ul> <p>Note: If you use either the Default HTML Form or Redirect identity extraction methods, the response from the redirect endpoint <b>must</b> maintain the order of the query parameters before the <code>state_nonce</code> query parameter, otherwise the authorization fails.</p> |
| Authentication      | <p>Authenticate application users with a user registry. Select an LDAP or Authentication URL user registry or create the SampleAuthURL User Registry.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Authorization       | <p>Various methods may be used to authorize application users. For a DataPower® API Gateway, the following methods for extracting the user credential are available:</p> <ul style="list-style-type: none"> <li>• Authenticated - Authorize authenticated users automatically.</li> <li>• Default HTML Form - Use default HTML form to authorize. If you select the Default HTML Form method, all scopes that are specified in the Scopes settings are added automatically to the authorization consent form.</li> <li>• Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form.</li> <li>•  Disabled - Disable authorization.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

3. Click Save when done.

## Results

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

## Related information

- [Creating a custom HTML login form for user security](#)
- [Creating a custom HTML authorization form for user security](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring tokens for a native OAuth provider

Set time to live for access tokens and refresh tokens, and a time period for maximum consent for all tokens.


## About this task

Access tokens are granted to the client application to allow the application to access resources on behalf of the application user. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or more narrow scope. You can also specify how long the consent given by the combination of any number of access and refresh token remains valid.

One of the following roles is required to configure tokens for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

You can select the token settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the token settings for an existing native OAuth provider. If you want to update the token settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

Perform the following steps to configure tokens for the native OAuth provider:

1. Click Tokens in the sidebar menu.

2. Define the settings to configure tokens.

| Field                                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Access tokens time to live                                               | Enter the expiration time period in seconds for access tokens.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <input type="checkbox"/> API Gateway only<br>One time use access token   | Click the check box to enable one time use for the access token. Access tokens are multiple use by default which allows them to be used for multiple requests. When one time use is enabled, the access token will be consumed after one use. The OAuth flow will need to be repeated to obtain another access token.<br>Note: If you select this option, you must also enable token management; see one of the following topics, depending on the user interface you are using: <ul style="list-style-type: none"><li><a href="#">Configuring token management and revocation for a native OAuth provider</a> (Cloud Manager)</li><li><a href="#">Configuring token management and revocation for a native OAuth provider</a> (API Manager)</li></ul>                                                                                                                                                                                                      |
| Refresh tokens                                                           | Click the check box to enable Refresh tokens. Set the Count to limit the number of times a refresh token can be issued. Set the Refresh Token Time to Live value to determine the time to live, or expiration time period, for each refresh token in seconds.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| One time use refresh token                                               | Clear the check box to disable one time use for the refresh tokens. Refresh tokens are one time use by default which allows them to be used one time only to generate an access token and a new refresh token. When refresh token one time use is disabled then the refresh token count is limited to one and the refresh token can be used multiple times to generate new access tokens, however, another refresh token will not be generated unless the initial OAuth flow (Authorization Code or Password) is repeated.<br>Note: If you select this option, you must also enable token management; see one of the following topics, depending on the user interface you are using: <ul style="list-style-type: none"><li><a href="#">Configuring token management and revocation for a native OAuth provider</a> (Cloud Manager)</li><li><a href="#">Configuring token management and revocation for a native OAuth provider</a> (API Manager)</li></ul> |
| Maximum consent                                                          | Click the check box to enable Maximum consent and enter the Maximum Consent Time to Live value in seconds. This is the time to live, or expiration time period, for all tokens, both access and refresh.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <input type="checkbox"/> API Gateway only<br>Token secret                | Click the check box to select the Shared Secret which was configured for the gateway. If no Shared Secret was entered in the Gateway Configuration, then enter an key name and key value to use as the token secret.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <input type="checkbox"/> API Gateway only<br>Proof Key for Code Exchange | Proof Key for Code Exchange (PKCE) is a method to protect OAuth 2.0 public clients from an authorization code interception attack when they use Authorization Code grant requests. You can enable this extension when deploying with the DataPower® API Gateway.<br>For more information, see <a href="#">RFC 7636</a> .<br>Select the options for your OAuth Providers: <ul style="list-style-type: none"><li>Enable proof key for code exchange<br/>If selected, enforces PKCE when submitted in Authorization Code grant requests.</li><li>Always required<br/>If selected, requires PKCE in all Authorization Code grant requests.</li><li>Allow plain<br/>Select this check box to allow the plain challenge method in Authorization Code grant requests.</li></ul>                                                                                                                                                                                    |

3. Click Save when done.

## Results

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

## Related information

- [Refresh tokens](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring token management and revocation for a native OAuth provider

Select whether to use a native gateway (DataPower) or third party endpoint for token revocation.

## About this task

Token management enables you to prevent replay attacks by configuring token revocation. API Connect supports token revocation using a native gateway (DataPower) or a third party endpoint. For a native gateway, quota enforcement is used to manage tokens. For a third party endpoint, a URL to an external service is used to manage tokens.


For more information, see the IETF RFC 7009 [OAuth 2.0 Token Revocation](#).

One of the following roles is required to configure token management and revocation for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

You can select the token management settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the token management settings for an existing native OAuth provider. If you want to update the token management settings for an

existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

---

Perform the following steps to configure revocation settings for tokens:

1. Select Token Management in the sidebar menu.
2. Enable Token Management by selecting the check box.
3. From the Type dropdown menu, select either Native or Third party. Native points to DataPower as the token storage location; Third party points to a revocation URL for token storage. For DataPower® API Gateway, only Native is supported.
4. For Native, select one or both of the Resource owner revocation path and Client revocation path.
  - Resource owner revocation path - Uses the standard OAuth revocation path to allow the resource owner (end user) to revoke the application permission.
  - Client revocation path - Uses the standard OAuth revocation path to allow the client (application) to revoke a single token when the application closes.For more information about managing tokens with the Native DataPower Gateway, see [Token management with the native DataPower Gateway](#).
5. For Third party, specify the Endpoint and TLS Client Profile.
  - Endpoint - Enter the URL to an external web server that contains information about access or refresh tokens. API Connect calls the URL to determine if the associated token can be trusted. The token server then checks a token *blacklist* (a data store of inactive tokens) to ensure that the token is still valid. If the token is still valid, API Connect continues the processing. For more information see [Token revocation](#).
  - TLS Client Profile - Select a TLS profile to verify the external endpoint.
6. Click Save when done.

## Results

---

Depending on the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring introspection for a native OAuth provider

Define an introspection path to allow the metadata for an access token to be examined.

### About this task


---

Token introspection allows an authorized holder of an access token to examine the contents of tokens using an introspection path. The access token to introspect must be obtained through the native OAuth provider. Introspection provides context for the token by allowing an authorized protected resource to query the authorization server to determine the set of metadata for a given token. The metadata includes whether or not the token is currently active, the scopes assigned to the token, and the authorization context in which the token was granted (including who authorized the token and which client it was issued to). API Connect token introspection conforms to IETF RFC 7662. See [OAuth 2.0 Token Introspection](#).

One of the following roles is required to enable introspection for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

You can select the introspection settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the introspection settings for an existing native OAuth provider. If you want to update the introspection settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

---

Perform the following steps to enable introspection:

1. Click Introspection in the sidebar menu.
2. Select the check box to enable Introspection. The OAuth standard path for introspection, `/oauth2/introspect` is automatically entered. This path will be used when another entity inspects the token contents.
3. Click Save when done.

## Results

---

Tokens will be queried using the `/oauth2/introspect` path. Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring metadata for a native OAuth provider

Use Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.

### About this task

---

Configure an Authentication URL or an External URL from which custom metadata is collected for inclusion in the token. The metadata is either stored inside the access token or it is sent along with the access token to the client application. For more information about how the metadata is collected, see [OAuth external URL and authentication URL](#).


Following are examples of metadata that can be included with the access token:

- Metadata about the authenticated resource owner
- Grant type that was used to obtain the token
- A confirmation code to be provided to the client application

One of the following roles is required to configure metadata collection for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions



You can select the metadata settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the metadata settings for an existing native OAuth provider. If you want to update the metadata settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

### Procedure

---

Perform the following steps to configure metadata collection:

1. Click Metadata in the sidebar menu.
2. Select Collect metadata to enable metadata collection.
3. The Authentication URL user registry is selected by default and is required. For more information about the Authentication URL, see [Authentication URL user registry](#).
4. Select the External URL to collect metadata from an external URL. Enter the endpoint and an option TLS Client Profile.
5.  If required, override the default Header name token value. The value of this header, if returned in the response from the OAuth endpoint, is placed in the response payload and indicated as `metadata`.
6.  If required, override the default Header name payload value. The value of this header, if returned in the response from the OAuth endpoint, is placed within the access token and indicated as `miscinfo`.
7. Save the OAuth Provider.
8. Click Save when done.

### Results

---

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the OIDC parameters for a native OAuth provider

Open ID Connect (OIDC) provides an additional authentication protocol based on OAuth 2.0. OIDC provides user information encoded in a JSON Web Token, or JWT.

### About this task

---

When you enable OpenID connect, a template is provided for generating ID tokens along with access tokens and the required assembly policies are automatically created. You can customize the policies to suit your needs in the API Editor. The sample key provided is for test purposes only and is used to sign the JWT token.

One of the following roles is required to configure an OIDC template for a native OAuth Provider:


- Administrator
- Owner
- Topology Administrator



- Custom role with the `Settings:Manage` permissions

Note: You can configure OIDC parameters only if the selected grant types for the native OAuth provider include at least one of the Implicit or Access code grant types; see [Configuring basic settings for a native OAuth provider](#).



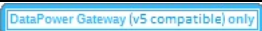


You can select the OIDC settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the OIDC settings for an existing native OAuth provider. If you want to update the OIDC settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, > OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

Perform the following steps to configure an OIDC template:

1. Click OpenID Connect in the sidebar menu.
2. Select the initial check box to configure an OIDC Template. Enter the following parameters:

| Field                                                                                                                      | Description                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Support hybrid response types (optional) | Select the response types for the OpenID Connect hybrid flow to be supported by this OAuth provider.                                                                                             |
|  Auto Generate OIDC API Assembly          | Select this option to generate the full OIDC assembly. Leave this option unselected to simply enable OIDC support in the OAuth provider and allow the developer to implement their own assembly. |
|  ID token issuer                          | Descriptive text to indicate the source of the key.                                                                                                                                              |
|  ID token signing key                     | Specify the JSON Web Key (JWK) to be used to sign the ID token.                                                                                                                                  |
|  ID token signing algorithm               | Select the algorithm used to sign the token.                                                                                                                                                     |

3. Click Save when done. You can edit the policies by using the API Editor.

## Results

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Editing a native OAuth provider by using the API Editor

You can edit the source and assembly policies for the Native OAuth Provider using the API editor.

### About this task

If you have configured an OIDC template, you can customize it in API Editor. In the API Editor, the Source tab allows you to edit the code for the configuration using a text editor. The API Assemble tab provides a graphical drag-and-drop editor (identical to the one in API Manager) that allows you to add additional elements to the assembly for the OAuth Provider.


Note: Take care when modifying the code directly on the Source tab of the API Editor because validation is limited. For example:

- If you change the name of auto generated assembly actions in the source code, the assembly will be prevented from updating dynamically when the OAuth provider settings are modified.
- You must ensure that the OAuth provider name matches the value specified in the `oauth-provider-settings-ref` field in each OAuth assembly action.

One of the following roles is required to configure tokens for a native OAuth Provider:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

You can modify the native OAuth provider configuration by selecting the API Editor page immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the configuration for an existing native OAuth provider. If you want to update the configuration for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, > OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

Perform the following steps to edit the OAuth configuration:

1. Click API Editor in the sidebar menu.
2. In the Source tab, view and edit the policies to customize the behavior for the OAuth provider.
3. In the API Assemble tab, use the drag and drop editor to add additional policies to the OIDC behavior.  
 Note: If you add a policy that references a TLS profile, an **invoke** policy for example, then when you publish an API that uses this OAuth provider, you must ensure that the TLS profile is enabled for the Catalog to which you publish the API. For details on how to enable a TLS profile in a Catalog, see [Creating and configuring Catalogs](#).
4. Click Save when done.

## Results

Depending upon the visibility setting, the OAuth Provider can be used to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring a third-party OAuth provider

Enter the secure endpoints to provide OAuth authentication from a third party.


### About this task

One of the following roles is required to configure OAuth Providers:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions



### Procedure

Perform the following steps to configure a third party OAuth provider:

1. In the Cloud Manager, click  Resources.
2. Select OAuth Providers > Add > Third party OAuth Provider.
  - a. Complete the following parameters for the first screen and click Next.

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Title                 | Enter a descriptive title for the gateway service. This title will be displayed on the screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Name                  | This field is auto-populated by the system and used as the internal field name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Supported grant types | <ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type Client Credentials. Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type Authorization Code.</li> <li>• Resource owner password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> </ul> |
| Gateway type          | Select the gateway type, either DataPower® Gateway (v5 compatible) or DataPower API Gateway.<br>For information about types of gateways, see <a href="#">API Connect gateway types</a> .<br><br>OAuth Providers apply to one gateway type.                                                                                                                                                                                                                                                                                                                                                                                            |

- b. Specify configuration settings for the endpoints.

| Field                                                                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authorization URL                                                                                                                       | An authorization URL where the resource owner grants authorization to the client application to access a protected resource. Example:<br><br><b><code>https://example.com/oauth2/authorize</code></b>                                                                                                                                                                 |
| Token URL                                                                                                                               | A token request URL where the client application exchanges an authorization grant for an access token. Example:<br><br><b><code>https://example.com/oauth2/token</code></b>                                                                                                                                                                                           |
| Introspect URL                                                                                                                          | The introspection URL is where the API gateway validates the access tokens that are issued by the third party provider. Example:<br><br><b><code>https://example.com/oauth2/introspect</code></b><br><br>For more information on integrating third party OAuth providers for introspection, see <a href="#">OAuth introspection for third-party OAuth providers</a> . |
| TLS Profile (optional)                                                                                                                  | Select an optional TLS profile for communicating with the third party provider.                                                                                                                                                                                                                                                                                       |
| Security                                                                                                                                | Default is Basic Authentication.                                                                                                                                                                                                                                                                                                                                      |
|  Basic authentication request header name            | The <code>x-introspect-basic-authorization-header</code> is available to provide a user-configured HTTP Basic authorization header.                                                                                                                                                                                                                                   |
|  Basic authentication request header name (optional) | The name of the header that will provide a user-configured HTTP Basic authorization. The default value is <code>x-introspect-basic-authorization-header</code> .                                                                                                                                                                                                      |

| Field                                                           | Description                                                                                                             |
|-----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| API Gateway only<br>Basic authentication<br>username (optional) | The default user name for HTTP Basic authentication.                                                                    |
| API Gateway only<br>Basic authentication<br>password (optional) | The default password for HTTP Basic authentication.                                                                     |
| API Gateway only<br>Custom header pattern<br>(optional)         | A regular expression for request headers that are to be passed to the third-party provider; for example, x-Introspect-* |

c. Enter the scopes in the third screen. A scope becomes an option in the request and response for an access token. Click Add to add additional fields for scopes. Click Next when done.

| Field             | Description     |
|-------------------|-----------------|
| sample_scope_1    | Scope for token |
| sample_scope_2    | Scope for token |
| additional scopes | Scope for token |

d. Review the settings on the Summary panel.

3. Click Save and Edit to complete the configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting the visibility for OAuth providers

The visibility setting determines which provider organizations can use an OAuth provider to secure APIs.


### About this task

One of the following roles is required to configure OAuth Providers:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings:Manage` permissions

### Procedure

Follow these steps to set the visibility for the OAuth providers in your on-premises cloud:

1. In the Cloud Manager, click  Resources.
2. Select OAuth Providers.
3. Choose Edit visibility from the actions menu next to the name of the OAuth provider that requires a visibility setting.
4. Select the visibility setting for the OAuth provider. The options are:
  - Private - the OAuth provider is not visible and cannot be used by any provider organization. It is accessible only to the admin organization in Cloud Manager.
  - Public - the OAuth provider is visible and can be used by all provider organizations
  - Custom - the OAuth provider is visible only to the provider organizations that you designate
5. For Custom visibility, select the provider organizations that you want to have access to the OAuth provider.
6. Click Save to complete the operation.

### Results

For Private, the OAuth provider is not visible and cannot be used by any provider organizations. For Public, the OAuth provider is visible and can be used by all provider organizations. For Custom, the OAuth provider is visible only to the provider organizations designated by you.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OAuth concepts for API Connect

OAuth is a token-based authorization protocol that allows third-party websites or applications to access user data without requiring the user to share personal information.

Note: In a multi-node cluster, OAuth operations will fail if quorum is lost. Quorum requires that the number of active nodes is greater than 50% of the total number of nodes in the cluster.

- [OAuth user scenario](#)  
Potential users of OAuth with IBM® API Connect have a number of methods to secure their API. The following scenario provides an overview of the available

options.

- [OAuth introspection for third-party OAuth providers](#)  
OAuth token validation can be offloaded to the third-party Open Authorization (OAuth)/Open ID Connect (OIDC) provider by using the Introspection URL. Clients can use a third-party OAuth or OIDC provider to obtain a token that is protected by IBM API Connect. IBM API Connect can use this feature along with the mentioned provider to protect access to the API.
- [OAuth external URL and authentication URL](#)  
You can use the Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.
- [Scope](#)  
Per IETF RFC 6749, the value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings.
- [Tokens](#)  
Tokens can be managed using refresh tokens and revocation URLs. You can extend the life of tokens using refresh tokens. You can end the life of a token by specifying a revocation URL.
- [Authentication URL user registry](#)  
You can use an Authentication URL user registry to specify a REST authentication service that manages user authentication, and optionally provides additional metadata to be embedded in the token.
- [Custom forms for user security](#)  
You can create custom forms for the authorization and identity extraction phase of OAuth.
- [Securing an API with a JSON Web Token](#)  
There are two methods to secure your API with a JSON Web Token. You can use the `jwt-generate` command, or you can use a token that has been generated external to IBM API Connect.
- [Troubleshooting OAuth](#)  
You can find the answers to some common questions in the Developer Portal, or in support communities and forums in DeveloperWorks, on GitHub, or on Youtube.

---

## Related information

- [The OAuth 2.0 Framework](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## OAuth user scenario

Potential users of OAuth with IBM® API Connect have a number of methods to secure their API. The following scenario provides an overview of the available options.

---

### Scenario overview

In this scenario, Alice is a user of an application. Alice can grant permission for an application to access specific information about Alice in a third-party system that is using OAuth. Depending on the type of OAuth that is supported by the target service, Alice does not enter her user name and password into the application. Instead, the application receives an access token that represents her credentials (user name and password). The application can now access the information about Alice in the target system.

For example, Alice maintains a list of books that she reads on a service that is provided by mybooks.com. Following the purchase of a smartphone, Alice installs an application on her new phone to display the book details. The phone application wants to call an API provided by mybooks.com, which can access the information. The mybooks.com API is secured by using the OAuth protocol.

To access the book details, the application must complete a two-step process:

1. The application must first obtain permission from Alice.
2. The application then uses that permission to call the target service and obtain the list of books.

In the first step, the application typically directs Alice to the provider of the target service, mybooks.com. Alice provides her user name and password, and gives permission for the application to access her information. It is important that Alice trusts that she is providing her credentials to the provider of the target service and not to an untrusted proxy application. For example, by checking that the security certificate of the website where Alice enters her credentials matches what Alice expects from the provider of the target service.

The result of this step is the access token that the application can use to call the API. The application then generates the appropriately formatted OAuth request. For example, the Authorization header, or HTTP query parameters, which includes the access token, consumer key, and signature method that are required by OAuth. This OAuth request is used to invoke the API proxy operation.

---

### Scenario within API Connect

No changes to the definition of your API operation are required to support this scenario.

1. Alice grants permission for the application to access her information *before* the invocation of the API.
2. When the application provides the Authorization header, or query parameters, containing the OAuth details about the call to the operation endpoint, the header is automatically passed through to the target service without any additional configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

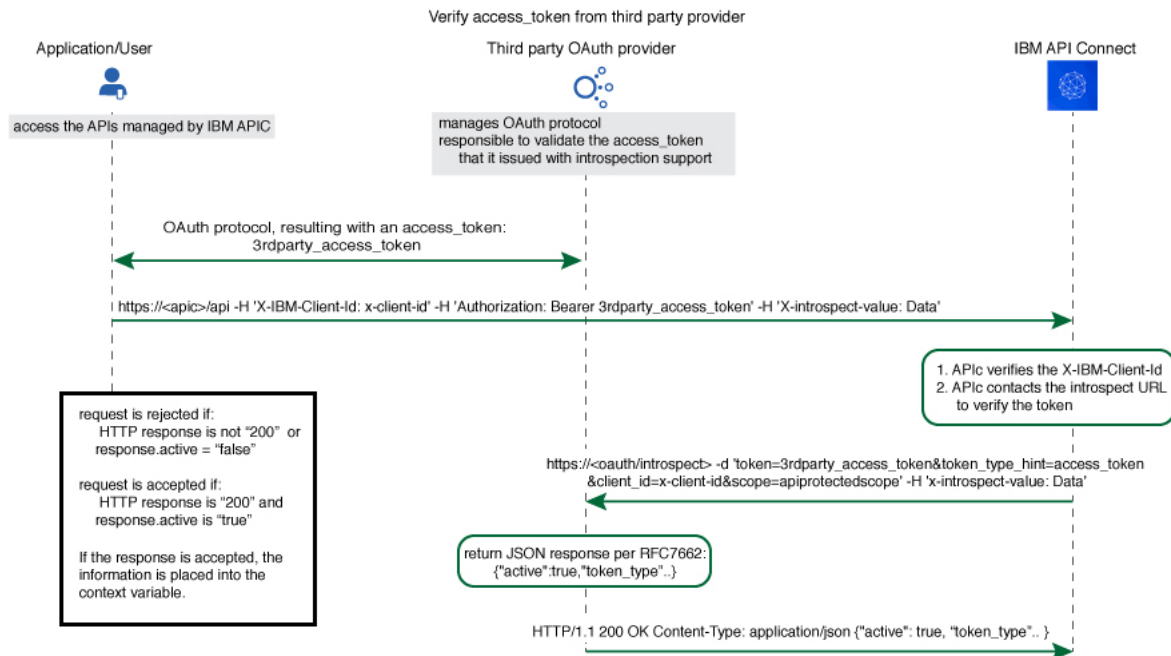
## OAuth introspection for third-party OAuth providers

OAuth token validation can be offloaded to the third-party Open Authorization (OAuth)/Open ID Connect (OIDC) provider by using the Introspection URL. Clients can use a third-party OAuth or OIDC provider to obtain a token that is protected by IBM API Connect. IBM® API Connect can use this feature along with the mentioned provider to protect access to the API.

You can use API Connect to protect an API that is secured by using the third-party OAuth access token in accordance with Introspection specification as defined in [RFC 7662](#). In addition to the specification, the `x-Introspect-` header is provided to pass other content to the third party as you require.

Authentication information can be carried in the request by configuring a basic authentication request header.

The following sequence diagram depicts the overall flow of request and response. The purpose of this diagram is only to provide a general visualization of the nature of the flow; for the precise details, refer to the explanatory information after the diagram.



The Introspect URL is configured in the Third Party OAuth provider configuration. See [Configuring a third-party OAuth provider](#).

When an API is protected by a third-party OAuth provider, API Connect will extract the bearer token and issue an HTTP POST request to the endpoint specified in the Introspect URL field.

The GET request is protected by API Connect with this feature.

You can use a header prefix to pass information to the third-party provider. The header prefix can include a regular expression and specifies the name pattern of the headers to use for sending additional information, such as `x-introspect-*`.

```
GET /petstore/pet/123 HTTP/1.1
Host: apiconnect.com
x-Introspect-type: dog
x-Introspect-name: simon
x-custom-apic: petstore123
Authorization: Bearer tGzv3JOkF0XG5Qx2T1KWIA
x-IBM-Client-Id: xxx-xxx
```

However, if you want to use a different header prefix, specify the required value in the Custom header pattern field in the third party OAuth provider configuration. For third-party OAuth provider configuration details, see [Configuring a third-party OAuth provider](#). The Custom header pattern feature is available only with the DataPower® API Gateway, if you are using the DataPower Gateway (v5 compatible) the header prefix must be `x-Introspect-`.

API Connect will issue this POST request to the introspection endpoint specified in `x-tokenIntrospect`, as illustrated in the code sample:

```
POST /oauth/introspectURL HTTP/1.1
Host: apiconnect.ibm.com
Content-Type: application/x-www-form-urlencoded
x-Introspect-type: dog
x-Introspect-name: simon
x-custom-apic: petstore123

token_type_hint=access_token&token=tGzv3JOkF0XG5Qx2T1KWIA
```

The third party OAuth/OIDC provider will respond with HTTP 200 indicating the request was successful and that payload contains the token information. API Connect honors the active claim as defined in the RFC specification.

If the value of the active claim is `true`, the token is treated as valid.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Cache-Control: no-store
```

```
Pragma: no-cache
```

```
{ "active": true,  
  "token_type": "bearer",  
  "client_id": "xxx-xxx",  
  "username": "John Smith",  
  ...  
}
```

If the value of the active claim is `false`, the token is treated as invalid.

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=UTF-8  
Cache-Control: no-store  
Pragma: no-cache
```

```
{  
  "active": false  
}
```

A response code other than `HTTP 200` indicates failure to process the request.

When the OAuth token is valid and active, context variables are populated with information from the introspect JSON response. For more information, see [API Connect context variables](#).

When contacting the introspection endpoint, API Connect uses `client_id/appId` and `client_secret/appSecret` to construct the HTTP Basic authorization header.

By default, if `x-introspect-basic-authorization-header` exists in the request, the value is used for the HTTP Basic authentication header when the introspection endpoint is contacted. API Connect verifies that the HTTP Basic authentication header value is Base64 encoded before it is sent, and encodes it if necessary as shown in the following example. If the header is already encoded, it is sent without modification.

```
GET /petstore/pet/123 HTTP/1.1  
Host: apiconnect.com  
x-introspect-basic-authorization-header: user:password
```

API Connect issues the following:

```
POST ..  
Authorization: Basic M3JkLXBhcncR5LWNsaWVudF9pZDozcmQtcGFydH1fY2xpZW50X3NlY3JldA==token_type_hint=access_token&token= ..
```

If you are using the DataPower API Gateway, you can specify your own value for the HTTP Basic authentication header by providing the required value in the Basic authentication request header name field in the third party OAuth provider configuration. The following rules determine what authentication data is passed to the introspection endpoint:

- If there is a basic authentication header in the request, the specified credentials are used. The value must be either a string in the format `user:password`, or the Base64 encoded equivalent; API Connect will Base64 encode the value if necessary before sending the request to the introspection endpoint.
- If there is no basic authentication header in the request but there are values specified in the Basic authentication username and Basic authentication password fields in the third party OAuth provider configuration, those values are used.
- If there is no basic authentication header in the request, and user name and password details are not supplied in the third party OAuth provider configuration, but `client_id` and `client_secret` values are supplied in the body of the request, these are used.
- Otherwise, an error is returned.

For third-party OAuth provider configuration details, see [Configuring a third-party OAuth provider](#).

If either, or both, of `scope` and `scope validate url` are configured, and if the response is an active token with a scope claim from the third-party OAuth Provider introspection endpoint, API Connect would further enforce the scope validation in the following order:

1. If `scope` is configured for the OAuth API protection, verify the third-party scope against the scope that is configured.
2. If `scope validation url` is configured, verify the third-party scope against the scope validation url.

For more information, see [Scope](#).

**DataPower Gateway (v5 compatible) only** By default the API Connect client ID and scope are sent to the third party OAuth provider. You can suppress this behavior in either of the following ways:

- Supply a `suppress-parameters` header as follows:

```
suppress-parameters: client_id
```

```
suppress-parameters: scope
```

or

```
suppress-parameters: client_id scope
```

depending on which parameters you want to suppress.

- Define an API property called `suppress-parameters` in the API definition itself, with one of the following string values:

```
client_id
```

```
scope
```

or

```
client_id scope
```

depending on which parameters you want to suppress. For information on how to define API properties, see [Setting API properties](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## OAuth external URL and authentication URL

You can use the Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.

---

### Including custom metadata in a token

In many scenarios, custom metadata needs to be included during the access token generation process. The metadata is either stored inside the access token or it is sent along with the access token to the client application. The client application can then send that access token, or the metadata in the payload, in a subsequent request to IBM® API Connect where the metadata is retrieved, validated, or sent to the downstream systems as required.

Examples include, but are not limited to:

- When resource owners are authenticated, metadata about the authenticated resource owner needs to be stored within the access token.
- The grant type that was used to obtain the token is another example of metadata stored within the access token.
- A confirmation code that needs to be sent to the client application along with access token is stored as metadata within the access token.

---

### Configuring External URL or Authentication URL in API Connect to obtain metadata

Metadata can be set by using either or both of the following URLs:

- External URL - When you call the External URL, an HTTP GET request is sent and API Connect expects an HTTP 200 OK along with an optional set of the specified response headers.
- Authentication URL - When you call the Authentication URL, the API Connect gateway sends a GET request with HTTP headers and then processes any HTTP response from the URL. For authentication, a REST authentication service is expected at the Authentication URL.  
See: [Authentication URL](#).

The External URL endpoint is entered in the Metadata section when you configure an OAuth Provider in the Cloud Manager or API Manager UI.

Use the following HTTP headers in the response, depending on the type of gateway you are using:

- DataPower® API Gateway:

`X-API-OAUTH-METADATA-FOR-PAYLOAD`

`X-API-OAUTH-METADATA-FOR-ACCESSTOKEN`

Note: These are the default header names for the DataPower API Gateway but you can override them; see [Configuring metadata for a native OAuth provider](#).

- DataPower Gateway (v5 compatible):

`API-OAUTH-METADATA-FOR-PAYLOAD`

`API-OAUTH-METADATA-FOR-ACCESSTOKEN`

The response header value from `X-API-OAUTH-METADATA-FOR-PAYLOAD` or `API-OAUTH-METADATA-FOR-PAYLOAD` is placed in the response payload and indicated as `metadata`.

The response header value from `X-API-OAUTH-METADATA-FOR-ACCESSTOKEN` or `API-OAUTH-METADATA-FOR-ACCESSTOKEN` is placed within the access token and indicated as `miscinfo`.

The two metadata response headers are case insensitive and you must escape any special characters in the string value content.

An example response payload that contains metadata along with the access token:

```
{
  "token_type": "bearer",
  "access_token": "AAEkNzhjDHYyyYy...cL0Mv6ct137w7ZU",
  "metadata": "m:metadata-for-payload_content",
  "expires_in": 3600,
  "scope": "read",
  "refresh_token": "AAEnj5SynCMybF...oEZ6JjxYax_HdNg",
}
```

This example output from the token introspection endpoint shows the contents of the access token with `"miscinfo"` containing the metadata information.

```
{
  "active": true,
  "token_type": "bearer",
  "client_id": "78c2f10f-799a-4e1f-8e0a-098634997a35",
  "username": "Fred Smith",
  "sub": "fred",
  "exp": 1479850049,
  "expstr": "2016-11-22T21:27:29Z",
  "iat": 1479846449,
  "nbf": 1479846449,
  "nbfstr": "2016-11-22T20:27:29Z",
  "scope": "read",
  "miscinfo": "m:metadata-for-access token_content",
  "client_name": "MobileApp"
}
```

## Input to the External URL

The following request headers are sent to the External URL.

Note: Any existing metadata values that were previously sent from the Authentication URL are also sent in the two request headers **x-existing-metadata-for-payload** and **x-existing-metadata-for-access-token**. The Metadata URL can make use of this information to create a new set of metadata values. The two request headers that are sent to the Metadata URL are displayed in bold text.

```
X-existing-metadata-for-payload    payload-from-auth-url
X-existing-metadata-for-access-token    token-from-auth-url
X-URI-in    /org/env/miscinfo/oauth2/token (the URL that was sent to APIConnect for this particular token request)
X-METHOD-in    POST
X-POST-Body-in    client_id=client_id&grant_type=password&scope=read&username=name&password=password
X-X-Client-IP    IP_address
X-X-Global-Transaction-ID    ID_number
...
```

Note: If you are using the DataPower API Gateway rather than the DataPower Gateway (v5 compatible), the **X-POST-Body-in** header is not supported.

## Retrieving Metadata in API Connect

As described in the previous example scenarios, the metadata can be retrieved from the access token in the Application API and sent to the downstream systems. Retrieval can be done in the API assembly, which is secured to accept tokens in the security definitions.

In the resource API that accepts an access token, the **miscinfo** field can be accessed in the Assembly with the **oauth.miscinfo** context variable, as in the example.

```
apim.setvariable('message.body', apim.getvariable('oauth.miscinfo'));
```

You can also use token introspection to look at the contents of the access token.

## Refresh tokens and metadata

The Authentication URL (if configured for authentication) is invoked first, during authentication of the resource owner. The External URL is invoked as the last step, just before the generation of the access token. The only exception is when access tokens are generated from a refresh token. In cases where refresh tokens are used to generate new access tokens, the External URL is not invoked. The metadata from the refresh token is retained and then copied into the newly generated access token.

## Identifying the source of the metadata

The metadata is prefixed with keywords to indicate whether it originated from the External URL or the Authentication URL.

- Metadata from the External URL is prefixed with **m**:
- Metadata from the Authentication URL is prefixed with **a**:

Note: When revocation is enabled, some internal details are also stored in the **miscinfo** field, in square-brackets within the access token as shown in the following example:

```
"miscinfo": "[tlsprofile@https://api-revoke-url:443/server/revocation-url]m:metadata-for-accesstoken_content"
```

## Maximum size of the metadata

Metadata for the access token cannot exceed 512 bytes.

Metadata for the payload does not have a specific size restriction, except for when you use the Authorization code grant type. These restrictions are described in following sections.

## Characters are not allowed in metadata in certain scenarios

When you use the Authorization code grant type, or when a consent form is used for implicit grant type, there is a temporary state or code where the metadata from the authentication URL is stored. API Connect internally uses two prefixes - **!ma** and **!mp** to differentiate between payload and token metadata received from the Authentication URL and store them internally in the temporary state/code. Hence these specific character sequences - **!ma** and **!mp** should not be used as the metadata itself.

## Grant types and metadata

The OAuth grant types described in the following sections include **authorization code (access code)**, **implicit**, and **client credentials (application)**.

Authorization code grant type

- When metadata is included from an Authentication URL for an Authorization code grant type, as it is a three legged flow, both the content and the payload are stored within the **dp-state** and carried on to the authorization code and to the access token. Note that around 10 characters are used internally to differentiate between the metadata for payload and metadata for the access token when stored in the **dp-state**. In addition, if revocation is enabled, that will also be part of the token metadata. Hence the combined size of the token metadata, the payload metadata (including the 10 characters of internal data), and internal revocation details, cannot exceed 512 bytes in total.

If the overall size of the metadata exceeds 512 bytes, then the access token generation succeeds, but the metadata fields contain an error message of "metadata too large" as shown in the example.

```
"metadata": "m:error: metadata too large for AZ code grant type[Authorization Code-metadata-url-payload]"
"miscinfo": "[r:gateway]m:error: metadata too large for AZ code grant type[Authorization Code-metadata-url-token]"
```



This size restriction can be overcome when the metadata is sent from the Metadata URL and not from the Authentication URL, because the metadata is not stored in `dp-state` or in the authorization code.

Example of the `authorization code (access code)` grant type:

```
$ curl -k -d
"grant_type=authorization_code&code=$mycode&client_id=$myid&scope=scope_introspect&redirect_uri="
https://9.70.153.90/fei/sb/introspectpl/oauth2/token
{ "token_type": "bearer",
  "access_token": "AAEkOTh1ZDhhNjYtYtQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ2OZn5T1_TqYFeIfB7BFf6HFgibEoOjWXXEA84oFsWuE4NY-
RRZVdnGSaXAIYJz7s5vczfk5EV-BIb_6P_1YKm3ahrfr5kI3sPO0uADEoseIP5-09anUpEM5yhsayXvZbJ_6VDYz-hyXSJHTNqVj-
PHBialoRkBD5qca6k00fV2M", "metadata": "a:[Authorization Code-Test-auth-url-payload]", "expires_in": 3600,
  "scope": "scope_introspect", "refresh_token": "AAFAg1EVmBwicr_L0FTZ4q6HZ-
RcQygniXFC9zbSKO4wd3hcniC4KQX21X0fL2c8cnmzCZgws8xxLzNyfjQhUJNG15C1GbIe3dwhXJdiWA0Go-
dudhVtCbG26sJRRXyYrMeRwXWnJsy1BETPI8HQEN4a_D7fmxKcTVRZBvq86byg95qe1ZKyERi0Lhxdd_O4Nvss" }

$ curl -k -H "X-IBM-Client-Id: $myid" -H "X-IBM-Client-Secret: $mysecret" -d
"token_type_hint=access_token&token=$mytoken" https://9.70.153.90/fei/sb/introspectpl/oauth2/introspect
{ "active": true, "token_type": "bearer", "client_id": "98ed8a66-a45e-4a33-a47d-fa68fd344746", "username": "anyuser",
  "sub": "anyuser", "exp": 1484766368, "expstr": "2017-01-18T19:06:08Z", "iat": 1484762768, "nbf": 1484762768,
  "nbfstr": "2017-01-18T18:06:08Z", "scope": "scope_introspect", "miscinfo": "[r:gateway]a:[Authorization Code-Test-auth-
url-token]", "client_name": "oauth_app" }
```

Implicit grant type

- When implicit grant type is used, the access token and the metadata are returned in the `Location` header as a fragment, as you see in the example.

```
< Location:
https://localhost#access_token=AAEkOTh1ZDhhNjYtYtQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ2buS2KfWdq-
nYBJSi4nmPxQBtLae17kKBPRMzwP5BC386n1xpoOTE1G748ZVH6Mq_TJL3GeV3PtXTVIkWOLBji_7tljiQfnpVfrNkovvZkhUexYmFkcDsmLSdaWxZ6Pc
IMPAC4ojT8qVlsYV-
ChTk36yqOx_NiKimZaDikDk7WTg&expires_in=3600&scope=scope_introspect&token_type=bearer&metadata=a%3A[Implicit-Test-
auth-url-payload]

$ curl -k -H "X-IBM-Client-Id: $myid" -H "X-IBM-Client-Secret: $mysecret" -d
"token_type_hint=access_token&token=$mytoken" https://9.70.153.90/fei/sb/introspectpl/oauth2/introspect
{ "active": true, "token_type": "bearer", "client_id": "98ed8a66-a45e-4a33-a47d-fa68fd344746",
  "username": "anyuser", "sub": "anyuser", "exp": 1484768365, "expstr": "2017-01-18T19:39:25Z", "iat": 1484764765,
  "nbf": 1484764765, "nbfstr": "2017-01-18T18:39:25Z", "scope": "scope_introspect", "miscinfo": "[r:gateway]a:[Implicit-
Test-auth-url-token]", "client_name": "oauth_app" }
```

Client credentials grant type

Authentication URL will not be invoked when using client credentials grant type, as there is no resource owner. The metadata from Authentication URL is not available for this grant type. However, content returned from Metadata URL will be included as metadata.

## Behavior when retrieving metadata with both an External URL and an Authentication URL

If an External URL is configured and a connection to the external server is successful, the response headers overwrite any existing metadata obtained from the Authentication URL to become the final value. Therefore, you must carefully examine the incoming request headers and create appropriate response headers from the External URL.

If an External URL is configured, but the connection to the External URL fails, then a failure message of `"error on metadata url"` is written for metadata in both the payload and the access token.

If an External URL is configured and the connection is successful, but the remote server does not send any of the specified HTTP response headers, a blank value is written for metadata in both the payload and the access token.

Attention: An External URL overwrites existing values from Authentication URL. This includes blank values.

If no External URL is configured, the metadata that is obtained from the Authorization URL is retained as the final value.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Scope

Per IETF RFC 6749, the value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings.

In IBM® API Connect, scopes have no inherent meaning. Instead, scopes are defined in the OAuth Provider so that an application can request an access token that is valid for one or more of the scopes. In the secured API, scopes are listed as requirements for an access token to be considered valid. All scopes that are listed by the security definition for the API must be granted by the access token.

## OAuth provider

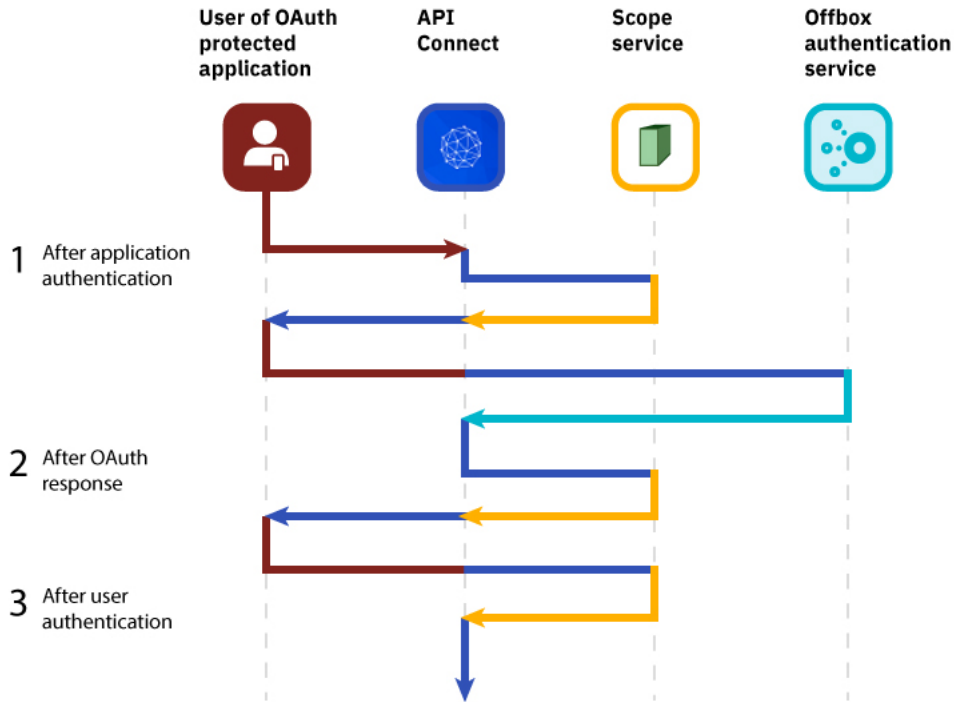
To provide more refined support for the OAuth scope handling, API Connect allows the [Authentication URL user registry](#) extension to modify the scope value.

When you define an [OAuth provider](#), the Advanced scope check extensions provide the flexibility to check and override allowed scopes. The optional extensions are Application scope check and Owner scope check.

The scope that is eventually received by the application is determined by the interactions that are described in the three following paragraphs. Scope processing follows the sequence of paragraphs 1, 2, and 3 in order, offering three opportunities to override the scope value. Figure 1 provides an overview of the process.

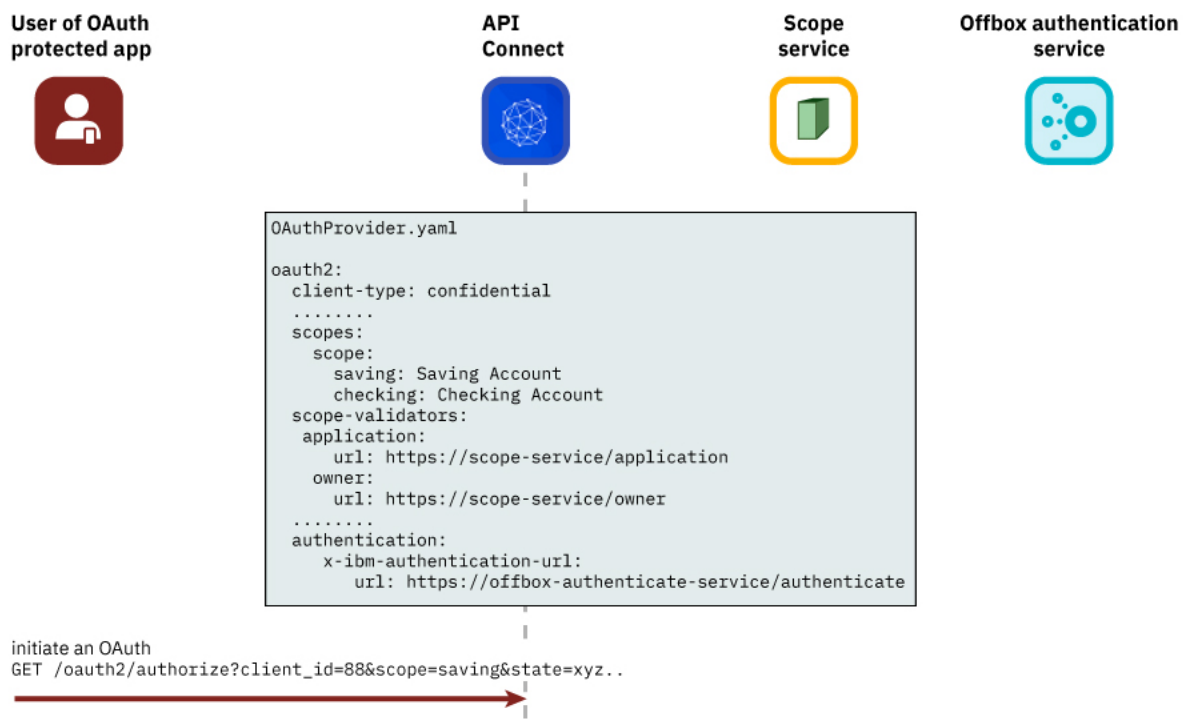
1. After the application successfully authenticates, and if OAuth Native provider `>Advanced scope check>` `>Application scope check` is configured, API Connect makes a call to allow extra verification and uses the contents of `x-selected-scope` to override the scope value that was initially requested by the application. When Application scope check is enabled, the HTTP response header `x-selected-scope` must be present, or the call fails.
2. If OAuth Native provider `>User Security>` `>Authentication` is configured to authenticate application users using an Authentication URL, then API Connect makes a call, as documented in [Authentication URL user registry](#). When the response code is `HTTP 200`, and the response header `x-selected-scope` is present, the value that is configured in `x-selected-scope` is used as the new scope value, overriding both what the application already requested and what was provided in the Application scope check described in paragraph 1. In the response header, `x-selected-scope` is an optional element.
3. After the user successfully authenticates, and if OAuth Native provider `>Advanced scope check>` `>Owner scope check` is enabled and configured with a valid URL, API Connect makes a call to allow the content of `x-selected-scope` to refine the scope value. When Owner scope check is enabled, the HTTP response header `x-selected-scope` must be present, or the call fails.

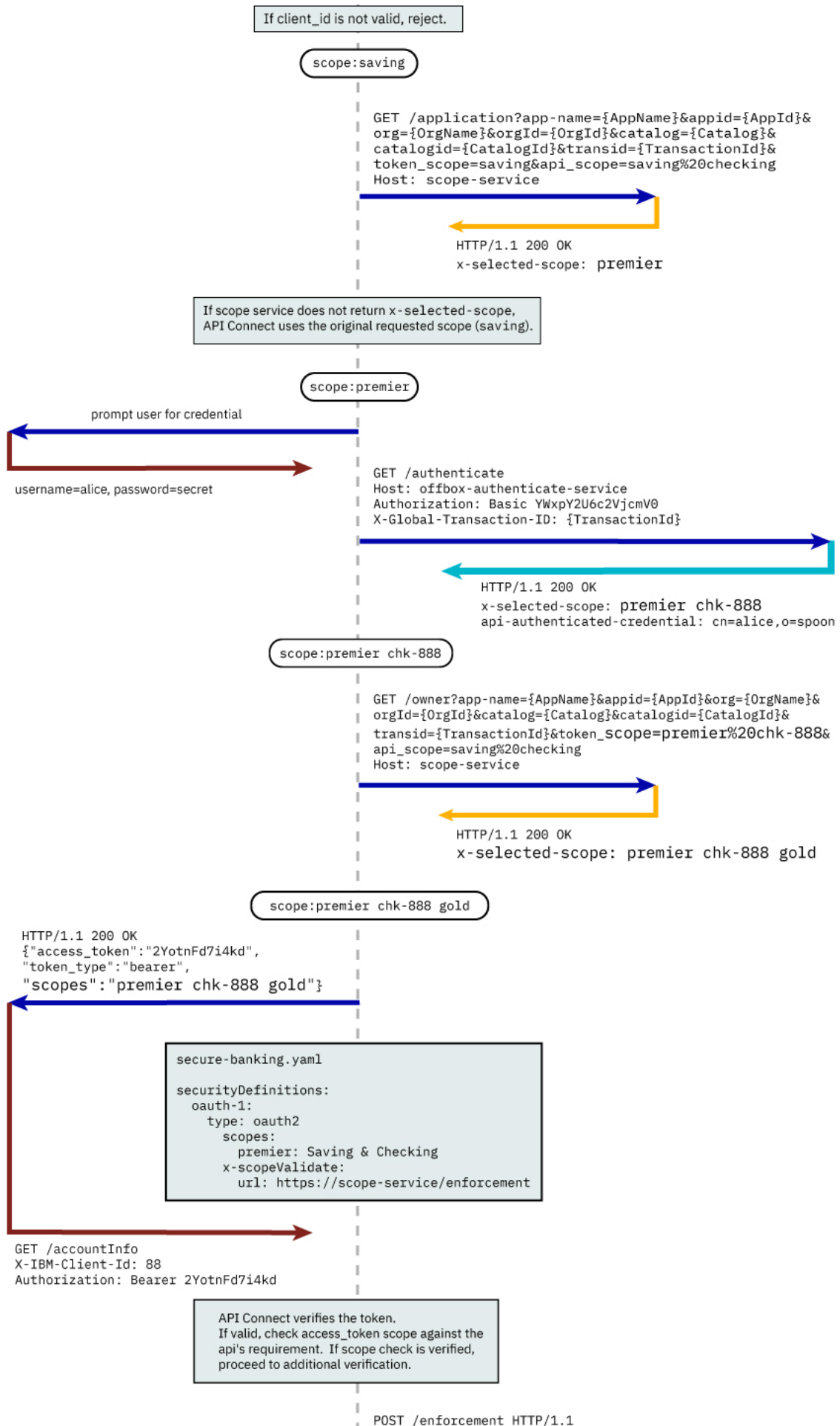
Figure 1. OAuth advanced scope overview

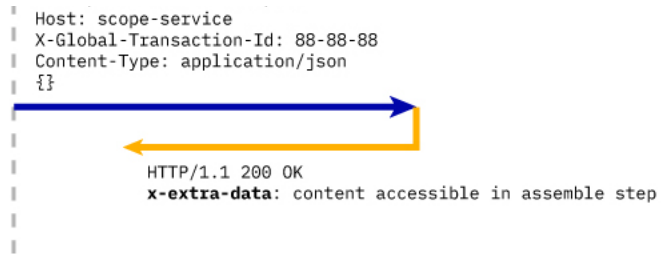


The final scope permission that is granted by the access token is the result of the flow described in paragraphs 1, 2, and 3. [Figure 2](#) shows a more detailed view of the transaction flow with examples that show when `x-selected-scope` provides a new scope value.

Figure 2. OAuth advanced scope detail







If API Connect receives HTTP 200, any HTTP headers in the response that start with "x-" will be accessible at `oauth.advanced-consent.<value>`  
 For the above example, `x-extra-data` is accessible at `oauth.advanced-consent.x-extra-data`.  
 If other than HTTP 200 is returned, it has the same effect as if the access token does not contain necessary permissions to access the resource.

## Consumer API enforcement

Standard scope validation

To access the API `/getaccount` the application must send a `GET` request with an access token that contains the scope, or scopes, defined in the OAuth provider.

```
GET /getaccount
HTTP/1.1
Host: server.example.com
X-IBM-Client-Id: 8888-8888-8888
Authorization: Bearer AAEkNjVkcWlYyJgtOWY5ZS00YWQwLWlYyZktZ
```

The following application OpenAPI file `secure-banking.yaml` defines the scope, or scopes, that must exist in the token to be granted access to the API `/getaccount`.

```
secure-banking.yaml
securityDefinitions:
  scope-only:
    type: oauth2
    description: ''
    flow: implicit
    authorizationUrl: ''
    scopes:
      checking: 'Checking Account'
      saving: 'Saving Account'
      mutual: 'Mutual Fund Account'
security:
  - scope-only:
    - checking
  - scope-only:
    - saving
    - mutual
```

In the examples, the access token `AAEkNjVkcWlYyJgtOWY5ZS00YWQwLWlYyZktZ` is able to access the API because it contains one, or a combination of `scope-only` that is defined in `secure-banking.yaml` such as:

- `checking`
- `saving mutual`
- `checking saving mutual`

Advanced Scope Check

Administrators can enable an additional scope check by configuring the consumer API property Advanced Scope Check URL that becomes `x-scopeValidate` as shown in the following OpenAPI file example.

```
securityDefinitions:
  advanced-scope-only:
    type: oauth2
    description: ''
    flow: implicit
    authorizationUrl: ''
    scopes:
      checking: 'Checking Account'
      saving: 'Saving Account'
      mutual: 'Mutual Fund Account'
  x-scopeValidate:
    url: 'https://advanced-scope-check.bk.com/validate-scope'
    tls-profile: 'ssl-client'
```

After API Connect successfully verifies the access token against any scope requirement, API Connect will make an `HTTP POST` request to the `x-scopeValidate` endpoint similar to the following example. Response code `HTTP 200` from the endpoint indicates a success. Any other response code, or a connection error, is treated as a permission error for the token.

```
POST /validate-scope?app-name=..&appid=..&org=..&orgid=..&catalog=..&catalogid=..&transid=..
HTTP/1.1
Host: advanced-scope-check.bk.com
Content-Type: application/json

{"context-root" : checking,
 "resource" : accountinfo,
 "method" : GET,
```

```

"api-scope-required" : [jointaccount],
"access_token" : {"client_id" : "2cd71759-1003-4a1e-becb-0474d73455f3",
  "not_after" : 174364070,
  "not_after_text" : "2017-07-11T02:27:50Z",
  "not_before" : 174360470,
  "not_before_text" : "2017-07-11T01:27:50Z",
  "grant_type" : "code",
  "consented_on" : 1499736470,
  "consented_on_text" : "2059-07-11T01:27:50Z",
  "resource_owner" : "cn=spoon,email=spoon@poon.com",
  "scope" : "jointaccount mutual",
  "miscinfo" : "[r:gateway]"
}
}

```

An example of successful response follows.

```

HTTP/1.1 200 OK
Cache-Control: no-store
Pragma: no-cache
X-Custom-For-Assemble-Process: audit

```

Any HTTP response header that begins with "x-" is kept as the context variable `oauth.advanced-consent`. Based on the example successful response, `X-Custom-For-Assemble-Process: audit` becomes `oauth.advanced-consent.x-custom-for-assemble-process`, and can be accessed in the assemble step.

Additional validation options

You can optionally use additional fields for validation:

Request Header

Defines the regular expression to match against **request** headers. Matching headers are included in the request to the advanced scope validation endpoint.

Response Context Variable

Defines the regular expression to match against **response** headers. Matching headers are saved as context variables in the format `oauth.advanced-consent.*`.

## Related information

- [IETF RFC 6749](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tokens

Tokens can be managed using refresh tokens and revocation URLs. You can extend the life of tokens using refresh tokens. You can end the life of a token by specifying a revocation URL.

This section contains information regarding token management, including refreshing and revoking tokens, redirecting, and managing tokens with the DataPower Gateway.

- [Refresh tokens](#)  
If you are using OAuth authentication, you can enable refresh tokens. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope.
- [Token revocation](#)  
In IBM® API Connect, you can revoke or refresh tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.
- [Authenticating and authorizing through a redirect URL](#)  
You can use a service that is hosted externally from IBM API Connect to collect authentication and authorization details from your user when an application requests access on that user's behalf.
- [Token management with the DataPower Gateway \(v5 compatible\)](#)  
API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.
- [Token management with the DataPower API Gateway](#)  
API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Refresh tokens

If you are using OAuth authentication, you can enable refresh tokens. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope.

When you are using OAuth authentication, API requests must include a valid access token, by using the Authorization HTTP header. Access tokens that are issued by the IBM® API Connect Token Endpoint are valid for 3600 seconds (1 hour) by default, as indicated by the `expires_in` property that is returned on the token request. The following code block shows an example API request with an Authorization header:

```
GET /bankingApi/accountSummary?client_id=32427ce5-bb7c-48a7-9de3-4bb629091103
HTTP/1.1
Accept: application/json
Host: api.ibm.com
Authorization: Bearer AAEFYy1hbGx1hdS5nVX4x6iTL2sb3ymBivQb...
```

After an access token expires, if the option is enabled in the OAuth provider configuration Tokens<sub>2</sub>.Refresh tokens screen, the application uses refresh tokens. Each refresh token is valid for approximately 31 days after it is issued (or for the Time to Live time period specified) and can be used only once to request a new access token. Along with the new access token, a new refresh token is also returned. For details on how to enable refresh tokens, see [Configuring a native OAuth provider](#).

If the access token is expired and the application does not have a refresh token, it must restart the OAuth exchange by using the choice of Grant Type(s) allowed by the OAuth provider.

Note: Refresh tokens are not supported for a user in an OIDC user registry when accessing the Cloud Manager or API Manager user interfaces.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Token revocation

In IBM® API Connect, you can revoke or refresh tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.

---

## Token revocation using an external third party service

This topic describes token revocation using a third party, external service. This option is configured in the Native OAuth provider, using the Token Management<sub>2</sub>.Type = Third party screen. The revocation URL is an endpoint that links to an external service which contains information about access or refresh tokens. API Connect is involved in the initial creation and validation of tokens. When an OAuth revocation URL is present, API Connect calls the URL to determine if the associated token can be trusted. The token server then checks a token *blacklist* (a data store of inactive tokens) to ensure that the token is still valid. If the token is still valid, API Connect continues the processing.

---

## Examples

A number of revocation examples follow. The first shows a sample fetch request and the response from a remote revocation URL.

---

## GET request and response

In this example, an API Gateway server issues a GET request to the Revocation URL and receives a result. It shows that different resource owners (Laura and Emily) can revoke all tokens when using the same application (client ID).

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
GET <revocationURL> HTTP/1.1
User-Agent: IBM-APIManagement/4.0
Accept: application/xml; text/xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>

<!--
Access Tokens and/or Refresh Tokens that are revoked can be individually listed.
To keep this list small, please only include access tokens and refresh tokens that are valid.
For access tokens, any token older than 20 minutes is no longer valid.
For refresh tokens, any token older than 44700 minutes is no longer valid.
-->

<token type="access">AAETb2F1dGgtcmV2b2t1LLWN1c3RvbfZaRlVbnPSc1...</token>
<token type="refresh">fZaRlVbnPSc1UGTjCRdq4mPbOosD2+aZIKbJ6bTeW...</token>

<!--
If a resource owner has revoked all tokens issued to a given application, please
list them as shown here.
-->

<resource-owner client-id="760d75a2-44b1-4485-8c6f-0d264fcf7398">laura</resource-owner>
<resource-owner client-id="760d75a2-44b1-4485-8c6f-0d264fcf7398">emily</resource-owner>

</oauth-revocation>
```

---

## POST request and response

The next example shows a post request and response.

Request:

```
POST <revocationURL> HTTP/1.1
User-Agent: IBM-APIManagement/4.0
Content-Type: application/xml
```

```
<?xmlversion="1.0" encoding="UTF-8"?>
<token>
  <token_type>bearer</token_type>
  <access_token>AAENYy1hbGwtcmVmcmVzaOfNeQKX8ZeojsBY9v0FI7/OerQvzKHq...</access_token>
  <expires_in>3600</expires_in>
  <scope>3600</scope>
  <resource-owner>alice</resource-owner>
  <client_id>83d9cdcd-ba72-4d00-abae-005da8da5fb1</client_id>
</token>
```

Response:

```
HTTP/1.1 200 OK
```

## Provide token information on revocation request

---

In this example, the application calls an API and passes a bearer token. In response, the Gateway fetches the revocation URL and provides information on the token being verified.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Alternatively, the same process occurs when using a refresh token to issue a new access token. The application sends a refresh request to the token service. The Gateway then fetches the revocation URL, providing information on the refresh token being verified.

Gateway:

```
GET <revocationURL>GET HTTP/1.1
refresh-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Note: If you are using a third party OAuth provider then, for API Calls with bearer tokens, when Introspect URL is enabled on the API, the revocation URL is not applicable. Instead, the third party endpoint must validate the token and also check for revocation before returning a 200 OK response to the Gateway.

## Revoking tokens issued to Alice up to and including a specific date

---

On May 1st, Alice loses her phone and needs to reset her password. As a result, the token provider wants to revoke every token issued before Alice lost her phone. In this example, the Gateway sends a GET request to the revocation service. The revocation service replies confirming revocation of the specified tokens.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>
  <resource-owner before="2015-05-01T09:30:10Z">alice</resource-owner>
</oauth-revocation>
```

## Revoking all tokens issued up to and including a specific date

---

Under certain catastrophic conditions, you may need to revoke all tokens issued up to and including a specific date, for example, May 1st. In this example, the Gateway sends a GET request to the revocation service. The revocation service replies confirming revocation of the specified tokens.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>
  <everytoken before="2015-05-01T09:30:10Z" />
</oauth-revocation>
```

## Putting it all together

---

The following shows the examples contained in this topic executed in a single action:

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
```

client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398  
resource-owner: alice

Revocation Service:

HTTP/1.1 200 OK  
Content-Type: application/xml  
Cache-Control: public, max-age=120  
Date: Fri, 08 May 2015 21:49:03 GMT

```
<?xml version="1.0" encoding="UTF-8"?>  
<oauth-revocation>  
  <resource-owner before="2015-04-08T09:30:10Z">mary</resource-owner>  
  <resource-owner before="2015-04-12T09:30:10Z">john</resource-owner>  
  <resource-owner before="2015-04-13T09:30:10Z">kevin</resource-owner>  
  <resource-owner before="2015-04-01T09:30:10Z">alice</resource-owner>  
</oauth-revocation>
```

Notes:

- In the previous example, there are no entries older than one month in the response (the maximum life of a refresh token).
- Each response is cached for up to two minutes according to response's directive.
- The `before` attribute uses the `xs:dateTime` syntax.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authenticating and authorizing through a redirect URL

You can use a service that is hosted externally from IBM® API Connect to collect authentication and authorization details from your user when an application requests access on that user's behalf.

### Before you begin

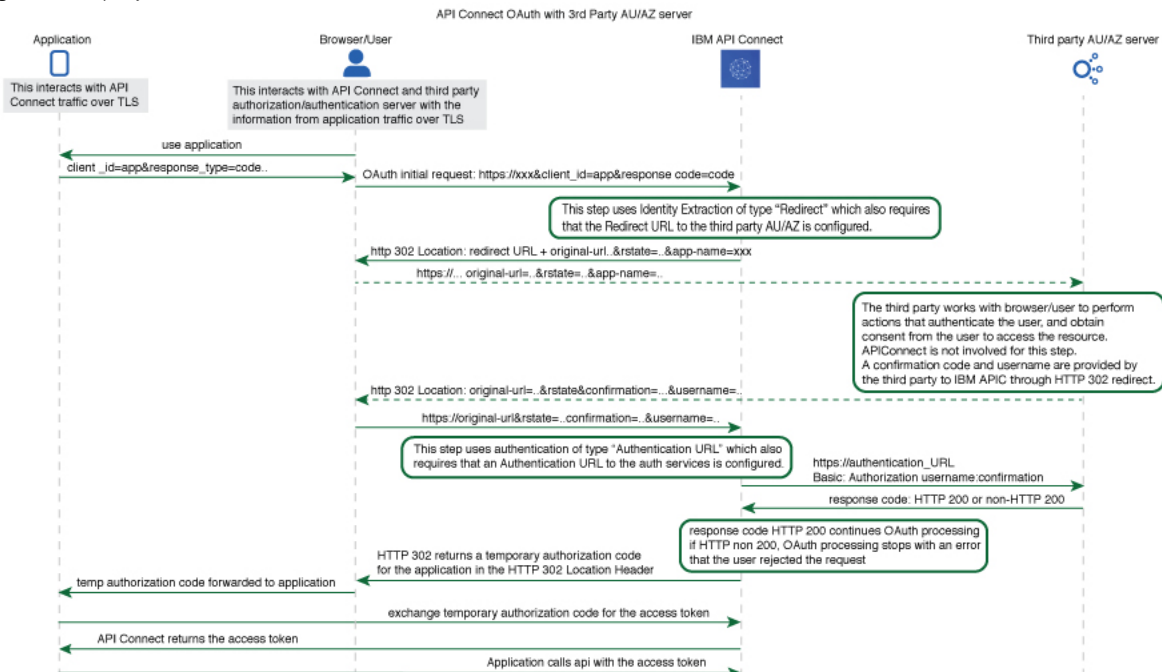
To complete this task, you will need to either create or have created an OAuth security definition that uses Implicit grant type or Access Code (Authorization Code) grant type. For more information, see [Creating an OAuth security definition](#).

### About this task

If you use methods for authentication that are not supported by API Connect, you can redirect users to a suitable URL at which they can authenticate. The user is then returned to the OAuth process after authentication and authorization have been confirmed.

The following illustration indicates the transaction flow for third party authentication.

Figure 1. Third party authentication (AU) and authorization (AZ) transaction flow



1. The application initiates a request to access an API protected with a third-party entity. API Connect redirects the application with an **HTTP 302** redirect based on **identity extraction** ->



- redirect** -> **redirect-url**, for user authentication (and optional authorization).
- The application communicates directly with the third-party entity to gather user identity. API Connect is not involved in this communication. After the third-party entity finishes processing authentication (and optional authorization), it returns an **HTTP 302** redirect that uses the **original-url** from the request, with the username and confirmation code appended.
  - API Connect receives the request that includes the username and confirmation code, and communicates with the authentication URL, based on **authentication** -> **x-ibm-authentication-url**, to confirm user identity before the request is completed.
  - An **HTTP 200** response from the third-party entity allows API Connect to continue the OAuth 2.0 request process as if the owner is authenticated. The request is then processed according to the **grants** type.
    - **accessCode** returns a temporary code to the application.
    - **implicit** returns the access token to the application.
- For any response other than HTTP 200, the request fails with a statement added to the error log.

## Procedure

To create an external form, and to indicate the URL to which API Connect will redirect users, complete the following instructions:

- Create your service for authentication and authorization. You will use the URL of the landing page of this service as your redirect URL.
  - To include elements in your form that are provided by API Connect, use the query parameters from the URL that your user is redirected to. When the user is redirected to your page, the URL they are sent to contains the following query parameters:

**app-name**  
The name of the application requesting access, as provided through the Developer Portal.

**appid**  
The id of the application requesting access.

**catalog**  
The name of the catalog where the product is being used by the application.

**catalogid**  
The id of the catalog where the product is being used by the application.

**catalogtitle**  
User-friendly display name for the catalog.

**org**  
The name of the consumer organization that hosts the application.

**orgid**  
The id of the consumer organization that hosts the application.

**orgtitle**  
User-friendly display name for the organization.

**original-url**  
The original URL that the user was directed to by the application, including query parameters from the original URL that are necessary for standard OAuth 2.0 requests. You can include these parameters in your service to provide information to the user. Additionally the **state\_nonce** is appended. The **state\_nonce** is a hash code generated by API Connect for verification purposes. The URL is URL-encoded and should be decoded before further use, the **state\_nonce** should remain unchanged.

**provider**  
The name of the API provider organization.

**providerid**  
The id of the API provider organization.

**providertitle**  
User-friendly display name for the provider organization.

**requested-scope**  
[optional] If [Application Scope check](#) is enabled and replaces the **scope** from the initial application request, this field holds the **scope** value from the initial application request, and the new replacement scope value is put into **original-url**.

**transid**  
transaction id used in the GW for the transaction which trigger this call

The URL to which the user is sent to when they are redirected to your page has the following form:

**Redirect\_URL?original-url=Original\_URL&state\_nonce=R\_State&app-name=Application\_Name**

where all variables are as described previously. The Redirect URL does not have a size limit enforced by API Connect.

- Create the stages of authentication, authorization, and any intermediate stages that you require to take place before you allow access to the application. Upon completion of these stages, redirect the user to the *Original\_URL* and append a user name, their confirmation code, and the application name to be evaluated for access grant or denial by API Connect. The confirmation code does not have a size limit enforced by API Connect. Original URL requires the following form:

**Original\_URL&username=User\_Name&confirmation=Confirmation\_Code**

where all variables are as described previously.

For example:

**https://your\_gateway.com/your\_org/your\_catalog/your\_api/oauth/authorize?response\_type=code&redirect\_uri=https://example.com/redirect&scope=/your\_api&client\_id=5af57a4a-6db9-4141-ad08-5709432af66e&state\_nonce=HoIbRG+6bZtq1B7LDkq4gj1D3SHKq1CbnYdHs/bMz2Y=&username=spoon&confirmation=12345678**

- To send your own error responses after the authentication and authorization service, redirect the user to the *Original\_URL* and append an error code. You can also append a user name. Use the following form:

**Original\_URL&username=User\_Name&error=Error\_Response**

where *Error\_Response* is the message you wish to send and all other variables are described as previously.

For example:

```
https://your_gateway.com/your_org/your_catalog/your_api/oauth/authorize?
response_type=code&redirect_uri=https://example.com/redirect&scope=/your_api&client_id=5af57a4a-6db9-4141-ad08-
5709432af66e&state_nonce=HoIbRG+6bZtq1B7LDkq4gj1D3SHKglCbnYdHs/bMz2Y=&username=User&error=access_denied
```

2. Create a service to validate the confirmation code and user name. API Connect makes a GET call to your authentication URL after the user is redirected back to the authorization URL. When the call is made, it includes in its authorization header the user name and confirmation code you supplied previously. Confirm that these are correct and respond with an HTTP success code such as 200 OK if you want to allow access, or non-200 HTTP response code, such as 401 Unauthorized to deny access.
3. In your OAuth provider configuration, supply the redirect URL that is used in Step 1 and the authentication URL that is used in Step 2.  
For more information on configuring an OAuth Provider, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Token management with the DataPower Gateway (v5 compatible)

API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

In order to manage tokens with the DataPower® Gateway (v5 compatible), you must set the Token Management Type to Native in your Native OAuth provider configuration. See [Configuring token management and revocation for a native OAuth provider](#) when using Cloud Manager or [Configuring token management and revocation for a native OAuth provider](#) when using API Manager.

Also, the DataPower quota enforcement server must be enabled on the DataPower Gateway to use the distributed cache to manage tokens. See [Quota enforcement](#).

When distributed cache support is enabled, replay protection is provided across the gateway cluster through the quota enforcement server. This support ensures that the same token cannot be reused across the members of the quota enforcement peer group.

Important:

If you have a cluster of DataPower Gateway servers, the OAuth data synchronization behavior across the servers depends on whether or not you enable revocation:

- If you enable revocation, API Connect uses the DataPower quota enforcement server, and OAuth data is synchronized across the servers. If an access token is obtained from one server, the OAuth data synchronization ensures that the same authorization code cannot be used to obtain an access code from another server. You must ensure that the DataPower quota enforcement server is configured.
- If you disable revocation, API Connect does **not** use the DataPower quota enforcement server and OAuth data does not synchronize across the cluster of DataPower Gateway servers. Therefore, to prevent the same authorization code being used to obtain an access code from more than one server you must configure DataPower to synchronize OAuth data across the servers, by using the DataPower Gateway console.

To configure DataPower to synchronize OAuth data, ensure that the DataPower quota enforcement server is configured. For more information, see [Configuring the quota enforcement server](#).

---

## Resource owner revocation

When Resource owner revocation path is selected in the Token Management screen, the configuration inserts two REST API calls to /oauth2/issued.

- An HTTP GET operation that retrieves a list of all granted permissions for a specific user.
- An HTTP DELETE operation that revokes an application for a specific user.

The setting inserts header-based security definitions of client ID and client secret as shown in the **View permissions example**. The API call to revoke a given application for a given user is shown in the **Revoke permissions example**.

View permissions example

To list all the applications granted by user `cn=spoon,o=ibm` with username `spoon` and password `spoon` using a registered administration application of `5287fe53-8747-438a-8262-681ec75b79c5`.

- Request:

```
GET /oauth2/issued
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
[
  {
    "clientId": "7369ad66-5674-b7d3-4567-de35283421aca",
    "owner": "cn=spoon,o=ibm",
    "clientName": "PetStore Application",
    "scope": "listpet",
    "issuedAt": 1503327054,
    "consentedOn": 1503327054,
    "expiredAt": 1503330654,
    "refreshTokenIssued": false,
    "appId": "d2031f0f27339315333734ab9",
```

```

    "org": "PetStoreOrg",
    "orgTitle": "Katie Pet Grooming Inc",
    "orgId": "5887803de4b06e6998c4b2c7",
    "provider": "SuperStore",
    "providerTitle": "Simon SuperStore",
    "providerId": "5887803de4b06e6998c4b2c7",
    "catalog": "publicapi",
    "catalogTitle": "For public",
    "catalogId": "5887803de4b06e6998c4b2d3"
  },
  {
    "clientId": "a8746323-9825-a842-8736-abd8202356ac8",
    "owner": "cn=spoon,o=ibm",
    ...
  }
]

```

#### Revoke permissions example

To revoke application `a8746323-9825-a842-8736-abd8202356ac8` by owner `cn=spoon,o=ibm`.

- Request

```

DELETE /oauth2/issued?client-id=a8746323-9825-a842-8736-abd8202356ac8
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=

```

- Response

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }

```

## Client revocation

When Client revocation path is selected in the Token Management screen, the result is the following:

This option inserts one REST API call to `/oauth2/revoke`, which supports OAuth 2.0 [IETF RFC 7009](#). An HTTP POST operation that an application can send to this API to revoke either an `access_token`, or `refresh_token` with `token_type_hint` as shown in the following examples:

#### Revoke `access_token`

- Request:

```

POST /oauth2/revoke
HTTP/1.1
Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=access_token&token=AAIHZGVmYXVsdD1-KqwD0Yc3EDn94lSWX14xuR...

```

- Response:

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }

```

#### Revoke `refresh_token`

- Request:

```

POST /oauth2/revoke
HTTP/1.1
Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=refresh_token&token=.....

```

- Response:

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Token management with the DataPower API Gateway

API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

In order to manage tokens with the DataPower® API Gateway, you must set the Token Management Type to Native in your Native OAuth provider configuration. See [Configuring token management and revocation for a native OAuth provider](#) when using Cloud Manager or [Configuring token management and revocation for a native OAuth provider](#) when using API Manager.

For token management with the DataPower API Gateway, you must configure the API Security Token Manager on the gateway. Complete the following steps:

1. Log in to the DataPower administration console, selecting apiconnect for the domain, and WebGUI for the Graphical Interface.
2. In the search box, enter API Security Token Manager, then click on the API Security Token Manager link that displays in the search results.
3. For the Administrative state, select enabled.
4. Click the + icon alongside the Gateway Peering field to create a new gateway peering object.
5. Provide a Name and a Local address.
6. In the Local port and Monitor port fields, provide values that aren't already in use by services for the Local address.
7. Ensure that Peer group mode is selected.
8. Alongside the Peers field, use the add button and accompanying field to add at least two peers, to ensure that quorum is achieved.
9. If the Enable SSL option is selected, select a value for the Identification credentials.
10. For Persistence location, select a setting other than memory.
11. When done, click Apply.
12. Repeat steps 1 to 11 for each DataPower API Gateway in the peer group.
13. When done, click Apply, then click Save Configuration to save your changes.

For more information, see [Defining the API security token manager](#) and [Creating a gateway peering instance](#).

## Resource owner revocation

When Resource owner revocation path is selected in the Token Management screen, the configuration inserts two REST API calls to /oauth2/issued.

- An HTTP GET operation that retrieves a list of all granted permissions for a specific user.
- An HTTP DELETE operation that revokes an application for a specific user.

The setting inserts header-based security definitions of client ID and client secret as shown in the **View permissions example**. The API call to revoke a given application for a given user is shown in the **Revoke permissions example**.

View permissions example

To list all the applications granted by user `cn=spoon,o=ibm` with username `spoon` and password `spoon` using a registered administration application of `5287fe53-8747-438a-8262-681ec75b79c5`.

- Request:

```
GET /oauth2/issued
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dn7nW5km5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
[
  {
    "clientId": "7369ad66-5674-b7d3-4567-de35283421aca",
    "owner": "cn=spoon,o=ibm",
    "clientName": "PetStore Application",
    "scope": "listpet",
    "issuedAt": 1503327054,
    "consentedOn": 1503327054,
    "expiredAt": 1503330654,
    "refreshTokenIssued": false,
    "appId": "d2031f0f27339315333734ab9",
    "org": "PetStoreOrg",
    "orgTitle": "Katie Pet Grooming Inc",
    "orgId": "5887803de4b06e6998c4b2c7",
    "provider": "SuperStore",
    "providerTitle": "Simon SuperStore",
    "providerId": "5887803de4b06e6998c4b2c7",
    "catalog": "publicapi",
    "catalogTitle": "For public",
    "catalogId": "5887803de4b06e6998c4b2d3"
  },
  {
    "clientId": "a8746323-9825-a842-8736-abd8202356ac8",
    "owner": "cn=spoon,o=ibm",
```

```
    ...
  }
]
```

Revoke permissions example

To revoke application `a8746323-9825-a842-8736-abd8202356ac8` by owner `cn=spoon,o=ibm`.

- Request

```
DELETE /oauth2/issued?client-id=a8746323-9825-a842-8736-abd8202356ac8
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uC1xT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

## Client revocation

When Client revocation path is selected in the Token Management screen, the result is the following:

This option inserts one REST API call to `/oauth2/revoke`, which supports OAuth 2.0 [IETF RFC 7009](#). An HTTP POST operation that an application can send to this API to revoke either an `access_token`, or `refresh_token` with `token_type_hint` as shown in the following examples:

Revoke `access_token`

- Request:

```
POST /oauth2/revoke
HTTP/1.1

Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uC1xT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=access_token&token=AAIHZGVmYXVsdD1-KqwD0Yc3EDn941SWX14xuR....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

Revoke `refresh_token`

- Request:

```
POST /oauth2/revoke
HTTP/1.1

Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uC1xT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=refresh_token&token=.....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authentication URL user registry

You can use an Authentication URL user registry to specify a REST authentication service that manages user authentication, and optionally provides additional metadata to be embedded in the token.

This support can optionally enable any of the following:

- Providing the authenticated credential to IBM® API Connect. For example, the user logs-in with user name: `spoon`, and password: `fork`. When the user is authenticated, the credential becomes `cn=spoon,o=eatery`. The credential is kept in the OAuth `access_token` to represent the user.
- Providing metadata support. Allow extra metadata to be stored in the `access_token`.
- Overriding the `scope` that the application receives after a successful OAuth protocol processing. By responding with a specific header, the Authentication URL endpoint can replace the `scope` value that the application receives. For example, you can provide a specific resource owner an account number within the `scope` header response for use in future processing steps.

When you call the Authentication URL user registry, the API Connect gateway sends a GET request with HTTP headers and then processes any HTTP response from the URL. For authentication, a REST authentication service is expected at the Authentication URL.

The following response from the REST authentication service indicates that user authentication is successful and that API Connect will use `cn=spoon,o=eatery` as the user identity.

```
HTTP/1.1 200 OK
Server: example.org
X-API-Authenticated-Credential: cn=spoon,o=eatery
```

For information on how to configure a User Security policy in an API assembly for use with an Authentication URL user registry, see [User Security policy](#).

For an example of an OAuth provider configuration that uses an Authentication URL user registry, see [Example - using multiple OAuth policies in an OAuth provider assembly](#).

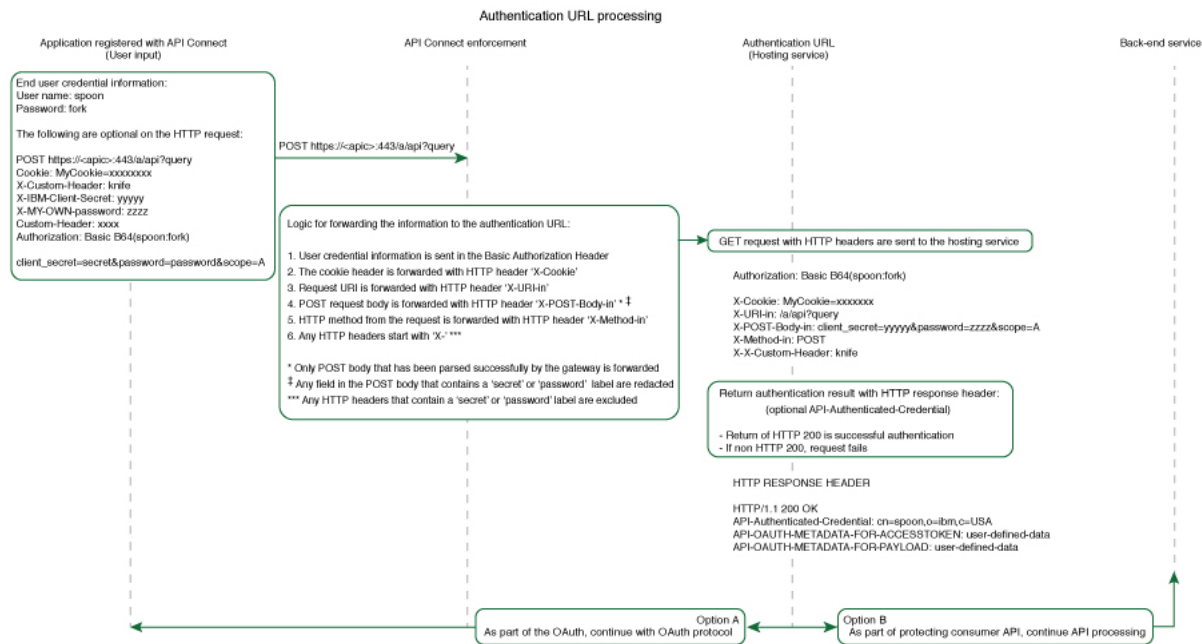
API Connect considers any non-200 HTTP response code a failed user authentication attempt.

When an Authentication URL user registry is invoked, two HTTP response headers are available that include metadata in the access token or the response payload that contains the access token. For more information, see [OAuth external URL and authentication URL](#). The two metadata response headers are:

```
API-OAUTH-METADATA-FOR-ACCESSTOKEN
API-OAUTH-METADATA-FOR-PAYLOAD
```

When an Authentication URL is invoked, an HTTP response header is available to override the requested `scope` from the application. For more information, see [Scope](#). The response header is:

```
x-selected-scope
```



If you are using the DataPower® API Gateway rather than the DataPower Gateway (v5 compatible), this diagram is provided for guidance only and is not fully accurate for this release.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Custom forms for user security

You can create custom forms for the authorization and identity extraction phase of OAuth.

## About this task

The Native OAuth provider configuration provides the capability for requiring additional authentication and authorization steps for user security. Custom HTML forms may be created to extract the identity of the user and to authorize the user. This capability applies to Implicit, Resource owner password, and Access code grant types.

- [Creating a custom HTML login form for user security](#)  
Custom HTML forms can be created for user security during the identity extraction stage in OAuth.
- [Creating a custom HTML authorization form for user security](#)  
Custom HTML forms can be created for user security during the authorization stage in OAuth.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a custom HTML login form for user security

Custom HTML forms can be created for user security during the identity extraction stage in OAuth.

---

### Before you begin

The Native OAuth provider configuration includes Identity Extraction when using the Implicit, Access code, or Resource owner password grant types. You have the option to select how to extract the user credential and one of the choices is Custom HTML Form. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager. For more information, see [Configuring a native OAuth provider](#). This topic describes how to create the Custom HTML form for identity extraction.

---

### About this task

During three-legged OAuth definitions (Implicit flow, Resource owner password flow, and Access (Authorization) code flow, the user is presented with a form for signing in to the service provided by the API. You can present a custom form or a default form. Your custom form must fulfill certain requirements.

Important: The fields used by IBM® API Connect to inject information into your form have case-sensitive field names.

---

### Procedure

To create a custom sign-in form for your Native OAuth provider, complete the following steps:

1. Create a well formed XHTML document that will be parsed and transformed by API Connect to inject hidden fields.
2. For your XHTML form, set the method as POST, the encoding type as `application/x-www-form-urlencoded`, and the action as `authorize`. Add any other parameters that you require.  
For example:

```
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">
```

3. Create a text input field that is named `username` and create a password input field named `password`.
4. Add the line `<EI-INJECT-HIDDEN-INPUT-FIELDS>`. This third element is a placeholder that API Connect replaces with input fields to complement the user-submitted data.
5. Create a button to initiate the sign-in process.  
For example:

```
<button  
id=" _home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.apionprem.doc_oauth_custom_login_form_login_but  
ton" type="submit" name="login" value="true">Log in</button>
```

6. Optional: Add text that is displayed the first time that the user visits the sign-in page. Use the tag `<EI-LOGINFIRSTTIME>` for the text that you want to display.
7. Optional: Add text that appears when the user is returned to the sign-in page if they fail to authenticate. Use the tag `<EI-LOGINFAILED>` for the text that you want to display.
8. Optional: Have an error message displayed when an error in the custom form prevents it from being displayed to the user correctly. Use the tag `<EI-INTERNAL-CUSTOM-FORM-ERROR/>`; the message text is generated automatically. You should detect such errors during testing to prevent this error message being displayed to the end user.
9. Optional: You can add elements that are loaded from external sources, such as images or JavaScript.
10. Insert spacing and other features as you require. Completing Steps [1](#) through to [8](#) results in a form similar to the following example:

```
<html lang="en" xml:lang="en">  
<head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/></head>  
<body>  
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">  
<h1>Please sign in</h1>  
<p>Username </p>  
<p ><input type="text" name="username" required="required" /> </p>  
<p>Password </p>  
<p ><input type="password" name="password" required="required" /> </p>  
<EI-INJECT-HIDDEN-INPUT-FIELDS/>  
<p > <button  
id=" _home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.apionprem.doc_oauth_custom_login_form_login_but  
ton" type="submit" name="login" value="true">Log in</button> </p>  
  
<EI-LOGINFIRSTTIME>  
<p>If you have forgotten your user name or password, contact your system administrator.</p>  
</EI-LOGINFIRSTTIME>  
  
<EI-LOGINFAILED>
```

```

    <p>At least one of your entries does not match our records.
      If you have forgotten your user name or password, contact your system administrator.</p>
  </EI-LOGINFAILED>

  <EI-INTERNAL-CUSTOM-FORM-ERROR/>

</form>
</body>
</html>

```

- Make your form available at a URL of your choice.
- If you have not already done so, configure your Native OAuth provider to use a Custom HTML form for identity extraction and provide the URL at which your form is available. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

## Related tasks

- [Creating a custom HTML authorization form for user security](#)

## Related information

- [Configuring API security](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Creating a custom HTML authorization form for user security

Custom HTML forms can be created for user security during the authorization stage in OAuth.

## Before you begin

The Native OAuth provider configuration includes user authorization when using the Implicit, Access code, or Resource owner password grant types. You have the option to select how to authorize application users and one of the choices is Custom HTML Form. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager. This topic describes how to create the Custom HTML form for authorization.

## About this task

During three-legged OAuth definitions (Implicit flow, Resource owner password flow, and Access (Authorization) code flow, the user is presented with a form through which they grant permission to an application to access their data through the API on their behalf. You can present a custom form or a default form. Your custom form must fulfill certain requirements.

Important: The fields used by IBM® API Connect to inject information into your form have case-sensitive field names.

## Procedure

To create a custom authorization form for your Native OAuth provider, complete the following steps:

- Create a well-formed XHTML document. This will be parsed and transformed by API Connect to inject hidden fields.
- For your XHTML form, set the method as POST, the encoding type as application/x-www-form-urlencoded, and the action as authorize. Add any other parameters that you require.

For example:

```
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">
```

- Add the line <AZ-INJECT-HIDDEN-INPUT-FIELDS/>. This line is a placeholder that API Connect will replace with input fields necessary for the completion of the OAuth process.
- Create two buttons with the following code so that the user can grant or deny permission. Edit the text to suit your preferences.

```
<button class="cancel" type="submit" name="approve" value="false">No Thanks</button>
<button class="submit" type="submit" name="approve" value="true">Allow Access</button>
```

- Optional: Display an error message when an error in the custom form prevents it from being displayed to the user correctly. Use the tag <AZ-INTERNAL-CUSTOM-FORM-ERROR/>; the message text is generated automatically. You should detect such errors during testing to prevent this error message being displayed to the end user.
- Optional: You can add to the form HTML elements that will load features from external sources, such as images or JavaScript. For example, <script src="http://www.example.com/example.js" />
- Insert spacing and additional elements as you require. Completing Steps 1 through to 6 results in a form similar to the following example:

```
<html lang="en" xml:lang="en">
  <head><title>Request for permission</title></head>
  <body class="customconsent">
    <div>
      <div>
        <form method="post" enctype="application/x-www-form-urlencoded" action="authorize">
          <AZ-INJECT-HIDDEN-INPUT-FIELDS/>
          <p>Greeting...</p><DISPLAY-RESOURCE-OWNER/>
        </form>
      </div>
    </div>
  </body>
</html>
```



```
<p>This app </p><OAUTH-APPLICATION-NAME/><p> would like to access your data.</p>
<div>
  <button class="cancel" type="submit" name="approve" value="false">No Thanks</button>
  <button class="submit" type="submit" name="approve" value="true">Allow Access</button>
</div>
</form>
</div>
<AZ-INTERNAL-CUSTOM-FORM-ERROR/>
</div>
</body>
</html>
```

8. Make your form available at a URL of your choice.
9. If you have not already done so, configure your Native OAuth provider to use a Custom HTML Form for authorization for User Security. Provide the URL as the endpoint at which your form is available. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

## Related tasks

---

- [Creating a custom HTML login form for user security](#)

## Related information

---

- [Configuring API security](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Securing an API with a JSON Web Token

There are two methods to secure your API with a JSON Web Token. You can use the **jwt-generate** command, or you can use a token that has been generated external to IBM® API Connect.

### About this task

---

JSON Web Token (JWT) is an OAuth 2.0 compliant method of authentication that can be useful to secure your API in API Connect.

You can secure your API with a JSON Web Token by using either of the following methods:

- Generate a token through the **jwt-generate** command, and then augment the response payload with your generated token replacing the **id** token.
- Use a token that was generated outside of API Connect and include it into the response payload, by using the metadata URL.

### Procedure

---

Create a **jwt-generate** policy in the assembly.

- a. In API Manager, open the Assembly tab.
- b. Add a **jwt-generate** policy to the assembly for the API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Troubleshooting OAuth

You can find the answers to some common questions in the Developer Portal, or in support communities and forums in DeveloperWorks, on GitHub, or on Youtube.

You can use the following links to go to the topics:

- [Enable an extended error description](#)
- [OAuth 2.0 support in the developer portal test tool](#)
- [OAuth requires quorum in a multi-node cluster](#)

### Enable an extended error description

---

The [OAuth 2.0 Authorization Framework](#) governs how IBM® API Connect is defined. According to the specification, OAuth 2.0 error conditions trigger a payload with error, and an optional field `error_description`. To prevent information leakage, the IBM DataPower® Gateway default setting returns the error only. However, during the development and testing phase of an application, it is useful to find out why an OAuth 2.0 request is being rejected. To enable this behavior, the caller can pass an HTTP request header in the OAuth request:

```
APIm-Debug: true
```

The presence of the header enables the 'just-in-time' debugging for that OAuth transaction, and `error_description` is returned as part of the error condition. The output helps the caller to determine why an OAuth request is rejected by API Connect as shown in the examples. Example error output without 'just-in-time' debugging:

```
{ "error": "invalid_request" }
```

Example error output with 'just-in-time' debugging:

```
{ "error": "invalid_request", "error_description": "Multiple OAuth client credentials are provided" }
```

## OAuth 2.0 support in the developer portal test tool

OAuth 2.0 support is exposed as an API through the provider OpenAPI definition. You can use the test tool that is included with API Connect to test the OAuth 2.0 configuration. For more information, see [Testing an API using the Developer Portal test tool](#).


## OAuth requires quorum in a multi-node cluster

In a multi-node cluster, OAuth operations will fail if quorum is lost. Quorum requires that the number of active nodes is greater than 50% of the total number of nodes in the cluster.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring the cloud settings

Before you can add a provider organization to your cloud, you must specify your cloud settings to configure an email server for notifications as well as user registry options and catalog defaults.

The cloud settings let you define basic configuration for your deployment. To view the current settings, log in to Cloud Manager and click  in the navigation pane.

Tip: When you are familiar with the cloud settings, you can modify them using the [Toolkit command-line interface](#) instead of the user interface. Visit the following pages to configure settings:

### Overview

Provide a title and description for your cloud administrator organization. For more information, see [Change the name of your cloud](#).

### Onboarding

Specify time-out settings for onboarding invitations and password resets. To specify a timeout value, select or type an integer value in the Number field and then choose a unit of time (Seconds, Minutes, or Hours) in the Unit field.

For more information on setting timeouts, see the following topics:

- [Configuring invitation timeouts](#)
- [Configuring the password-reset notification and timeout](#)

### User Registries

Configure one or more user registries to authenticate users of Cloud Manager or API Manager. The user credentials for all users who login to each applications must be stored in the specified registries. For more information on configuring user registries, see [Selecting user registries for Cloud Manager and API Manager](#).

### Roles

Create roles and assign them to members of the Admin organization to specify which tasks members are authorized to perform. For more information on assigning roles, see [Administering members and roles](#).

### Role Defaults

The Role Defaults for the Admin organization are not configurable, but you can customize the base roles for provider organizations and consumer organizations. For more information on customizing base roles, see [Role Defaults overview](#).

### Endpoints

View the list of endpoints configured for API Manager, provider APIs, and consumer APIs during the installation process.

### Notifications

Select the email server for your on-premises cloud notifications, configure notification settings, and specify the sender name and email address to be included in the emails. You can also preview and customize email templates used for notifications. Be sure to configure the email server, because it is needed for sending invitations and other notifications to users. For more information on setting up notifications, see [Setting up notifications](#).

Note: Before you can set up notifications, you must [configure an email server](#) for your deployment.

### Catalog Defaults

Specify one or more gateway services to be provided to every Catalog by default, so that Products (which contain APIs) can be made available to users. When a new Catalog is created in a provider organization, it is assigned to the default gateway service. Any changes you make to these settings do not affect existing Catalogs. For more information on selecting gateway services, see [Configuring the default gateway services for Catalogs](#).

Use the following list to locate instructions for using specific cloud settings:

- [Using the CLI to modify cloud settings](#)  
Use the toolkit CLI to edit the cloud settings.
- [Change the name of your cloud](#)  
You can change the name of your cloud.
- [Configuring an email server for notifications](#)  
Connect to an email server in your API Connect on-premises cloud. The email server sends registration invitations and other event-driven emails. You must configure the email server before you add any provider organizations or register a Portal service. Otherwise, the owner of the organization will not receive the email with the details that they require to access the API Manager.

- [Setting up notifications](#)  
To configure email notifications, configure an email server, sender information, and email templates.
- [Configuring invitation timeouts](#)  
You can set the time period for when registration invitations for Cloud Settings access and the Admin organization expire.
- [Configuring the password-reset notification and timeout](#)  
Configure the time allowed for a user to reset a password, and customize the notification that the user receives.
- [Selecting user registries for Cloud Manager and API Manager](#)  
Select the user registries to authenticate users for Cloud Manager and API Manager.
- [Configuring timeouts for access tokens and refresh tokens](#)  
Configure timeout values for the access tokens and refresh tokens for user registries.
- [Viewing platform and UI endpoints](#)  
A view-only list of platform and UI endpoints is provided.
- [Configuring the default gateway services for Catalogs](#)  
Configure the list of default gateway services that are available to new Catalogs.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using the CLI to modify cloud settings

Use the toolkit CLI to edit the cloud settings.

### About this task

Some cloud settings can be modified in the Cloud Manager UI, but all settings can be modified with the toolkit CLI. You can modify the cloud settings configuration file and upload the changes, or issue commands to modify individual settings directly. If multiple people modify the cloud settings, any newer settings overwrite older settings regardless of the method used to update the settings.

For more information about working with the toolkit CLI, see the following topics:

- For details on using `apic` to modify cloud settings, see [apic cloud-settings:update](#).
- For details on all the cloud settings commands, see [apic cloud-settings](#)

### Procedure

1. Run the following command to log in to a management server as a member of the cloud administration organization:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm realm
```

Cloud settings are stored on a management server and you must log in to the server before you can access the settings with the CLI. For more information on logging in, see [Logging in to a management server](#).

2. Optionally retrieve the current list of cloud settings with the following command:

```
apic cloud-settings:get --server mgmt_endpoint_url --output -
```

where `mgmt_endpoint_url` is the URL of the server where the cloud settings are stored.

3. Add or modify settings using one of the following methods:

- Issue each setting from the command line.  
For simple changes, you can issue commands to modify settings without editing the configuration file.

- Issue the following command to start the update process:

```
apic cloud-settings:update --server mgmt_endpoint_url -
```

where `mgmt_endpoint_url` is the URL of the server where the cloud settings are stored. Include the hyphen at the end of the command to indicate that input follows on the next command line.

- Type the information for a single setting on one line, and press ENTER to submit it.  
Specify each setting on a single line and use the following format to specify the setting's name and value:

```
setting_name: setting_value
```

For example, the following command updates a single setting:

```
apic cloud-settings:update --server mgmt_endpoint_url -
refresh_expires_in: expiration time
Ctrl+D
```

- Type Ctrl+D to indicate that the data entry is complete, as shown in the example.
- Upload a file containing the settings.  
If you have multiple settings or lengthy values to add or modify, you might prefer to make all of the changes at once by creating or modifying the configuration file and then uploading it.
- Create a .yaml file with your settings.  
You can choose the file name but must use YAML format. The file can contain as few or as many settings as needed.

Settings use the following format; notice that the setting's name must be enclosed in quotation marks as shown in the example. The last setting in the file does not end with a comma.

```
"setting_name": setting_value,
```

- Upload the file with the following command:

```
apic cloud-settings:update --server mgmt_endpoint_url my_cloud_settings.yaml
```

For information on using input files with the toolkit CLI, see [./com.ibm.apic.toolkit.doc/rapic\\_cli\\_command\\_line\\_input.html](#).

4. When you are ready to log out of the management server, use the following command:

```
apic logout --server mgmt_endpoint_url
```

## Results

---

The settings are cached for a maximum of five minutes, so you might experience a delay before your changes take effect.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Change the name of your cloud

You can change the name of your cloud.

### About this task

---

You can change the name of your cloud using the Overview page.


One of the following roles is required to change the name of your cloud:

- Administrator
- Owner
- A custom role with the `cloud settings:manage`

### Procedure

---

Follow these steps to change the name of your cloud:

1. In the Cloud Manager, click  Settings.
2. Select Overview..
3. Click Edit to change the name of your cloud.
4. Enter a Title and the name is auto-generated. Enter a brief Summary and click Save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring an email server for notifications

Connect to an email server in your API Connect on-premises cloud. The email server sends registration invitations and other event-driven emails. You must configure the email server before you add any provider organizations or register a Portal service. Otherwise, the owner of the organization will not receive the email with the details that they require to access the API Manager.

### Before you begin

---

You must complete the tasks in the following section:

1. [Defining your topology](#).

### About this task

---

This task can be completed by users who are assigned one of the following roles:


- Cloud Owner
- Cloud Administrator
- Topology Administrator
- Custom role with `Settings:Manage` permission

The email server sends invitation emails with activation links and other system messages. More than one email server may be configured in Cloud Manager, but only one email server may be selected as the active server to send notifications at any one time. For instructions on how to select the active email server and to view a list of the email messages that are sent, see [Setting up notifications](#).

Note: If you do not select an email server for sending notification emails, it is not possible for a user to reset their password from either the Cloud Manager or API Manager login pages.

## Procedure

To configure one or more email servers to use for email notifications in Cloud Manager, complete the following steps:

1. In the Cloud Manager, click  Resources.
2. Choose Notifications.
3. Enter the values to configure the email server.

Field	Description
Title (required)	Enter a descriptive title for the availability zone. This title will display on the screen.
Name (required)	This field is auto-populated by the system and used as the internal field name.
Address (required)	IP address for the email server
Port (required)	Port that the email server runs on
Authenticate User	Enter a user name to use for authorization with the SMTP server. The user name is required if the SMTP server requires authentication.
Authenticate Password	Enter the password for the Authenticate User. This password is used for authorization with the SMTP server. The password is required if the SMTP server requires authentication.
TLS Client Profile (optional)	Select an optional TLS Client Profile that will be used to communicate with the email server if the Secure Connection option is selected.
Secure Connection	Select this option to specify that TLS security is used. The following conditions apply: <ul style="list-style-type: none"><li>• If this option is not selected then TLS security is not used even if the email server is configured to use the STARTTLS protocol.</li><li>• If this option is selected and a TLS Client Profile is selected, that profile is used and the STARTTLS protocol configuration on the email server is honored.</li><li>• If this option is selected and no TLS Client Profile is selected, the Default TLS client profile is used and the STARTTLS protocol configuration on the email server is honored.</li></ul>

4. You can send a test email to confirm that the email server is properly configured. Click Test email in the Test Connection panel to open a dialog window. Enter one or more valid email addresses in the Recipients field. Separate multiple recipients with commas. Click Send test email. Check that the test email was received by the recipients.
5. If the test email is successful, click Save to save the email server configuration. If the test is unsuccessful, check the configuration values and update as needed.  
Note: In order to send emails, you must select an email server as your active server in Settings > Notifications.

## Results

The email server is configured. It must be selected in Cloud Settings > Notifications as the active email server. See [Setting up notifications](#)

## What to do next

Select the email server to use for your cloud, enter the sender information, and customize the email templates. See [Setting up notifications](#).

Note: If you do not select an email server for sending notification emails, it is not possible for a user to reset their password from either the Cloud Manager or API Manager login pages.

## Related tasks

- [Setting up notifications](#)
- [Editing an email server](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting up notifications

To configure email notifications, configure an email server, sender information, and email templates.

### About this task

In Settings > Notifications, you select the email server you want to use as the active server to send notification emails in Resources > Notifications. An active email server is required to send email notifications. More than one email server may be configured in Cloud Manager, but only one email server may be selected to send notifications at any one time. The email server must be configured and selected for notifications before adding a Portal Service to your cloud.


You also configure the sender information to be included on all the emails and optionally, edit the standard text for the email templates.

One of the following roles is required to add roles to select the email server, enter the sender information, and edit the email templates:

- Administrator
- Owner
- A custom role with the `Cloud settings:Manage` permission

Note: If you do not select an email server for sending notification emails, it is not possible for a user to reset their password from either the Cloud Manager or API Manager login pages.

## Procedure

1. In the Cloud Manager, click  Settings.
2. In the Settings navigation list, click Notifications.
3. To configure the active email server:
  - a. Click Edit in the Sender & Email Server section.
  - b. Select an email server in the list; a check mark is displayed next to the currently selected server.
  - c. Click Save.
4. To delete an email server from the list, click the options menu next to the server name and select Delete.
5. To change the sender name and email address for notifications:
  - a. Click Edit in the Sender & Email Server section.
  - b. Fill in the Name and Email address that notifications will be sent from.
  - c. Click Save.
6. To preview and edit the notification email templates, see [Customizing email notification templates](#)

## What to do next

You can now [create a provider organization](#).

- [Customizing email notification templates](#)  
API Connect includes a set of email templates for invitations and other notification emails. You can preview the templates and customize the text if needed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Customizing email notification templates

API Connect includes a set of email templates for invitations and other notification emails. You can preview the templates and customize the text if needed.

## About this task

The API Connect emails are sent automatically when certain system events occur. The email templates are organized by scope, as explained in Table 1.


Table 1. Notification scopes

Scope	Events that trigger the notification
admin	Admin organization activities such as invitations, portal sign up, and password reset.
catalog	Activities related to Catalogs; for example: such as application life cycle events, invitations to catalogs, product approvals, and subscription approvals.
consumer	Activities to consumer applications and subscription requests in the Developer Portal, invitations to Consumer organizations, and password reset requests for Developer Portal accounts.
provider	Invitations
space	Invitations

One of the following roles is required to configure email notifications:

- Administrator
- Owner
- A custom role with the `Cloud settings:Manage` permission

## Procedure

1. In the Cloud Manager, click  Settings.
  2. In the Settings navigation list, click Notifications.  
On the Notifications page, templates display in a list, sorted by scope. When selecting a template to modify, be sure to select the version that applies to the appropriate scope.
  3. To preview the text for a template, select Preview from the actions menu next to the template name.
  4. To edit the text for the template, either click the template name, or select Edit Template in the action menu.  
The text includes variables; for example, `{{activationLink}}`.
- To obtain the complete list of variables that are available for a particular notification template, complete the following steps:
- a. Log in to the management server from the command line as a member of the cloud administration organization; for details, see [Logging in to a management server](#). You can use the same management server URL, user name, and password in the login command that you use to log in to the Cloud Manager user interface.
  - b. Enter the following command:

```
apic notification-templates:get template_name --server mgmt_endpoint_url --scope cloud --subcollection template_scope
--fields variables --output -
```

where *template\_name* is the name of the required notification template, as displayed in the Template column in the user interface, and *template\_scope* is the scope name displayed in the Scope column alongside that template.

For example:

```
apic notification-templates:get member-invitation --server https://myserver.com --scope cloud --subcollection catalog
--fields variables --output -
```

The variables that are available for the template are displayed; for example:

```
variables:
- org
- catalog
- activationLink
- expiresAt
```

The `--output -` parameter causes the command output to be written to the command line. You can specify `--output filepath` to have the output written to a .yaml file at the specified location, or omit it altogether to have a file written to the current folder.

5. Click Save when done.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring invitation timeouts

You can set the time period for when registration invitations for Cloud Settings access and the Admin organization expire.

### About this task

---


One of the following roles is required to change the name of your cloud:

- Administrator
- Owner
- A custom role with the `cloud settings:manage` permission

### Procedure

---

Follow these steps to set the invitation timeout:

1. In the Cloud Manager, click  Settings.
2. In the Settings navigation list, click Onboarding.
3. Click Edit next to the invitation timeout setting that you want to change:
  - Cloud settings invitation timeout
  - Admin organization invitation timeout
4. To specify a timeout value, select or type an integer value in the Number field and then choose a unit of time (Seconds, Minutes, or Hours) in the Unit field.
5. Click Save.

### Results

---

Invitations contain a link, which expires after the specified the timeout. If the user does not register before the expiration, you can send a new invitation.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the password-reset notification and timeout

Configure the time allowed for a user to reset a password, and customize the notification that the user receives.

### Before you begin

---

Configure an email server that can be used for mailing notifications to users. If you do not configure an email server, a link cannot be send to a user who wants to reset a password. For information on configuring the email server, see [Setting up notifications](#).


### About this task

---

When a user clicks **Forgot Password?** and provides an email address on the sign-in page, an email is generated with a link where the user can reset the password. When you deploy API Connect, the password-reset settings are configured with default values. You can optionally modify the time-out setting to set a new expiration for resets as well as the notification that is emailed to users.

## Procedure

---

1. Click  in the navigation pane.
2. In the Settings navigation list, click **Onboarding**.
3. Configure the password reset timeout:
  - a. In the Settings navigation list, click **Onboarding**.
  - b. Click **Edit** next to **Reset Password Time to Live**.
  - c. To specify a timeout value, select or type an integer value in the **Number** field and then choose a unit of time (**Seconds**, **Minutes**, or **Hours**) in the **Unit** field.
  - d. Click **Save**.
4. Customize the notification template for password resets:
  - a. In the Settings navigation list, click **Notifications**.
  - b. In the **Notifications Templates** list, select the appropriate version of the "password-reset" template by clicking the name.  
You can create different versions of the template for different user roles.
  - c. Customize the template as explained in [Customizing email notification templates](#).
  - d. Click **Save**.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Selecting user registries for Cloud Manager and API Manager

Select the user registries to authenticate users for Cloud Manager and API Manager.

### About this task

---

In **Settings > User registries**, you select one or more user registries to authenticate users of Cloud Manager and API Manager. The user credentials for all users who login to the applications must be stored in the selected registries.

One of the following roles is required to select the user registries for Cloud Manager and API Manager:


- Administrator
- Owner
- A custom role with the `cloud.settings:manage` permission

For details on how to control which user registries can be used by provider organizations for authenticating Developer Portal users, or for securing APIs, see [Setting visibility for a user registry](#).

## Procedure

---

To select the user registries for Cloud Manager and API Manager, follow these steps:

1. In the Cloud Manager, click  **Settings**.
2. In the Settings navigation list, click **User Registries**.
3. The current list of user registries is displayed for Cloud Manager and for API Manager. To add user registries, click **Edit** in either the Cloud Manager or API Manager panel.  
**Note:** The list of user registries is derived from the user registries that were previously configured as a Resource in **Resources > User Registries**. For more information, see [User registries overview](#).
4. Select the user registry by placing a check mark next to the name in the list. Remove a user registry by removing the check mark.
5. Click **Save**.

## Results

---

The user registry list is updated to reflect the selected user registries available for authenticating users.

## What to do next

---

To set a user registry as the default, click the options menu next to the registry name in the list, and select **Set as default**.

- [Setting a default user registry for Cloud Manager and API Manager](#)  
You can select a default user registry for Cloud Manager and for API Manager.

## Related tasks

---

- [Configuring an LDAP user registry in the Cloud Manager](#)
- [Configuring an Authentication URL user registry](#)



- [Configuring a Local User Registry](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Setting a default user registry for Cloud Manager and API Manager

You can select a default user registry for Cloud Manager and and for API Manager.

### About this task

---

In Settings > User registries, you can set a user registry as the default user registry.

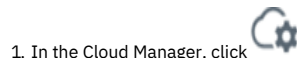
One of the following roles is required to select the user registries for Cloud Manager and API Manager:

- Administrator
- Owner
- A custom role with the `cloud_settings:manage` permission

### Procedure

---

To set a default user registry for Cloud Manager and API Manager, follow these steps:



1. In the Cloud Manager, click Settings.
2. In the Settings navigation list, click User Registries.
3. To set a user registry as the default, click the options menu next to the registry name in the list, and select Set as default.

Note: The list of user registries is derived from the user registries that were previously configured as resources in Resources > User Registries. For more information, see [User registries overview](#)

### Results

---

The user registry list is updated to reflect the default user registry.

### Related tasks

---

- [Configuring an LDAP user registry in the Cloud Manager](#)
- [Configuring an Authentication URL user registry](#)
- [Configuring a Local User Registry](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring timeouts for access tokens and refresh tokens

Configure timeout values for the access tokens and refresh tokens for user registries.

IBM® API Connect supports refresh tokens for user registries that authenticate users of the Cloud Manager, the API Manager, and the Developer Portal. Refresh tokens, when enabled, are generated every time an access token is created. Refresh tokens can be used to obtain a new access token without having to prompt a user to re-login. If you enable the use of refresh tokens, then when the token expires it is refreshed automatically in the background so that the user can continue working without interruption. When the user logs out, the refresh token is revoked and the application can no longer use it.

Note: If users log in with non-OIDC user registries, the use of refresh tokens is supported in API Manager, API Designer, the Developer Portal, and the CLI Toolkit. When the refresh token expires, the user is always redirected to the login page.

If users log in with OIDC user registries:

- The use of refresh tokens is supported only in the Developer Portal.
- If the OIDC providers do not support refresh tokens, API Connect will not issue a `refresh_token`, regardless of the setting of `refresh_token_enabled` in the cloud settings.
- For more information about OIDC configuration, see [Configuring an OIDC user registry](#).

Use the following properties to specify refresh tokens for cloud settings:

- `refresh_expires_in`  
Integer. Represents the expiration time, in seconds, for refresh tokens issued by API Connect. Must be greater than `access_token_expires_in`.
- `refresh_token_enabled`  
Boolean. Disabled by default. To enable, set to `true`. When enabled, generates a `refresh_token` field in the response to the `/token` API call.

The following extract from a cloud settings file shows example settings for these properties:

```
"type": "cloud_setting",
"api_version": "2.0.0",
"name": "cloud-setting",
"access_token_expires_in": 28800,
.
.
.
"refresh_expires_in": 57600,
"refresh_token_enabled": false,
.
.
.
```

Use the toolkit CLI (`apic`) to update the cloud settings with new values for the properties. You can either specify values as command line arguments, or enter them manually in a configuration file as explained in [Using the CLI to modify cloud settings](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Viewing platform and UI endpoints

A view-only list of platform and UI endpoints is provided.

### About this task

---

You can view the list of endpoints configured during the installation process. Included in this view-only list are the following endpoints:

- API Manager URL - the URL for opening the API Manager UI
- Platform REST API endpoint for the Admin and Provider REST APIs - this endpoint handles the REST APIs for Admin and Provider functions.
- Platform REST API endpoint for consumer REST APIs - this endpoint handles the REST APIs for consumer (Portal) functions.


One of the following roles is required to view endpoints:

- Administrator
- Owner
- A custom role with the `cloud settings:manage` or `cloud-settings: view` permissions

### Procedure

---

To view the endpoints, follow these steps:

1. In the Cloud Manager, click  Settings.
2. Select Endpoints.
3. View the list of endpoints.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring the default gateway services for Catalogs

Configure the list of default gateway services that are available to new Catalogs.

### About this task

---

One of the following roles is required to configure Catalog Defaults:

- Administrator
- Owner
- A custom role with the `cloud settings:manage` permission

You publish APIs by adding them to a Product and then publishing the Product to a Catalog. To be able to publish Products to a Catalog, the Catalog must be assigned at least one gateway service so that the APIs in the Product are available to be called at a gateway service endpoint. You can assign more than one gateway service to a Catalog, and they can be of mixed types, DataPower® API Gateway and DataPower Gateway (v5 compatible).


A Product is gateway type specific, either DataPower API Gateway or DataPower Gateway (v5 compatible). By default, when you publish a Product to a Catalog, it is published to all the assigned gateway services of the same gateway type as the Product, but you can choose to publish the Product only to selected gateway services, of that type, at publish time; see [Publishing a draft Product](#) for more information. The APIs in the Product will be available to be called at all the specified gateway service endpoints.

The procedure described in this topic specifies the gateway services that are assigned by default to newly created Catalogs, but you can change the assigned gateway services for a specific Catalog; see [Creating and configuring Catalogs](#).

## Procedure

---

To configure the gateway services that are assigned to new Catalogs by default, complete the following steps:

- 
1. In the Cloud Manager, click [Settings](#).
  2. In the Settings navigation list, click [Catalog Defaults](#).
  3. Click [Edit](#) to change the list of default gateway services.
  4. Select the required gateway services, then click [Save](#).

## Results

---

When a new Catalog is created in a provider organization, the default gateway services are automatically assigned to it. Any changes you make to the list of default services affects only the Catalogs that are created after you save your changes. Existing Catalogs are not affected.

## Related information

---

- [Working with Catalogs](#)
- [API Connect gateway types](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Administering provider organizations

Manage the provider organizations who have accounts to publish APIs in your API Connect cloud.

### About this task

---

Perform these tasks to manage your provider organizations:

- [Creating a provider organization](#)  
For developers to publish APIs, they must be a member of a provider organization. The first step is to create a provider organization and specify an organization owner. Then members can be added to the provider organization and can start publishing APIs.
- [Editing a provider organization](#)  
You can edit the title of a provider organization.
- [Deleting a provider organization](#)  
You can delete a provider organization. All Catalogs and Spaces are also removed.
- [Viewing current provider organizations](#)  
View the list of current provider organizations for your cloud, including organization title, owner's name and email address, and status. From the list screen, you can add new provider organizations and owners, edit and delete provider organizations, send messages, and change owners.

### Related tasks

---

- [Creating a provider organization](#)
- [Editing a provider organization](#)
- [Deleting a provider organization](#)
- [Viewing current provider organizations](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a provider organization

For developers to publish APIs, they must be a member of a provider organization. The first step is to create a provider organization and specify an organization owner. Then members can be added to the provider organization and can start publishing APIs.

### Before you begin

---

This task can be completed by users who are assigned one of the following roles:

- Cloud Owner
- Cloud Administrator

- Organization Manager
- Custom role with the `Provider-org:Manage` permission

You must also complete the following tasks before beginning:

- [Configuring an email server for notifications](#)
- [Setting up notifications](#)

## About this task


Note that an email server must be configured and an active email server must be selected before a provider organization account can be created.

There are two options for creating a provider organization account, as follows:

- Option 1: Create a provider organization and send an email invitation to a new user to register and become the organization owner. Choose the Add > Invite organization owner option.
- Option 2: Create a provider organization and specify the user who will be the organization owner. Choose the Add > Create organization option.

## Procedure


- **Option 1:** To create a provider organization and invite a new user to register as the owner, complete the following steps:

1. In the Cloud Manager, click  Provider Organizations.
2. Click Add > Invite organization owner
3. Enter a valid email address and click Invite. An email containing an activation link is sent to the email address inviting the person to register. The link opens the registration form. Alternately, you can copy and paste the activation link into an email; click Get Activation Link and click Copy in the pop up window. Note that the email server requires the Authenticate User and Authenticate Password values in order to send invitations. See [Configuring an email server for notifications](#).

The new provider organization is placed in Pending status, and is changed to Enabled status after the owner completes the registration form to activate their account. The owner can then log in to the API Manager user interface with the API Connect user name that they specified during account activation.

Note: If you use this option with a Local User Registry, and the user already exists in the registry, the user must re-activate their account by using the Sign In option, **not** by completing the registration form and using the Sign Up option; attempting to re-register will fail. This stipulation includes users that were previously provider organization owners but whose organization was deleted.

- **Option 2:** To create a provider organization and specify the user who will be the organization owner, complete the following steps:

1. In the Cloud Manager, click  Provider Organizations.
2. Click Add > Create organization.
3. In the Title field, enter a title for the provider organization. If desired, the title can be edited after the provider organization is created.
4. The internal Name is automatically generated for you by the system. The Name field contains the string that is included in the organization segment of the URL for API calls. For more information, see [Calling an API](#).  
Tip: The Name field is auto-generated to ensure its validity in the URL.  
The value in the Name field is also used to identify the provider organization in developer toolkit CLI commands. To view the CLI commands to manage provider organizations, see [apic orgs](#).

5. Select the user registry for the provider organization owner.

The remaining procedure varies according to the type of the selected user registry, as follows:

- Local User Registry
  - Select whether the user is an Existing user or a New User.
    - For an existing user, complete the following steps:
      - Enter the name of an existing API Connect user that has previously been invited to register and has activated their account.
      - Click Create. The provider organization is created and immediately enabled. The specified owner can log in to the API Manager user interface. If the new provider organization is the user's only provider organization, the API Manager user interface opens in that provider organization; if the user is a member of more than one provider organization, they can then select the new provider organization from the Organization list.
    - For a new user, complete the following steps:
      - Enter a unique user name for the new user.
      - Supply an email address, name details, and a password.
      - Click Create. The user account is created, and the provider organization is created and immediately enabled. The specified owner can log in to the API Manager user interface with the specified user name and password; the user interface open in the new provider organization.
- LDAP
  - Enter the name of a user that exists in the selected user registry.
  - Click Create. The provider organization is created and immediately enabled. The specified owner can log in to the API Manager user interface with their LDAP user name. If the new provider organization is the user's only provider organization, the API Manager user interface opens in that provider organization; if the user is a member of more than one provider organization, they can then select the new provider organization from the Organization list.
- Authentication URL and OIDC
  - Enter the name of an existing API Connect user that has previously been invited to register and has activated their account. If the required organization owner has not previously been invited and activated their account, you must use the invitation mechanism described in [Option 1](#).
  - Click Create. The provider organization is created and immediately enabled. The specified owner can log in to the API Manager user interface. If the new provider organization is the user's only provider organization, the API Manager user interface opens in that provider organization; if the user is a member of more than one provider organization, they can then select the new provider organization from the Organization list.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Editing a provider organization

You can edit the title of a provider organization.

### Before you begin

---


This task can be completed by users who are assigned the following roles:

- Cloud Owner
- Cloud Administrator
- Organization Manager
- Custom role with the `Provider-org:Manage` permission

### Procedure

---

To update a provider organization, complete the following tasks:

1. In the Cloud Manager, click  Provider Organizations.
2. Navigate to the provider organization that you want to update, select Edit from the Actions menu.
3. Enter a new Title for the organization.
4. Click Save.

### Results

---

The provider organization title is updated.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting a provider organization

You can delete a provider organization. All Catalogs and Spaces are also removed.

### Before you begin

---

This task can be completed by users who are assigned the following roles:

- Cloud Owner
- Cloud Administrator
- Organization Manager
- Custom role with the `Provider-org:Manage` permission

### About this task

---


The user account for the owner of the deleted provider organization remains in the associated API Connect user registry. The user can still log in to the API Manager user interface; if they are a member of one or more other provider organizations then those organizations are accessible, otherwise only information links are available, with no access to view or manage any resources.

The user can be added again as a provider organization owner at a later time; see [Creating a provider organization](#).

### Procedure

---

To delete a provider organization, complete the following tasks:

1. In the Cloud Manager, click  Provider Organizations.
2. Navigate to the provider organization that you want to delete in the list and select Delete from the Actions menu.
3. Confirm that you want to delete the provider organization.  
The provider organization is deleted and removed from the Provider Organizations list.

### Results

---

The provider organization, and its Catalogs and Spaces, are deleted.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Viewing current provider organizations

View the list of current provider organizations for your cloud, including organization title, owner's name and email address, and status. From the list screen, you can add new provider organizations and owners, edit and delete provider organizations, send messages, and change owners.

---

### Before you begin

You must also complete the following tasks before beginning:

- [Configuring an email server for notifications](#)
- [Setting up notifications](#)
- [Creating a provider organization](#)

This task can be completed by users who are assigned one of the following roles:

- All roles
- Custom role with the `Provider-org:Manage` or the `Provider-org:View` permissions


---

### About this task

All roles can view the list of provider organizations. Use the provider organization list screen to manage your provider organizations.

---

### Procedure

1. In the Cloud Manager, click  Provider Organizations.
2. View the list of provider organizations and their status. The statuses are as follows:
  - **Pending:** A provider organization is in `Pending` status until the invited owner successfully completes the registration form.
  - **Active:** A provider organization is in `Active` status if the owner is an existing user with an account in a user registry.
3. The Actions menu performs different functions depending on the organization's status. Use the Actions menu to edit or delete an Active organization. For Pending organizations, use the Actions menu to delete the organization and to resend the invitation to the owner.
  - For an Enabled organization, you can Edit or Delete the organization.
  - For a Pending organization, you can Resend the Invitation, or Delete the request.
4. Add new provider organizations using the Add menu. See [Creating a provider organization](#)

---

### Related tasks

- [Creating a provider organization](#)
- [Editing a provider organization](#)
- [Deleting a provider organization](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Administering members and roles

Cloud Administrators can add members and assign them roles to enable them to work in Cloud Manager. The Cloud and Topology Administrators can also create roles and role defaults. You can also delete users to prevent them from accessing Cloud Manager.

---

### About this task

To manage roles, add members, and assign roles to member, perform the following tasks:

- [Creating roles in the admin organization](#)  
Cloud Manager ships with a set of pre-configured roles. As the Cloud Manager administrator, you can create custom roles, add permissions, and assign the roles to members of the Admin organization.
- [Editing roles in the admin organization](#)  
As the Cloud Manager administrator, you can edit the roles in the Admin Organization. Both pre-configured roles and custom roles may be edited, except for the Owner, Viewer, and Member roles. The Owner, Viewer, and Member roles may not be edited or deleted.
- [Deleting roles in the admin organization](#)  
Roles can be deleted (except for the Owner, Viewer, and Member roles) from the Admin organization. If there are members assigned to a role that has been deleted, they will be assigned the Viewer role.
- [Viewing members and roles](#)  
Use the Members page to view the current members of the Admin organization and manage the role assignments.
- [Adding members to the admin organization](#)  
As the Cloud Manager administrator, you can add and remove members of the Admin organization and assign roles to them. Once removed, a user can no longer access the Cloud Manager.
- [Assigning roles to members](#)  
As the Cloud Manager administrator, you manage the roles that are assigned to members. A member's role determines the tasks they are authorized to perform in Cloud Manager.

- [Deleting a member](#)

As the Cloud Manager administrator, you can delete members of the API Connect cloud administration organization. Once deleted, the user and associated roles are removed, however the user's account still remains in API Connect.

- [Role Defaults overview](#)

Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating roles in the admin organization

Cloud Manager ships with a set of pre-configured roles. As the Cloud Manager administrator, you can create custom roles, add permissions, and assign the roles to members of the Admin organization.

### About this task

---

If you are assigned a role that has the `Settings: manage` permission, you can create custom roles for the Admin organization. The Admin organization roles apply only to Cloud Manager users.


One of the following roles is required to add custom roles:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: manage` permissions

### Procedure

---

Perform the following steps to add custom roles to the Admin Organization:

1. In the Cloud Manager, click  Settings.
2. Select Roles.
3. Click Add to add a new role.
4. Enter the Title for the role. The Name will be auto-generated.
5. Use the check boxes to select the permissions for the role. For a description of the permissions and the actions they enable, see [User roles and permissions in the Cloud Manager UI](#).
6. Save the custom role.

### Results

---

The custom role appears in the list of roles and can be assigned to members.

### What to do next

---

Assign the role to a member in the Members list. See [Assigning roles to members](#)

### Related tasks

---

- [Editing roles in the admin organization](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing roles in the admin organization

As the Cloud Manager administrator, you can edit the roles in the Admin Organization. Both pre-configured roles and custom roles may be edited, except for the Owner, Viewer, and Member roles. The Owner, Viewer, and Member roles may not be edited or deleted.

### About this task

---

You can customize the per-configured roles (except for the Owner, Viewer, and Member roles) to reflect your business needs.

If you are assigned a role that has the `Settings: manage` permission, you can edit the roles for the Admin organization. The Admin organization roles apply only to Cloud Manager users.

One of the following roles is required to edit roles:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: manage permissions`


Following are the edits you can perform for a role:

- Change the title
- Add or delete permissions

## Procedure

---

Perform the following steps to edit roles in the Admin organization:

1. In the Cloud Manager, click  Settings.
2. Select Roles.
3. Locate the Role you want to edit, and select Edit from the overflow menu.
4. Enter the desired Title for the role. The Name will be auto-generated.
5. Use the check boxes to select the permissions for the role. For a description of the permissions and the actions they enable, see [User roles and permissions in the Cloud Manager UI](#).
6. Save the updated role.

## Results

---

The Role will be updated with a new name and/or permissions. An updated role will affect the members who are assigned the role. If you change the name of a role, you will have to assign it to the members. If you change the permissions for a role, the way that members use the product may change.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deleting roles in the admin organization

---

Roles can be deleted (except for the Owner, Viewer, and Member roles) from the Admin organization. If there are members assigned to a role that has been deleted, they will be assigned the Viewer role.

## About this task

---

All custom roles may be deleted. Pre-configured roles can also be deleted, except for the Owner, Viewer, and Member roles. Owner, viewer, and Member role cannot be deleted or edited. Members who are assigned a deleted role are given the most limited role, Viewer, with View-only privileges. These members will need to be assigned a new role if they require any other privileges.

If you are assigned a role that has the `Settings: manage` permission, you can delete roles for the Admin organization. The Admin organization roles apply only to Cloud Manager users.


One of the following roles is required to delete custom roles:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: manage permissions`

## Procedure

---

Perform the following steps to delete roles in the Admin organization:

1. In the Cloud Manager, click  Settings.
2. Select Roles.
3. Locate the Role you want to delete, and select Delete from the overflow menu.
4. Confirm that you want to proceed with deleting the role.

## Results

---

The Role will be deleted. Members who had been assigned the role are given the Viewer role.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---


## Viewing members and roles

Use the Members page to view the current members of the Admin organization and manage the role assignments.

### Procedure

---

To view the members of the Admin organization and their roles in Cloud Manager, follow these steps:

1. In the Cloud Manager, click  Members.
  2. From the Members list, you can scroll through the list of users to see their current roles and their status.
    - **Pending** - The invitation with the activation link has been sent to the user, but they have not yet completed the registration form.
    - **Enabled** - The user has completed the registration form and is now able to login and access the features associated with their role.
- Note: If your role has the proper permissions, you can assign roles, add, and delete members from this list.

### What to do next

---

- To add members, see [Adding members to the admin organization](#)
- To change the roles assigned to members see [Assigning roles to members](#)
- To delete a member, see [Deleting a member](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding members to the admin organization

As the Cloud Manager administrator, you can add and remove members of the Admin organization and assign roles to them. Once removed, a user can no longer access the Cloud Manager.

### About this task

---

Membership in the Admin organization is required to use Cloud Manager to view and manage the topology of the API Connect cloud, to set up and manage provider organizations, and to manage other features of the cloud. There are two ways to add a member:

- Invite member - send an email invitation to a new user to register as a member of the Admin organization.
- Add member - specify the user and add them as a member of the Admin organization immediately.


One of the following roles is required to add members:

- Administrator
- Owner

### Procedure

---


- To send an email invitation to a new user to register as a member of the Admin organization, complete the following steps:

1. In the Cloud Manager, click  Members.  
The list of current members for the Admin organization is displayed. You can view the owner of the Admin Organization by expanding View Owners.
2. Click Add > Invite member.
3. Enter a valid email address.
4. Select the roles that you want to assign to the user.
5. Click Invite. An email containing an activation link is sent to the email address inviting the person to register. The link opens the registration form. Alternately, you can copy and paste the activation link into an email; click Get Activation Link and click Copy in the pop up window.  
Note that the email server requires the Authenticate User and Authenticate Password values in order to send invitations. See [Configuring an email server for notifications](#).

The user is placed in Pending status, and is changed to Enabled status after they complete the registration form to activate their account. They can then log in to the Cloud Manager user interface with the API Connect user name that they specified during account activation.

Note: If you use this option with a Local User Registry, and the user already exists in the registry, the user must re-activate their account by using the Sign In option, **not** by completing the registration form and using the Sign Up option; attempting to re-register will fail. This stipulation includes users that were previously members but were deleted from the admin organization.

- To specify the user and add them as a member of the Admin organization immediately, complete the following steps:

1. In the Cloud Manager, click  Members.  
The list of current members for the Admin organization is displayed. You can view the owner of the Admin Organization by expanding View Owners.
2. Click Add > Add member.
3. Select the required user registry  
The remaining procedure varies according to the type of the selected user registry, as follows:
  - Local User Registry
    - Select whether the user is an Existing user or a New User.

- For an existing user, complete the following steps:
  - Enter the name of an existing API Connect user that has previously been invited to register and has activated their account.
  - Select the roles that you want to assign to the user, then click Create. The user is added and their membership is immediately enabled. The specified user can log in to the Cloud Manager user interface.
- For a new user, complete the following steps:
  - Enter a unique user name for the new user.
  - Supply an email address, name details, and a password.
  - Select the roles that you want to assign to the user, then click Create. The user account is created, and the user membership is immediately enabled. The specified user can log in to the Cloud Manager user interface.
- LDAP
  - Enter the name of a user that exists in the selected user registry.
  - Select the roles that you want to assign to the user, then click Create. The user is added, and the user membership is immediately enabled. The specified user can log in to the Cloud Manager user interface with their LDAP user name.
- Authentication URL and OIDC
  - Enter the name of an existing API Connect user that has previously been invited to register and has activated their account.
  - Select the roles that you want to assign to the user, then click Create. The user is added, and the user membership is immediately enabled. The specified user can log in to the Cloud Manager user interface.

## What to do next

The new member can log in and work in Cloud Manager. The member's authorization is defined by the roles assigned to them.

For instructions on assigning roles, see [Assigning roles to members](#). The list of roles and permissions can be viewed at [User roles and permissions in the Cloud Manager UI](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Assigning roles to members

As the Cloud Manager administrator, you manage the roles that are assigned to members. A member's role determines the tasks they are authorized to perform in Cloud Manager.

### About this task

Members in the Admin organization use Cloud Manager to view and manage the topology of the Cloud, to set up and manage provider organizations, and to work with other features of the API Connect Cloud. The member's role determines what tasks they are authorized to perform.


When assigning roles, you need to be aware that all members always have a basic set of permissions, which are contained in the **Member** role. The Member role is automatically granted to all members and cannot be removed. For this reason, you do not explicitly assign the Member role. You can grant additional permissions by selecting one or more roles. If additional roles are not assigned, then the user will have only the basic Member role.

One of the following roles is required to assign roles to members:

- Administrator
- Owner

## Procedure

Perform the following steps to assign one or more roles to a member:

1. In the Cloud Manager, click  Members.
2. The list of current members for the Admin organization is displayed.
3. To change the roles assigned to a member, select the roles by marking the check boxes next to their name. The roles will be updated instantly. The list of roles and permissions can be viewed at [User roles and permissions in the Cloud Manager UI](#)

## Results

The member will be assigned the permissions for the role(s).

## What to do next

The new member can work in Cloud Manager. The member's authorization is defined by the roles assigned to them.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting a member

As the Cloud Manager administrator, you can delete members of the API Connect cloud administration organization. Once deleted, the user and associated roles are removed, however the user's account still remains in API Connect.

## About this task

---

One of the following roles is required to delete a member:

- Administrator
- Owner


Note: The user account of the deleted user remains in the associated API Connect user registry. The user can still log in to the Cloud Manager user interface but only information links are available; there is no access to view or manage any resources.

The user can be added again as a member at a later time; see [Adding members to the admin organization](#).

## Procedure

---

To delete a member of the cloud administration organization, complete the following steps:

1. In the Cloud Manager, click  Members.
2. Choose Delete from the overflow menu that is adjacent to the member you want to delete.
3. Confirm the deletion.

## Results

---

The member is deleted and removed from the Members list in Cloud Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Role Defaults overview

---

Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.

## Role Defaults concepts

---

Role Defaults determine the base roles applied to either Provider or Consumer Organizations. The Role Defaults for the Admin organization are not configurable. Role Defaults help to standardize base roles for all organizations. Roles and permissions may be added or modified in the Role Defaults to customize the base roles as needed. Role Defaults are a "point-in-time" configuration, which effects only new organizations. You can add, delete, or modify roles (except for the Owner, Member, and Viewer roles) in the organization to customize the permissions for each organization.

Rules for editing Role Defaults:

- Edits to Role Defaults are not applied retroactively to existing organizations. Edits apply only to new organizations created after the changes are made.
- Role Defaults apply to all new organizations; roles can then be edited at the organization, catalog, and space level.
- Edits made to roles in an organization do not affect the Role Defaults. Role Defaults are the "parent" and organization roles are the "child". The child inherits from the parent, but changes do not filter up from the organization to the Role Default.
- [Managing role defaults for provider organizations](#)  
Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.
- [Managing role defaults for consumer organizations](#)  
Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.

## Related tasks

---

- [Managing role defaults for provider organizations](#)
- [Managing role defaults for consumer organizations](#)

## Related information

---

- [Creating custom roles](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing role defaults for provider organizations

Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.

### About this task

---


If you are assigned a role that has permissions to manage Cloud Settings, you can create manage the Role Defaults for the provider organizations. One of the following roles is required to add roles to the role defaults:

- Administrator
- Topology Administrator
- Owner

Perform the following steps to add or edit roles to the Roles Defaults for provider organizations:

### Procedure

---

1. In the Cloud Manager, click  Settings.
2. Select Role Defaults.
3. Click Add to add a new role.
  - a. To edit a role, select Edit from the overflow menu for the role.
  - b. To delete a role from the Role Default, select Delete from overflow menu and confirm the deletion. Owner, Member, and Viewer roles cannot be deleted.  
Note: Role deletions apply to new members added after the deletion. If a current member is assigned the role, they will continue to have the permissions connected with the role. You need to manage the deletions for each member in the Members screen.
4. Enter the Title for the role. The Name will be auto-generated.
5. Use the check boxes to select the permissions for the role. For a description of the permissions and the actions they enable, see [User roles and permissions in the Cloud Manager UI](#).
6. Save the role.

### Results

---

The role appears in the list of roles for the provider organization Role Defaults. It will be included in the roles that can be assigned to members for all new provider organizations. The roles can be edited for individual organizations.

### Related concepts

---

- [Role Defaults overview](#)

### Related tasks

---

- [Managing role defaults for consumer organizations](#)
- [Assigning roles to members](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing role defaults for consumer organizations

Role Defaults are role templates that determine the default roles created in new Provider and Consumer organizations. Cloud Manager ships with a set of pre-configured Role Defaults. As the Cloud Manager administrator, you can edit the pre-configured Role Defaults.

### About this task

---


If you are assigned a role that has permissions to manage Cloud Settings, you can create and manage the Role Defaults for consumer organizations. One of the following roles is required to add roles to the role defaults:

- Administrator
- Topology Administrator
- Owner

Perform the following steps to add or edit roles to the roles defaults for consumer organization:

### Procedure

---

1. In the Cloud Manager, click  Settings.

2. Select Role Defaults.
3. Click Add to add a new role.
  - a. To edit a role, select Edit from the overflow menu for the role.
  - b. To delete a role from the Role Default, select Delete from overflow menu and confirm the deletion. Owner, Member, and Viewer roles cannot be deleted.  
Note: Role deletions apply to new members added after the deletion. If a current member is assigned the role, they will continue to have the permissions connected with the role. You need to manage the deletions for each member in the Members screen.
4. Enter the Title for the role. The Name will be auto-generated.
5. Use the check boxes to select the permissions for the role. For a description of the permissions and the actions they enable, see [User roles and permissions in the Cloud Manager UI](#).
6. Save the role.

---

## Results

The role appears in the list of roles for the consumer organization Role Defaults. It will be included in the roles that can be assigned to members for new consumer organizations. The roles can be edited for catalogs, but permissions inherited from the Role Default cannot be deleted. Permissions can be added at the catalog level, but not deleted.

---

## Related concepts

- [Role Defaults overview](#)

---

## Related tasks

- [Managing role defaults for provider organizations](#)
- [Assigning roles to members](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Monitoring the cloud

API Connect generates events to monitor the status of your cloud.

---

## About this task

You can monitor your API Connect cloud and configure destination targets for your analytics data by using the following tasks:

- [Counting total API calls for your Analytics service](#)  
Count the API calls for the IBM API Connect Analytics subsystem and return the total categorized by response status.
- [Customizing Analytics data with filters](#)  
Define filters to refine the Analytics data that you want to store. Using filters, you can add fields, remove fields, or modify field contents.
- [Configuring data retention and index rollover time periods](#)  
You can set a time period for data retention and index rollover using the Advanced Analytics Configuration option for an analytics service.
- [Configuring analytics offload for API Connect](#)  
The event data that is generated and collected in your API Connect on-premises cloud can be forwarded to different destination targets for display and analysis. Destination targets include the API Connect user interfaces, and third-party systems that are external to API Connect.
- [Configuring the audit log to track API calls](#)  
Configure auditing in IBM API Connect to monitor user operations such as who published which product and when.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Counting total API calls for your Analytics service

Count the API calls for the IBM® API Connect Analytics subsystem and return the total categorized by response status.

---

## Before you begin

This procedure requires you to deploy the Analytics subsystem and store data in IBM API Connect. If your organization did not deploy the Analytics subsystem, then you cannot use this process to track usage. If you deployed Analytics but offload all of your data and do not store it in IBM API Connect, then you must capture usage information from the third-party application where you offload data.

---

## About this task

If your IBM API Connect bases costs on usage, you can use the following procedure to track this information so that you can accurately report it.

## Procedure

1. Find an analytics-storage pod (for example, coordinating, data, master, or basic) by running the following command.

```
kubect1 get pods
```

2. Build a query for your request to the analytics datastore.

- a. Set the `size` to 0 to disable the return of raw data.

The following example shows how to specify the `size`.

```
{
  "size": 0
}
```

Tip: If you want to return some raw data to verify that the range query in the next step works, set the `size` to 1 instead.

- b. Add a `range` query for date filtering.

The following example shows how to specify the time range in your query:

```
{
  "size": 0,
  "query": {
    "range": {
      "datetime": {
        "gte": "YYYY-MM-DD",
        "lt": "YYYY-MM-DD"
      }
    }
  }
}
```

This setting filters the query by specifying a time range that begins on or after (is greater than or equal to) the `gte` date, and ends before (is less than) the `lt`.

Note:

When specifying the time range, note the following considerations about the value that you specify for the date:

- Format: The value must use the ISO-8601 format (`YYYY-MM-DDThh:mm:ss.yyyZ`) or a shortened version of that format (for example, `YYYY-MM-DD`).
- Timezone: The value must follow UTC time (Coordinated Universal Time) because the query will be executed by the analytics-storage pod (which runs on UTC time), so it's important to allow for any time difference when you run a query.

- c. Build the aggregation.

The aggregation code uses regular expressions to match the patterns for response codes (for example all of the responses with a code that begins with "1").

The query counts the documents within the specified time range that match each set of response codes, and then calculates the total number of documents.

The following example shows the complete query. The `aggs` section contains the code to count the total number of API calls.

```
{
  "size": 0,
  "query": {
    "range": {
      "datetime": {
        "gte": "YYYY-MM-DD",
        "lt": "YYYY-MM-DD"
      }
    }
  },
  "aggs": {
    "status_codes": {
      "filters": {
        "filters": {
          "1xx": {
            "regexp": {
              "status_code": "1.*"
            }
          },
          "2xx": {
            "regexp": {
              "status_code": "2.*"
            }
          },
          "3xx": {
            "regexp": {
              "status_code": "3.*"
            }
          },
          "4xx": {
            "regexp": {
              "status_code": "4.*"
            }
          },
          "5xx": {
            "regexp": {
              "status_code": "5.*"
            }
          }
        }
      }
    }
  }
}
```

3. Submit a request with your query to the analytics-storage pod that you located in step 1.

Run the following command, where:

- `analytics-storage-pod` represents the name of the analytics-storage pod that you identified in step 1.
- `analytics_query` is the content of the query, as shown in step 2c.

```
kubectl exec -it analytics-storage-pod -- \
curl_es /apic-api-r/_search?pretty \
-d 'analytics_query' \
-H 'Content-Type: application/json'
```

#### 4. Review the response.

Most of the information in the response can be ignored because it doesn't apply to your use case. Locate the `aggregations.status_codes.buckets` field, which contains the individual fields that represent the status codes.

The status code fields are labeled as `1xx`, `2xx`, `3xx`, `4xx`, and `5xx`. Each of the status code fields has an object as its value with a field called `doc_count`.

In each status code field, locate the `doc_count` field and note its value, which represents the total number of API calls for the current status code within the specified time range.

The following example shows a sample response with an aggregated total of 100,000 API calls.

```
{
  "took" : 19,
  "timed_out" : false,
  "_shards" : {
    "total" : 2500,
    "successful" : 2500,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 100000,
    "max_score" : 0.0,
    "hits" : [ ]
  },
  "aggregations" : {
    "status_codes" : {
      "buckets" : {
        "1xx" : {
          "doc_count" : 10000
        },
        "2xx" : {
          "doc_count" : 60000
        },
        "3xx" : {
          "doc_count" : 5000
        },
        "4xx" : {
          "doc_count" : 15000
        },
        "5xx" : {
          "doc_count" : 10000
        }
      }
    }
  }
}
```

## What to do next

To ensure you correctly track the API calls, you should run the query on a daily basis and store the accumulated daily totals where you can make sure they are backed up. You can create a script based on the query and schedule it to run daily. For more information about creating the script, see [Creating a script to count API calls](#).

- [Creating a script to count API calls](#)  
Optionally create a script containing a query that counts the API calls for the IBM API Connect Analytics subsystem.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a script to count API calls

Optionally create a script containing a query that counts the API calls for the IBM® API Connect Analytics subsystem.

## Before you begin

Create a query to count the total number of API calls in the Analytics subsystem as explained in [Counting total API calls for your Analytics service](#).

## About this task

If you need to count your API calls to track the totals for your contract, then you must run the query to count API calls on a periodic basis. You can automate the process by creating a script that contains the query. To ensure you correctly track the API calls, schedule the script to run on a daily basis and store the accumulated daily totals where you can make sure they are backed up.

The example script includes the following changes to the original query. The changes make it easier to run the script repeatedly without editing it each time.

- Accept the namespace and release name of the analytics service as input.  
If you only deployed one release in the namespace, then you can leave the release name blank. For appliance deployments, use `default` as the value of the namespace, and leave the release name blank.
- Identify the analytics-storage pod automatically and saves it in the `POD` variable to be used by your query.
- Change the query's time range variables from ISO format to date math syntax so that you can run the script again without editing the time range.  
You should run the script at the same time every day to ensure consistent counts. Using date math, you can use variables to calculate yesterday's date and today's date. Then, your script can use those dates to retrieve the total number of calls for the 24 hours that elapsed between the beginning of yesterday and the beginning of today.
  - Set the start date (`gte`) to the value `now-1d/d`, which represents the beginning of yesterday.
  - Set the end date (`lt`) to `now/d`, which represents the beginning of today.

Note: Remember, the values for the time range variables are calculated from the current system time of the analytics-storage pod, which follows UTC time (Coordinated Universal Time). It's important to allow for any time difference when you run a query because your timing might cause an inadvertent date change for the results.

For example, if today is January 2 in EST and your current time is 5:00 AM (2019-01-02T05:00:00), then the system time on the analytics-storage pod (in UTC) is 10:00 AM on January 2 (2019-01-02T10:00:00). The date math calculation for yesterday (`now-1d/d`) is equal to 2019-01-01T00:00:00, and the date math calculation for today (`now/d`) is equal to 2019-01-02T00:00:00. The query captures the API call counts for "yesterday", which is the 24 hours of the date 2019-01-01T00:00:00. But if your current time in EST is 8:00 PM on January 2, the time on the system time on the analytics-storage pod is 1:00 AM on January 3. "Yesterday" is now January 2 and "today" is now January 3, so the query returns the call count for January 2 instead of January 1.

## Procedure

1. Create the script.

The following example shows the completed script. In the script, the `kubectl exec` statement must be on a single line.

```
#!/bin/bash

NAMESPACE=$1
RELEASE_NAME=$2

POD=$(kubectl get po -n $NAMESPACE -o custom-columns=NAME:metadata.name | grep -e "storage-master-0" | head -n 1 |
grep -e "$RELEASE_NAME-analytics-storage" | \

kubectl exec -it $POD -n $NAMESPACE -- curl -s /apic-api-r/_search?pretty -H 'Content-Type: application/json' -d '{
"size": 0, "query": { "range": { "datetime": { "gte": "now-1d/d", "lt": "now/d" } } }, "aggs": { "status_codes": {
"filters": { "filters": { "1xx": { "regexp": { "status_code": "1." } }, "2xx": { "regexp": { "status_code": "2." } },
"3xx": { "regexp": { "status_code": "3." } }, "4xx": { "regexp": { "status_code": "4." } }, "5xx": { "regexp": {
"status_code": "5.*" } } } } } } } } }
```

2. Save the file and make it executable, as shown in the following example.

```
chmod +x path/to/my-script-filename.sh
```

3. Run the script.

Invoke it with your namespace and release name as shown in the following example:

```
./my-script-filename.sh namespace release_name
```

Remember, you can omit the release name if it's not needed, as shown in the following example:

```
./my-script-filename.sh namespace
```

For appliance deployments, you can use `default` as the namespace and omit the release name as shown in the following example:

```
./my-script-filename.sh default
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Customizing Analytics data with filters

Define filters to refine the Analytics data that you want to store. Using filters, you can add fields, remove fields, or modify field contents.

### About this task

Use filters to add or remove fields before storing Analytics data. Using an Analytics filter is the only supported way to add top-level Analytics fields.

Tip: If you want to search or visualize data that is stored in request and response headers, you can add fields to a filter to include that information in your Analytics data. For more information on including header fields in your Analytics data, see the example for adding a field to the filter in the topic, [Sample filters for customizing Analytics data](#).

You can filter data for both storing in API Connect and offloading to third-party systems. Before configuring filters, note the following considerations:

- Filters are defined in configuration files, which you modify using the Cloud Manager. You cannot edit the files directly.




- The following filters might cause problems if used incorrectly, so you should only use them when instructed to do so by API Connect Support: `19_filter.conf`, `39_filter.conf`, and `49_filter.conf`.
- Typically you will edit the `69_apic_filter.conf` file. If [analytics message queue](#) is enabled, that filter only applies to data that is stored in API Connect. If you enable message queue and you want to filter data for offloading, you must also modify the `69_offload_filter.conf` file.
- The following fields cannot be modified because they are used by the Analytics service: `org_id`, `catalog_id`, `space_id`, and `developer_org_id`.

See [Sample filters for customizing Analytics data](#) to review filtering examples.

## Procedure

Complete the following steps to define the filter for every Analytics service.

1. Use the Cloud Manager to add your filter:


- a. In the Cloud Manager, click  Topology.
- b. Locate the Analytics service where you want to filter data and click its title to edit the service.
- c. On the Edit Analytics Service page, click Advanced Analytics Configuration to load Kibana.
- d. In Kibana, click Ingestion in the "API Connect" section to open the configuration files selection page.
- e. Select the filter that you want to edit by clicking its tab (all filter files include the word "filter" in the name).  
Remember: Typically you will edit the `69_apic_filter.conf` file. You should only use the other files with guidance from API Connect Support.
- f. Click Save to update the file.

2. If you added or removed a field with the filter, confirm the new field list:


- a. In Kibana, click Index Patterns in the "API Connect" section.
- b. On the Index Patterns page, click Refresh field list and confirm that your new field shows (or that a field you removed no longer shows) in the list.

3. Invoke an API that generates data that your new filter will modify.

4. Use API Manager to view Analytics data for the API that you just invoked:

- a. Log on to the API Manager that contains the Catalog for which you want to view the analytics.
- b. Select Manage  in the navigation.
- c. From the list of available Catalogs in the API Manager Dashboard, select the appropriate Catalog.

If you want to view analytics for a particular Space, click Spaces  in the navigation and then select the Space.

- d. Select Analytics  in the navigation.  
The default analytics dashboard for the Catalog or Space is displayed.
- e. Click Discover.
- f. On the Discover tab, select the All Events saved search.

Verify that the event data for your API reflects the changes imposed by the filter. For example, if you added a field, make sure the new field is included in the event data. If you removed a field, make sure that field is not included. You can use the search filters and time period settings on the page to refine your search.

If you don't see your filtering change reflected in the data, then your filter might be coded incorrectly. Repeat the procedure to verify your coding in the filter and test it again.

- [Sample filters for customizing Analytics data](#)

Review sample filters to see how to code your own filters for refining Analytics data by adding fields, removing fields, or modifying field contents.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Sample filters for customizing Analytics data

Review sample filters to see how to code your own filters for refining Analytics data by adding fields, removing fields, or modifying field contents.

Follow the examples to create filters and add them to the appropriate configuration files as explained in [Customizing Analytics data with filters](#). The sample filters use Ruby code to customize data. For information on using Ruby with Logstash filters plugins, see [Ruby filter plugin](#) in the Logstash documentation.

Typically you will edit the `69_apic_filter.conf` file. If [analytics message queue](#) is enabled, that filter only applies to data that is stored in API Connect. If you enable message queue and you want to filter data for offloading, you must also modify the `69_offload_filter.conf` file.

Remember: You should only use the `19_filter.conf`, `39_filter.conf`, and `49_filter.conf` files with guidance from API Connect Support.

## Adding a new field

You can add custom data to API event data by adding a field to the logging document. For example, if you want to search or visualize data that is stored in request and response headers, you can add fields to a filter to include that information in the top level of your analytics' data.

To avoid naming conflicts with future or current analytics fields, include a prefix that ensures your new field name is unique. For example, instead of naming the field `employee_num` you might name it `x_mycompany_employee_num`.

The following example copies the contents of the `X-Employee-Num` field from the request header and adds it to the `x_mycompany_employee_num` field in the event data. Adding the field to the event data enables you to use the information in visualizations.

```
# Here you should specify any filters that you want to take affect after
# the second stage of APIC Analytics filter processing.
#
# At this point the data has been modified by APIC Analytics, and is in the
# expected documented format. This is where you should feel free to modify
```

```

# the data prior to it being stored in APIC Analytics Storage.
#
# In message queue deployments, this will only affect data that will
# ultimately be stored in APIC Analytics Storage, and will not affect data
# that will be offloaded.
filter {
  if "apicapievent" in [tags] {
    if [request_http_headers] {
      ruby {
        code => "event.get('[request_http_headers]').collect {|i| event.set('[x_mycompany_employee_Num]', i['X-Employee-Num'])}
if i.has_key?('X-Employee-Num')}"
      }
    }
  }
}

```

## Modifying an existing field

Sometimes you don't want to remove a field entirely from your data, you just want to redact sensitive information such as IDs to prevent them from being exposed in visualizations. You can modify the contents of a field and replace information with a symbols or a message.

Note: The following fields should not be modified if the data is being written to internal analytics storage: `org_id`, `catalog_id`, `space_id`, `developer_org_id`, `datetime`, and `@timestamp`.

The following example replaces sensitive information in the "Employee-Name" and "Employee-ID" header fields with the string "\*\*\*\*\*sanitized\*\*\*\*\*".

```

# Here you should specify any filters that you want to take effect after
# the second stage of APIC Analytics filter processing.
#
# At this point the data has been modified by APIC Analytics, and is in the
# expected documented format. This is where you should feel free to modify
# the data prior to it being stored in APIC Analytics Storage.
#
# In message queue deployments, this will only affect data that will
# ultimately be stored in APIC Analytics Storage, and will not affect data
# that will be offloaded.
filter {
  if "apicapievent" in [tags] {
    if [request_http_headers] {
      ruby {
        code => "headers=['X-Employee-Name','X-Employee-ID']; newHeaders = event.get('[request_http_headers]').collect {|i|
headers.each {|header| i[header] = '*****sanitized*****' if i.has_key?(header)}; i}; event.set('[request_http_headers]',
newHeaders)"
      }
    }
  }
}

```

## Removing an existing field

Use the mutate `remove_field` operation to delete a field from the Analytics data. To remove multiple fields, delimit the field names with commas.

The following example removes the following fields from the Analytics data: `request_http_headers`, `response_http_headers`, `request_body`, `response_body`, and `query_string`.

```

# Here you should specify any filters that you want to take effect after
# the second stage of APIC Analytics filter processing.
#
# At this point the data has been modified by APIC Analytics, and is in the
# expected documented format. This is where you should feel free to modify
# the data prior to it being stored in APIC Analytics Storage.
#
# In message queue deployments, this will only affect data that will
# ultimately be stored in APIC Analytics Storage, and will not affect data
# that will be offloaded.
filter {
  if "apicapievent" in [tags] {
    mutate {
      remove_field => ["request_http_headers", "response_http_headers", "request_body", "response_body", "query_string"]
    }
  }
}

```

## Filter and output plugin for splunk

When you create a filter for use with splunk, you must also create an output plugin for it.

Filter:

```

# Here you should specify any filters that you want to take effect after
# the second stage of APIC Analytics filter processing.
#
# At this point the data has been modified by APIC Analytics, and is in the
# expected documented format. This is where you should feel free to modify
# the data prior to it being stored in APIC Analytics Storage.
#
# In message queue deployments, this will only affect data that will
# ultimately be stored in APIC Analytics Storage, and will not affect data
# that will be offloaded.
filter {
  if "apicapievent" in [tags] {

```

```

ruby {
  code => "event.set('@metadata[newevent]', event.to_json)"
}
}
}

```

Output plugin:

```

apicapievent" in [tags] {
  http {
    url => "http://your-domain/services/collector/event"
    http_method => "post"
    codec => "json"
    content_type => "application/json"
    id => "offload_http"
    format => "message"
    message => '{"event": "%{[@metadata][newevent]}", "index": "applications-int-ms", "sourcetype": "json:apimlogs"}'
    headers => ["Authorization", "Splunk XXXX"]
  }
}
}

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring data retention and index rollover time periods

You can set a time period for data retention and index rollover using the Advanced Analytics Configuration option for an analytics service.

### Before you begin

This task assumes you have registered an analytics service and associated it with a gateway service. See [Registering an analytics service](#) and [Associating an analytics service with a gateway service](#).

One of the following roles is required to set the Data Retention and Index Rollover values:

- Administrator
- Topology Administrator
- Owner
- A custom role with the `topology:manage` permission

### About this task

Each API call is recorded as a document in a shared index. Periodically, a new index is created and the previous index is stored. You can specify Index Rollover and Data Retention settings that control how often new indexes are created, and how long indexes are stored.

The Index Rollover settings determine how long an index accumulates data before it "rolls over" into storage and a new index is created. You can control the duration of an index two ways: by setting a maximum age for the index (default is 1 day), and by setting a maximum number of documents that can be recorded in the index (default is 25 million documents). When the rollover occurs, a new index is created and documents are recorded there, while the previous index is stored until its age exceeds the Data Retention setting. If you change the rollover settings, consider both the amount of data being stored within each index as well as the number of indexes being stored. Allowing indexes to grow too large, or storing too many indexes at once, might cause issues.

The Data Retention setting determines how long the stored indexes (and the data they contain) are retained. Once every day, all indexes that are older than the specified retention period are purged. Retention is based on the index's age, not the age of the data within that index. When the index's own creation date exceeds the retention period, the index and all its data is deleted even if some of the data stored in that index is younger than the retention period. Data is purged by deleting entire records so you cannot choose to delete only certain fields.


The default retention period is 90 days. Reasons for changing this setting include storage constraints, and data retention requirements for your organization. You might want to set this value to be less than the default if a large number of API events are stored, especially if payload logging is enabled on the APIs. If you modify the retention value, you should modify the index rollover setting as well to ensure that they remain in sync.

The most recently created index (the index that is currently being written to) is not deleted, even if you set the retention period as small as 1 hour. If you regularly need to delete data quickly, adjust the Index Rollover setting so that a new index is rolled over to sooner, and the old index can be deleted.

To change the data retention and index rollover settings, you must configure you settings in the Cloud Manager, and then edit the schedule in the related cronjob as explained in the following procedure.

## Procedure

1. Modify the Cloud Manager properties that specify the data retention and index rollover settings:

- a. In the Cloud Manager, click  Topology.
- b. Locate the Analytics service that requires data retention and index rollover settings, then click the title to open the Edit Analytics Service page.
- c. Click the Advanced Analytics Configuration link to load Kibana.
- d. In Kibana, click Storage in the API Connect section.
- e. For Data Retention, select a number and a unit of time, and then click Save.
- f. Enter the desired value for Index Rollover and click Save.

## 2. Modify the schedule for the cronjob that manages data retention and index rollovers.

If you want to set the data retention or index rollover frequency to more than once per day, then you must also edit the schedule that determines when the related cronjob runs.

On all platforms, the retention and rollover actions are triggered using a [Kubernetes cronjob](#), which runs once per day by default. If you want the cronjob to run more frequently, modify its schedule by completing the following steps:

### a. View the current cronjob schedule by running the following command:

- Kubernetes:

```
kubectl -n <apic-analytics-namespace> get cronjob
```

- OpenShift and Cloud Pak for Integration:

```
oc -n <apic-analytics-namespace> get cronjob
```

- VMware:

- Run the following command to connect as the API Connect administrator, replacing `<ip_address>` with the appropriate IP address:

```
ssh <ip_address> -l apicadm
```

- When prompted, select Yes to continue connecting.
- When you are connected, run the following command to receive the necessary permissions for working directly on the appliance:

```
sudo -i
```

- View the current cronjob schedule by running the following command:

```
kubectl -n <apic-analytics-namespace> get cronjob
```

The response displays the current **SCHEDULE**, as in the following example:

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
analytics-cj-retention	30 1 * * *	False	0	8h	266d
analytics-cj-rollover	15,45 * * * * *	False	0	19m	266d

### b. Run the following command to edit the cronjob so you can modify the schedules:

- Kubernetes:

```
kubectl -n <apic-analytics-namespace> edit cronjob <cronjob_name>
```

- OpenShift:

```
oc -n <apic-analytics-namespace> edit cronjob <cronjob_name>
```

- VMware:

```
kubectl -n <apic-analytics-namespace> edit cronjob <cronjob_name>
```

### c. Adjust the cronjob's rollover and retention schedules, and then save your changes.

Important: Do not make any other changes to the cronjob. Other changes might result in the retention and rollover actions failing to complete successfully. To configure the job to run more than once a day, modify the schedule and update the minutes and hour values as needed (the values in positions 1 and 2).

The format for the schedule consists of 5 fields, separated with spaces. Each field is represented with a \* and based on its position, controls the value shown in the following list:

- \* \* \* \* \* schedule includes 5 fields
- \* \* \* \* +----- position 5: day of week (0 - 6 where 0=Sunday)
- \* \* \* +----- position 4: month (1 - 12)
- \* \* +----- position 3: day of month (1 - 31)
- \* +----- position 2: hour (0 - 23)
- +----- position 1: minutes (0 - 59)

For example, the following schedule runs a job at 30 minutes (position 1) past the first hour (position 2) every day: 30 1 \* \* \*

If you want a job to run more than once an hour, you can specify multiple minutes values by separating them with a comma. The following schedule runs a job at 15 minutes and 45 minutes past the every hour of every day: 15,45 \* \* \* \*

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring analytics offload for API Connect

The event data that is generated and collected in your API Connect on-premises cloud can be forwarded to different destination targets for display and analysis. Destination targets include the API Connect user interfaces, and third-party systems that are external to API Connect.

### About this task

API event data can be forwarded to a third-party system for data consolidation within data warehouse systems, enhanced monitoring, and richer analytical data processing. The data offload capability in API Connect enables users to form correlations with other data sources on a single platform, and helps to reduce the time between analysis, decision, and action. When you offload event data to a third party system, the data is streamed in real time to the target system as the events occur. Multiple target systems are supported for the data offload.

You can configure a combination of destination systems to which data is forwarded for analysis and display. You can choose one of the following options:

- Enable access to analytics data in the API Manager user interface. This is the default setting.
- Disable access to analytics data in API Connect and instead re-route the data streams to one or more third-party systems. When you select this method, the data is not written to the API Connect analytics management server.
- Enable access to analytics data in API Connect and simultaneously stream the data to one or more third-party systems.
- Enable the Analytics Message Queue. The Analytics Message Queue acts as a broker between the ingestion pipelines to ensure data collection continued uninterrupted if the offline endpoint becomes nonfunctional. See
- Completely disable access to analytics data.

Use the following information to configure destination targets for your analytics data:

- [Supported event types for analytics offload](#)  
API Connect generates API event data. The captured event data can be viewed in the API Manager or offloaded to third-party systems.
- [Supported third-party systems for analytics offload](#)  
You can offload the analytics data for API events to several third-party systems. The supported systems include HTTP servers, Elasticsearch clusters, Kafka clusters, and Syslog servers.
- [Providing a custom certificate for analytics offload](#)  
Optionally configure a private or self-signed certificate to secure the connections between IBM® API Connect Analytics and your external offload endpoint.
- [Configuring output plugins for analytics offload](#)  
In API Connect version 2018.3.7, and later, you can configure output plugins for third-party systems by editing the `outputs.yml` and `offload_output.conf` files. In earlier versions, you can configure output plugins for third-party systems in the `logstash.conf` file to offload the analytics data for API Connect. The output plugins point to one of the following target systems: HTTP, Elasticsearch, Kafka, and Syslog servers. The output plugin for displaying analytics data in the API Manager is included by default.
- [Configuring the analytics message queue](#)  
The analytics message queue is embedded in the data ingestion pipeline. It is most useful when offloading data from the analytics subsystem to a third-party system. It ensures that data collection continues in the event of a failure.
- [Configuration files for analytics offload customization](#)  
Beginning with API Connect version 2018.3.7, settings for analytics offload customization are maintained in the `offload_output.conf`, `19_filter.conf`, `39_filter.conf`, `49_apic_filter.conf`, and `69_apic_filter.conf` files.
- [Disabling access to analytics event data in API Connect](#)  
If you configured the IBM API Connect Analytics subsystem to offload data to a third-party product, you might not want to store a copy of the data in API Connect. You can disable the storage of analytics data in API Connect by modifying the `outputs.yml` file.
- [Troubleshooting your analytics offload](#)  
Perform these checks if the offload fails. Common causes for an offload failure are connection or security-related issues.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Supported event types for analytics offload

API Connect generates API event data. The captured event data can be viewed in the API Manager or offloaded to third-party systems.

### API events

---

An API event is logged each time an API operation is invoked, and an event record is generated for each API event on the Gateway server. The API event record contains information about the API call. When an analytics service is associated with a gateway, it will capture API events.

The content of the record depends on the logging policy that is set for the operation. For information about how to configure your logging preferences for API events, see [activity-log policy](#) and [Including components in your assembly](#).

By default, API providers can view API event data in a number of ways:

- In the analytics dashboard for a catalog in API Manager. See `Manage > Catalog_name > Analytics`. These dashboards group together a set of related *visualizations* that depict analytics data as a graphical or metric representation. Default dashboards are provided for viewing and customizing analytics data for APIs, Plans, Products, and Catalogs (including Spaces if enabled). Custom dashboards can also be created. For more information about accessing and working with the API event data shown in API, Plan, and Product dashboards, see [Managing your Products](#).
- By using the debug function for the integrated test tool that is provided in the API Manager user interface. The debug function is typically used to view information about the API configuration in order to identify possible causes for an API invocation failure.

API consumers who are members of a consumer organization can also view analytics data within dashboard views in the Developer Portal user interface. The dashboard views depict API invocation metrics about the APIs used either by a single application or within the entire consumer organization. For more information about accessing the analytics data for APIs that are invoked by developer applications, see [Analytics in the Developer Portal](#).

Effect of switching off IBM API Connect Analytics for API events:

For information on disabling access to the analytics data, see [Disabling access to analytics event data in API Connect](#). Following are the effects of disabling access to analytics data within API Connect:

- Data streaming from the Gateway to the UIs is immediately terminated. Dashboards will still be visible, but may contain no data, or they might show data for a time period in the past, depending on when Analytics was disabled.
- No API data will be displayed when API developers use the debug function while testing their APIs locally in the API Manager user interface.

---

## Related concepts

- [Supported third-party systems for analytics offload](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Supported third-party systems for analytics offload

You can offload the analytics data for API events to several third-party systems. The supported systems include HTTP servers, Elasticsearch clusters, Kafka clusters, and Syslog servers.

For details about how to set up the data offload to these systems, see [Configuring output plugins for analytics offload](#). Following are descriptions of the third-party systems:

- [HTTP](#)
- [Elasticsearch \(Version 6 or 7\)](#)
- [Apache Kafka \(Version 1.0.0\)](#)
- [Syslog](#)

Attention: When you set up data offloading, the third-party endpoint must be available at all times to prevent the loss of analytics data.

### HTTP

---

You can offload API Connect analytics data to a web server that processes requests by using HTTP. Note that if the offload HTTP target is unreachable, the events will be lost.

To configure data offload to an HTTP server, you must provide the server URL. You can optionally add standardized or custom HTTP headers to define additional operating parameters.

You can also configure your data offload to forward the real-time event stream to other systems that provide enhanced search, analytics, and visualization capabilities. For example, you can forward your API Connect event data to a Splunk deployment. Or, you can forward your API Connect event data to an in-memory data structure store like Redis.

Refer to the Elasticsearch documentation for configuring HTTP plugins at <https://www.elastic.co/guide/en/logstash/6.8/plugins-outputs-http.html>

### Elasticsearch (Version 6 or 7)

---

You can offload API Connect analytics data to an Elasticsearch cluster, either a single node or a collection of nodes.

To configure data offload to an Elasticsearch cluster, you must provide one or more server URLs, define how indices should be created by specifying a dynamically configured name, specify the number of primary shards for storing the indexed documents, and specify a number of replica shards for redundancy.

The naming syntax that you specify for your indices determines how many indices are created and how they are characterized. For example, you can define indices that are created based on a date pattern (for example, daily or weekly), or indices that are created to store data by provider organization or by API name.

Refer to the Elasticsearch documentation for configuring plugins at <https://www.elastic.co/guide/en/logstash/6.8/plugins-outputs-elasticsearch.html>

### Apache Kafka (Version 1.0.0)

---

You can offload API Connect analytics data to an Apache Kafka target system that provides a stream processing platform for handling real-time data feeds. In this scenario, API Connect acts as a *producer* of data for a Kafka system that runs as a cluster on one or more servers (known as Kafka brokers).

The data that you offload to Kafka will be published to a *topic*, which is divided into a number of *partitions* that are implemented as structured commit logs and spread across the Kafka brokers. The data is written to the tail of these logs and can subsequently be distributed to the *consumers* that are configured in Kafka to process published feeds.

To configure data offload to a Kafka cluster, you must provide host and port connection details for one or more servers, and the name of the Kafka topic to which you want to publish the offloaded data. For logging and monitoring purposes within Kafka, you can optionally specify a string identifier by which API Connect can be uniquely identified.

Refer to the Elasticsearch documentation for configuring plugins at <https://www.elastic.co/guide/en/logstash/6.8/plugins-outputs-kafka.html>. Note that the Logstash plugin is based on `kafka-client version 2.3.1` and not `v2.1.0` as specified in the logstash doc.

### Syslog

---

You can offload API Connect analytics data to a Syslog server (or collector) that is configured to accept the event data for data consolidation, analysis, or review. The implementation supports both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). The Syslog server can optionally be configured to forward the API Connect data to an external datastore for further processing or archiving.

Restriction: Syslog does not support the use of chained client certificates for TLS profiles.  
To configure data offload to Syslog, you must provide host and port connection details for the server.

Attention: Due to a third-party issue with Logstash, most of the facility values are not set correctly on the offload server. As a temporary workaround, set the facility to "user-level" to ensure that the corresponding value is correctly sent to the offload server. When Logstash corrects this problem, you must update the offload configuration to change the value to a more appropriate setting.

Refer to the Elasticsearch documentation for configuring plugins at <https://www.elastic.co/guide/en/logstash/6.8/plugins-outputs-syslog.html>.

## Related concepts

- [Supported event types for analytics offload](#)

## Related information

- [Splunk](#)
- [Redis](#)
- [Elasticsearch](#)
- [Kafka](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Providing a custom certificate for analytics offload

Optionally configure a private or self-signed certificate to secure the connections between IBM® API Connect Analytics and your external offload endpoint.

### About this task

If you configure Analytics for offloading and your endpoint requires a particular certificate, you can configure the certificate now and then reference it in the offload plugin when you complete the next task.

### Procedure

1. Create a Kubernetes secret to contain the certificates, and then apply it to the cluster where the Analytics subsystem runs.
  - a. Log in to a server that has kubectl access to the Kubernetes cluster where API Connect Analytics is deployed.  
For OVA deployments, you must log in to one of the running Analytics VMs as a user with kubectl access.
  - b. Encode the certificates file in base64.  
For offload certificates, you only need to provide the certificates that your endpoint requires. Some offload plugins require certificates as JKS files and others accept text-based files such as PEM files; use the format that is appropriate for your plugin.

Attention: If you use a JKS file for certificates, then this step is required because the non-text JKS file must be encoded in advance. If you use a text-based file such as PEM, then you can choose between encoding the file now or skipping this step and adding the keys to the secret in the next step.  
There are several ways to encode the file. If `cat` and `base64` are available, then you can run the following command to encode the file:

```
cat keystore.jks | base64
```

Copy the output so that you can paste it into the secret in the next step.

- c. Create a Kubernetes secret and add the certificate.
  - i. Create a YAML file to contain the secret; for example, you can call the file `offload_certs.yaml`. The secret's name that you specify in the file does not have to match the file name. In this example, the secret's name is `offload-certificates`.  
Add as many keys as you want, using any field names that you want. For each key, a file based on the corresponding field name is created in the container at `$(OFFLOAD_CERTS_DIR)`, and you can use it in your offload plugin's configuration. For example, if you paste a certificate in the `cacert` field as shown in the following example, then a file named `cacert` is stored in the `$(OFFLOAD_CERTS_DIR)` location.

```
apiVersion: v1
kind: Secret
metadata:
  # Change value of name to be whatever you wish secret to be called
  name: offload-certificates
  # Only base64 encoded data should be placed in data section; JKS keys would go here
data:
  keystore.jks: "output_of_base64_encoded_jks"
stringData:
  cacert: |-
    -----BEGIN CERTIFICATE-----
    MIIDIDCCAomgAwIBAgIEND70zzANBqkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzEQMA4GA1UE
    ChMHRXF1aWZheDEtMCsGA1UECzMkRXF1aWZheCBTZWN1cmUgQ2VydG1maWNhdGUgQXV0aG9yaXR5
    MB4XDTK4MDgyMjE2NDE1MVoXDTE4MDgyMjE2NDE1MVoVotjELMAkGA1UEBhMCMVVMxEDA0BgNVBAoT
    B0VxdWlmYXg5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5LTA5
    nzANBqkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwV2xWGCiYu6gmi0fCG2RFGiYCh7+2gRvE4RiIcPR
    fM6fBeC4AfBONOziipUEZKzxa1NfBbPLZ4C/QgKO/t0BCeZhABRP/PvwDN1Du1sr4R+AcJkVV5MW
    8Q+XarfCaCMczE1ZMKxRHjuvK9buY0V7xd1fUNLjUA86iOe/FP3gx7kCAwEAaAaOCAQkwwgEgEFMHAG
    A1UdHwRPMGcwZaBjocGGkXzBdMQswCQYDVQQGEwJVUzEQMA4GA1UEChMHRXF1aWZheDEtMCsGA1UE
    CxMkRXF1aWZheCBTZWN1cmUgQ2VydG1maWNhdGUgQXV0aG9yaXR5MQ0wCwYDVQQDEwRDUkwxMBoG
    A1UdEAQTMGBGDzIwMTgwODIyMTY0MTUxWjALBgNVHQ8EBAMCAQYwHwYDVR0jBBgwFoAUSOzo+SvS
    spXXR9gjIBBPM5iQn9QwHQYDVR0OBBYEFjmaPkr0rKV10fYIyAQZtOYkKJ/UMAwGA1UdEwQFMAMB
    Af8wGgYJKoZIhvcZ9B0EABA0wCxsFVjMuMGMDAgBAMA0GCSqGSIsB3DQEBBQUAA4GBAFjOKer89961
    zK5F7WF0bnj4JXMJTENAKaSbn+2kmOeUJXRmm/kEd5jhW6Y7qj/WsjTVbJmcVfewCHRPSqnI0kK
    BIZCe/zuf6IWUrvnZ9NA2zsmWLIodz2uFHdh1voqZiegDfqnclzqcPGUIWVEX/z87y1oqaKHee95
    70+sB3c4
    -----END CERTIFICATE-----
```

- ii. Paste the certificate into the file.  
The method for adding the certificate to the secret depends on whether you encoded the file in the previous step:



- Base64-encoded file: In the **data** section of the secret, type a name for the field (you can choose the name). Paste the encoded output from the previous step (enclosed in "") as the value.
  - Plain text file: In the **stringData** section of the secret, type a name for the key or certificate (you can choose the name), and paste the value.
- iii. Save the file.
- d. Update the cluster with the new certificate.  
Run the following command to update the cluster:

```
kubectl apply -f offload_certs.yaml -n your_namespace
```

where:

- **offload\_certs.yaml** is the secret's file name (not the name of the secret).
- **your\_namespace** is the name of your deployment's namespace. The namespace is only needed for Kubernetes deployments. For OVA deployments, you can omit the entire **-n your\_namespace** parameter from the command. By default, kubectl is already configured to match the namespace where your Analytics resources are deployed.

- e. Optionally create another Kubernetes secret, which can be used in environment variables.

If you have sensitive data such as a certificate passphrase that must be available in text format, you can store it in another secret.

If your passphrase is not sensitive (such as the JKS default passphrase), then you can supply it directly in the logstash configuration, and skip this step.

- Create a new YAML file; for example, call the file **offload\_env\_var.yaml**. The secret's name that you specify in the file does not have to match the file name. In this example, the secret's name is **offload-environment**.
- Paste the environment variable name and the passphrase (in plain text) into the **stringData** section of the file as shown in the following code example:

```
apiVersion: v1
kind: Secret
metadata:
  name: offload-environment
stringData:
  # Example environment variable names, you'll reference these from within logstash
  OFFLOAD_JKS_PASSWORD: password123
```

- Run the following command to update the cluster:

```
kubectl apply -f offload_env_var.yaml -n your_namespace
```

where **offload\_env\_var.yaml** is the secret's file name.

2. Update the Analytics subsystem so that it can use the new secrets.

- Create an **extra-values.yaml** file that specifies the names of your certificate and environment variables secret files.

Include the following lines in the **extra-values.yaml** file, making sure to use the names of your Kubernetes secrets (value the of the **name** field in the **metadata** section of the secret, which might not match the file's name):

```
apic-analytics-ingestion:
# offloadSecretEnv will mount all objects in specified secret name provided as environment variables
offloadSecretEnv: offload-environment
# offloadSecretCerts will mount all objects in specified secret as files on containers, using key as file name
offloadSecretCerts: offload-certificates
```

where:

- The **offloadSecretEnv** setting provides the name of the secret that is used by environment variables (to contain the certificate passphrase, for example). Keys provided in this file are made available as environment variables. The sample **offload\_env\_var.yaml** file in step 1 uses **offload-environment** as the secret's name.
- The **offloadSecretCerts** setting provides the name of the secret that contains the certificates needed for connecting to the offload endpoint. Keys provided in this file are stored as files in the **\$(OFFLOAD\_CERTS\_DIR)** directory. The sample **offload\_certs.yaml** file in step 1 uses **offload-certificates** as the certificates secret's name.

- Update the Analytics subsystem.

- Log in to the server where you run **apicup**, and navigate to the project directory.  
In Kubernetes deployments this is probably the same server that you used for step 1, but for OVA deployments it is a different server.
- Run the following **apicup** commands to reinstall the Analytics subsystem using the configuration settings in the **extra-values.yaml** file.

```
apicup subsys set analytics extra-values-file=~ /extra-values.yaml
apicup subsys install analytics
```

3. Update the logstash configuration to use the new secret.

Modify the logstash configuration and replace the references to the certificate file with references to the new secret:

- Items specified in the secret provided to **offloadSecretEnv** are available to use in logstash config files as environment variables; for example: **\$(OFFLOAD\_JKS\_PASSWORD)**.
- Keys from **offloadSecretCerts** are stored as files in **\$(OFFLOAD\_CERTS\_DIR)**. The path and file name can be referenced in the logstash configuration; for example, **\$(OFFLOAD\_CERTS\_DIR)/keystore.jks** for a JKS file or **\$(OFFLOAD\_CERTS\_DIR)/cacert** for a PEM file.
- For information on configuring the logstash plugins, see [Configuring output plugins for analytics offload](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring output plugins for analytics offload



In API Connect version 2018.3.7, and later, you can configure output plugins for third-party systems by editing the `outputs.yml` and `offload_output.conf` files. In earlier versions, you can configure output plugins for third-party systems in the `logstash.conf` file to offload the analytics data for API Connect. The output plugins point to one of the following target systems: HTTP, Elasticsearch, Kafka, and Syslog servers. The output plugin for displaying analytics data in the API Manager is included by default.

## About this task

Important: After you upgrade from a version of API Connect that is earlier than 2018.3.7 to version 2018.3.7, or later, you must reconfigure any customizations that you made in your `logstash.conf` file. Beginning with version 2018.3.7, the sections of the `logstash.conf` file are separated into smaller files to make them easier to use. Use the following chart to recreate the customizations:

Location of customization within <code>logstash.conf</code> file	Type of customization	Customization with new configuration files
input	Any customizations	This was unsupported in the previous versions and remains unsupported.
filter	Enable <code>geoip</code>	In the <code>install/upgrade</code> section of the <code>extra_values.yml</code> file, set the <code>apic-analytics-ingestion.geoIpEnabled</code> value to <code>true</code> .
filter	Additions to the pipeline (pre-apiconnect changes)	Add these to the <code>49_apic_filter.conf</code> file.
filter	Additions to the pipeline (post-apiconnect changes)	Add these to the <code>69_apic_filter.conf</code> file.
output	Disable the API Connect native analytics storage.	Set <code>apic_output_enabled</code> to the value of <code>false</code> in the <code>outputs.yml</code> or set <code>apic-analytics-ingestion.outputApicEnabled</code> to <code>false</code> in the <code>install/upgrade</code> section of the <code>extra_values.yml</code> file.
output	Add offload configuration	<p>Complete one of the following procedures to add the offload configuration:</p> <ul style="list-style-type: none"> <li>Set <code>offload_output_enabled</code> to <code>true</code> in the <code>outputs.yml</code> file</li> <li>In the <code>install/upgrade</code> section of the <code>extra_values.yml</code> file, set the value of <code>apic-analytics-ingestion.outputOffloadEnabled</code> to <code>true</code>.</li> </ul> <p>After you complete either of these steps, you must add your offload configuration to the <code>offload_output.conf</code> file.</p>

If you need to review the previous changes to your `logstash.conf` file, they are stored in the `analytics-ingestion-pipeline` configuration map in the Kubernetes instance. One of the following roles is required to configure the output plugins for analytics offload:

- Administrator
- Topology Administrator
- Owner
- A custom role with the `topology:manage` permission

Output plugins control the analytics data offload to both the API Manager or to a third-party system. The data will be offloaded to the output plugins that are configured in the `offload_output.conf` file.

If you are using SSL to secure the communications between API Connect and your offload endpoint, then you will need to include a path to a file containing a list of trusted certificates. Both KeyStore and PEM bundles are available.

Note: The example uses default certificates. If you want to provide private or self-signed certificates instead, configure them as explained in [Providing a custom certificate for analytics offload](#) and then reference them in your `logstash` configuration.

Keystore:

```
Keystore: ` /usr/lib/jvm/jre/lib/security/cacerts `
passphrase: ` changeit `
type: ` JKS `
```


PEM:

```
cacert: ` /etc/pki/tls/cert.pem `
```

Important: When you configure the offload of analytics data, or enable or disable access to analytics data in API Connect, the configuration is applied across all provider and consumer organizations in the cloud infrastructure.

## Procedure

Complete the following steps to configure analytics offload for your cloud:

- In the Cloud Manager, click  Topology.
- Locate the Analytics service that is associated with the Gateway service, then click the title to open the Edit Analytics Service page.
- Click the Advanced Analytics Configuration link to load Kibana.
- In Kibana, click Ingestion in the API Connect section to open the configuration files selection page in the editor.
- Open the `outputs.yml` file by selecting its tab.
- Change the value of `offload_output_enabled`: to `true`.
- Save the values.
- Open the `offload_output.conf` file by selecting its tab.
- Add your new offload configuration within the `if` statement. The following content is an example of this file:

```
output {
  if "apicapievent" in [tags] {
    kafka {
      topic_id => "test"
      bootstrap_servers => "example.com:9093"
      codec => "json"
      id => "kafka_offload"
      ssl_truststore_location => "/usr/share/logstash/jdk/lib/security/cacerts"
      ssl_truststore_password => "changeit"
      ssl_truststore_type => "JKS"
    }
  }
}
```

```

    security_protocol => SSL
  }
}
}

```

Note: To disable storing the data in API Manager, change the `apic_output_enabled` setting in the `outputs.yml` file to `disabled` as shown in: [Disabling access to analytics event data in API Connect](#).

- The following examples show how to configure output plugins in `offload_output.conf` for each supported offload type. These examples use SSL with a public SSL certificate mounted at the endpoint. For a complete list of configuration options, refer to the logstash documentation.

#### HTTP

The following example configures an HTTP output plugin. For more options, refer to the Logstash documentation: [Http output plugin](#).

```

http {
  url => "example.com"
  http_method => "post"
  codec => "json"
  content_type => "application/json"
  id => "offload_http"
  truststore => "/usr/lib/jvm/jre/lib/security/cacerts"
  truststore_password => "changeit"
  truststore_type => "JKS"
}

```

The following example configures an HTTP output plugin using a custom certificate. For more information on using a custom certificate for your offload endpoint, see [Providing a custom certificate for analytics offload](#).

```

http {
  url => "https://example.com:443"
  http_method => "post"
  codec => "json"
  content_type => "application/json"
  id => "offload_http"
  cacert => "${OFFLOAD_CERTS_DIR}/cacert.pem"
  client_cert => "${OFFLOAD_CERTS_DIR}/client.pem"
  client_key => "${OFFLOAD_CERTS_DIR}/client.key"
}

```

#### Elasticsearch

The following example configures an Elasticsearch output plugin. For more options, refer to the Logstash documentation: [Elasticsearch output plugin](#).

```

elasticsearch {
  hosts => "https://example.com:443"
  index => "apiconnect"
  ssl => true
  cacert => "/etc/pki/tls/cert.pem"
}

```

#### Kafka

The following example configures a Kafka output plugin. For more options, refer to the Logstash documentation: [Kafka output plugin](#).

```

kafka {
  topic_id => "test"
  bootstrap_servers => "example.com:9093"
  codec => "json"
  id => "kafka_offload"
  ssl_truststore_location => "/usr/share/logstash/jdk/lib/security/cacerts"
  ssl_truststore_password => "changeit"
  ssl_truststore_type => "JKS"
  security_protocol => SSL
}

```

#### Syslog

The following example configures a Kafka output plugin. For more options, refer to the Logstash documentation: [Syslog output plugin](#).

Restriction: Syslog does not support the use of chained client certificates for TLS profiles.

```

syslog {
  host => "example.com"
  port => 601
  protocol => "ssl-tcp"
  id => "offload_syslog"
  appname => "apiconnect"
  msgid => "%{org_id}"
  facility => "log_audit"
  severity => "informational"
  codec => "json"
  ssl_cacert => "/etc/pki/tls/cert.pem"
}

```

- Click Save.

Important: It might take a few minutes for the new configuration to be applied to all your pods. Check the logs on the pods to ensure that there are no Logstash errors.

- If you want to include the `client_geoip` and `gateway_geoip` fields in your Analytics data, configure your Kubernetes ingresses/cluster to include the `X-Forwarded-For` header in the data that is collected by the DataPower gateway and passed to APIC Analytics.

This step is only required if you want to include the fields in your Analytics data. For information on the configuration task, see the following Kubernetes documentation:

- [NGINX Configuration "use-forwarded-headers"](#)
- [Troubleshooting NGINX Ingress](#)

## What to do next

Log in to the target system and verify that you can see the data stream for API Connect events.  
Attention: The third-party endpoint must be available at all times to prevent the loss of analytics data.  
**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring the analytics message queue

The analytics message queue is embedded in the data ingestion pipeline. It is most useful when offloading data from the analytics subsystem to a third-party system. It ensures that data collection continues in the event of a failure.

### About this task

The analytics message queue is an optional service that provides a reliable ingestion pipeline, especially when offloading analytics to a third party system. The message queue ensures that the analytics pipeline will not be blocked when either analytics storage or the offload endpoint becomes unavailable. Data collection will continue internally in the analytics subsystem, and will be buffered if analytics offload has been configured. With the message queue enabled, data storage will continue even if the offload endpoint is unresponsive. Additional benefits include better handling of traffic spikes and a larger buffer to handle Elasticsearch issues such as red indices, mapping issues, etc.

### Procedure

Complete the following steps to configure an analytics message queue:

1. Enable the message queue using the APICUP installer and install the analytics subsystem. The command is `apicup subsys set analytics enable-message-queue=true`.
2. Set an optional `mq-storage-class`, or use the `storageClass` object.  
This parameter is used in Kubernetes installations only. It is not used in VMware environments (OVA files).
3. Install the Analytics subsystem to activate the message queue.
4. Optionally, configure analytics third-party offload if required. See [Configuring analytics offload](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuration files for analytics offload customization

Beginning with API Connect version 2018.3.7, settings for analytics offload customization are maintained in the `offload_output.conf`, `19_filter.conf`, `39_filter.conf`, `49_apic_filter.conf`, and `69_apic_filter.conf` files.

The settings for configuring the offload of analytics data are contained in multiple files to make it easier to customize the content. Each of these files is displayed as a tab in the Cloud Manager Console. The files can be opened and changed in the Cloud Manager Console by following the instructions in [Configuring output plugins for analytics offload](#).

### outputs.yml file

The `outputs.yml` file is where you set the values for enabling and disabling the settings for where your analytics events are maintained. You can offload the data to a third-party collection repository, maintain the data on the API Connect internal collection repository, enable both methods to collect the data internally and externally, or disable them both to collect no analytics data. To enable the offloading of data, the value for the `offload_output_enabled` setting must be set to `true`. If the `offload_output_enabled` is set to `false`, no data is offloaded to an external repository. To maintain the data in the internal API Connect analytics data collection repository, the `apic_output_enabled` setting must be set to `true`.

When the value of the `apic_output_enabled` parameter is set to `false`, the internal API Connect analytics data collection is disabled. There is no analytics data captured internally when this is set to `false`.

Important: When the values of the `offload_output_enabled` and the `apic_output_enabled` parameters are both set to `false`, no analytics data is being collected. The following example shows sample contents of the `outputs.yml` file:

```
apic_output_enabled: true
offload_output_enabled: false
```

### offload\_output.conf file

You can specify outputs in the `offload_output.conf` file to identify which API events that you want to offload. The API events that are not offloaded are either collected in the API Connect if internal analytics data collection is enabled, or discarded if it is not enabled.

The following example shows sample contents of the `offload_output.conf` file:

```
output {
  if "apicapievent" in [tags] {
```

```
}  
}
```

---

## 19\_filter.conf

You can specify filters in the 19\_filter.conf file to apply filters to the data immediately after it enters the pipeline from the gateway.

Note: Do not filter using this file unless you are instructed to do so by the API Connect support team.

The following example shows sample contents of the 19\_filter.conf file:

```
filter {  
  if "apicapievent" in [tags] {  
  }  
}
```

---

## 39\_filter.conf

You can specify filters in the 39\_filter.conf file to that apply to the data after it is filtered according to the settings in the 19\_filter.conf file.

Note: Do not filter using this file unless you are instructed to do so by the API Connect support team.

The following example shows sample contents of the 39\_filter.conf file:

```
filter {  
  if "apicapievent" in [tags] {  
  }  
}
```

---

## 49\_apic\_filter.conf

You can specify filters in the 49\_apic\_filter.conf file to that apply to the data after it is filtered according to the settings in the 19\_filter.conf file and the 39\_filter.conf file.

Note: Do not filter using this file unless you are instructed to do so by the API Connect support team.

The following example shows sample contents of the 49\_apic\_filter.conf file:

```
filter {  
  if "apicapievent" in [tags] {  
  }  
}
```

---

## 69\_apic\_filter.conf

You can specify filters in the 69\_apic\_filter.conf file to that apply to the data after it is filtered according to the settings in the 19\_filter.conf file, the 39\_filter file, and the 49\_apic\_filter.conf file. The filter settings in this file only affect the analytics data that is stored on the internal API Connect analytics data collection. It cannot be used to filter offloaded data.

You can filter the events in this file based on the event record fields that are described in [API event record fields](#).

The following example shows sample contents of the 69\_apic\_filter.conf file:

```
filter {  
  if "apicapievent" in [tags] {  
  }  
}
```

---

## Related concepts

- [Supported third-party systems for analytics offload](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Disabling access to analytics event data in API Connect

If you configured the IBM® API Connect Analytics subsystem to offload data to a third-party product, you might not want to store a copy of the data in API Connect. You can disable the storage of analytics data in API Connect by modifying the outputs.yml file.

---

## Before you begin

One of the following roles is required to perform this task:

- Administrator
- Topology Administrator
- Owner
- A custom role with the `topology:manage` permission

---

## About this task

The Gateway reports all API events to the API Connect analytics engine, for real-time and historical reporting. By default, the analytics data is stored in API Connect and can be viewed from the dashboards in API Manager.

Important: When you enable or disable the storage of analytics data in API Connect, that change is applied across all provider and consumer organizations in the cloud infrastructure.


There are two ways to disable (or enable) the storage of analytics data in API Connect:

- Disable all analytics collection by unassociating the analytics service from the gateway service. If you disable all analytics collection, there will be no data in either the API Manager Dashboards or third-party offloads. See [Associating an analytics service with a gateway service](#) for how to unassociate an analytics service.
- Disable analytics storage in API Connect by setting the `apic_output_enabled` to `false` in the `outputs.yml` file. This approach lets you retain the third-party offload capability. Do this if you want to [configure third-party output plugins](#) and don't want to view the analytics in API Manager.

## Procedure

---

Complete the following steps to disable the storage of analytics data in API Connect.

1. In the Cloud Manager, click  Topology.
2. Locate the Analytics service that has been associated with the Gateway service, then click the title to open the Edit Analytics Service page.
3. Click the Advanced Analytics Configuration link to load Kibana.
4. In Kibana, click Ingestion in the API Connect section to open configuration file selection page.
5. Open the `outputs.yml` file by selecting its tab.
6. In the `outputs.yml` file, set the value of `apic_output_enabled` to `false`.

The `outputs.yml` file contains some comments plus two settings by default:

```
# Here you should specify whether certain functionalities are enabled or
# disabled. Specifically, the apic_output_enabled flag indicates whether
# data should be stored in APIC Analytics Storage, while the
# offload_output_enabled flag indicates whether data should be offloaded
# to the configured offload output in offload_output.conf.
apic_output_enabled: true
offload_output_enabled: false
```

To disable the Analytics user interface, set `apic_output_enabled` to `false` as in the following example:

```
# Here you should specify whether certain functionalities are enabled or
# disabled. Specifically, the apic_output_enabled flag indicates whether
# data should be stored in APIC Analytics Storage, while the
# offload_output_enabled flag indicates whether data should be offloaded
# to the configured offload output in offload_output.conf.
apic_output_enabled: false
offload_output_enabled: false
```

7. Click Save.

## Results

---

Analytics data will not be stored in API Connect, and will not be available for viewing in the API Manager analytics dashboards. If you configured data offloading, that will not be affected.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting your analytics offload

Perform these checks if the offload fails. Common causes for an offload failure are connection or security-related issues.

When you save the configuration files, their syntax is first validated to verify that it can be applied. However, this does not confirm that other parts of your analytics collection ecosystem are trouble-free, for example, that your endpoints are running, or that the TLS certificates are validated. Errors in these areas may also result in no data output.

To verify issues you will need to examine the logs for one of your Kubernetes analytics ingestion pods. (Note that it may take a few minutes for the pods to pick up changes in the configuration files after edits are applied.)

For information on obtaining logs, refer to [Configuring remote logging for a VMware deployment](#) and [Configuring logging for a Kubernetes deployment](#).

Following are some common checks you can make to troubleshoot the offload:

- Verify that the specified URL or the host and port values for the target system are accurate.
- Verify that the server or cluster of servers in the target system are running and reachable. If applicable, also verify that the connection problems are not caused by firewall settings.
- If you are using Transport Layer Security (TLS) to establish a private and secure communication channel to the target system, investigate whether the issue is caused by SSL certificate errors; for example, untrusted server certificates, certificates that have expired or are not yet valid, or missing intermediate (or chain) certificates.
- If offloading to an Elasticsearch or Kafka target, verify that your configuration settings are valid; for example, the user credentials for authenticating to Elasticsearch or the Kafka topic name.

## Related tasks

- [Creating a TLS Server Profile](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring the audit log to track API calls

Configure auditing in IBM® API Connect to monitor user operations such as who published which product and when.

Audit logging is useful when you want to monitor the create, update, and delete activity in your deployment; for example, who made changes, when, and what they changed.

## What events are audited?

The audit focuses on "who" did "what" and "when", including the request payload and the result of the operation. All create, update, and delete operations are audited. Generally, read operations are only audited if the calling user logged in with an OIDC Provider user registry.

Internal calls and calls between system components are not audited because they are part of the product functionality and not executed by users.

## How are audit events logged?

Audit events are logged by sending the information about API calls (the requests) to one or more endpoints. Audit logging messages are sent as HTTP/HTTPS requests and use the payload format defined by the Cloud Auditing Data Federation (CADF). If the endpoint requires authentication, you can configure a header that will be included in the audit logging request. Audit logging requests are issued once only, on the assumption that every endpoint is available to receive the requests. There is no error checking to ensure that requests are received and stored by the endpoints, and failed requests are not re-sent. If you suspect that audit logging requests are not successfully reaching an endpoint, you can check the server logs on the console, where any error messages are recorded.

There are two modes that control how the auditing is sent to an endpoint: blocking and non-blocking. In non-blocking mode (the default), an audit event is sent to the endpoint and the system does not wait for a response from the endpoint. Non-blocking mode provides the best performance. Blocking mode should be used with caution: when an audit event is sent, the system pauses and waits for a response from the endpoint. Blocking mode has a negative impact on performance.

With the microservices nature of API Connect, any changes to the audit settings can take up to five minutes to be fully honored by all of the microservices supporting it.

## What's included in the audit logging request?

When an API call is made, the system injects a request header to capture information, including the request URL and the payload (with sensitive information redacted), plus a subset of the data returned in the response to that call. The system injects additional information to describe the "who", "what", and "when" of the call.

The payload containing the audit logging record includes information about the user who issued the API call, the date and time of the call, the resource affected by the call, the operation performed, and the payload. To see the most recent set of fields that are included in the payload, refer to the OpenAPI schema for the audit logging request in the Provider API, which you can download at [https://us-south.functions.cloud.ibm.com/api/v1/web/API%20Connect%20Native\\_apic-on-prem/default/index.http](https://us-south.functions.cloud.ibm.com/api/v1/web/API%20Connect%20Native_apic-on-prem/default/index.http).

The following code example shows a sample payload for an audit logging message (IDs and users are not real):

```
{
  "id": "f6fcacb5-e8eb-4e2c-0f67-53b6a792d7f0",
  "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
  "action": "update",
  "outcome": "success",
  "eventType": "activity",
  "reason": {
    "reasonCode": "200",
    "reasonType": "HTTP"
  },
  "eventTime": "2020-02-01T15:18:16.011Z",
  "initiator": {
    "id": "/api/user-registries/38385c9f-7837-583c-01f7-9d8c37a9a80d/10de0db0-27b5-35d7-b4b5-635eabb48b4a/users/24c6cddc-7a7e-5fc6-882d-5317d822048e",
    "name": "admin:default-idp-1/tjwatson",
    "typeURI": "service/security/account/user"
  },
  "target": {
    "id": "0beb6d21-6207-5381-b9a7-cc91a3e82c19",
    "typeURI": "tls_client_profile"
  },
  "observer": {
    "id": "target"
  },
  "requestPath": "/api/orgs/admin/tls-client-profiles/uma-tls/1.0.0",
  "requestData": {
    "url": "/api/orgs/admin/tls-client-profiles/uma-tls/1.0.0",
    "body": {
      "visibility": {
        "type": "public"
      }
    },
    "namespace": "38385c9f-6726-472b-91f7-9d8c37a9a80d",
```

```

    "updated_at": "2020-02-01T15:18:15.924Z"
  },
  "responseData": {
    "id": "0beb6d21-6207-5381-b9a7-cc91a3e82c19",
    "url": "https://my_host.example.com:3003/api/orgs/38385c9f-7837-583c-01f7-9d8c37a9a80d/tls-client-profiles/0beb6d21-6207-5381-b9a7-cc91a3e82c19",
    "name": "uma-tls",
    "version": "1.0.0",
    "title": "Uma TLS Client Profile"
  },
  "attachments": [
    {
      "timestamp": {
        "start": "2020-02-01T15:18:15.802Z",
        "end": "2020-02-01T15:18:16.011Z"
      },
      "cloud_name": "management.example.com",
      "request_id": "f6fcacb5-e8eb-4e2c-0f67-53b6a792d7f0",
      "method": "patch",
      "operation": "tls_client_profile_updateByNameVersion",
      "summary": "Update the TLS Client Profile object by name and version",
      "resource": "tls_client_profile",
      "user": {
        "url": "/api/user-registries/38385c9f-7837-583c-01f7-9d8c37a9a80d/10de0db0-27b5-35d7-b4b5-635eabb48b4a/users/24c6cddc-7a7e-5fc6-882d-5317d822048e",
        "context": "admin",
        "idp_name": "default-idp-1",
        "name": "tjwatson"
      },
      "registration": {
        "url": "/api/cloud/registrations/9d13c715-e3dc-55ef-92f1-1c2651c0a660",
        "type": "toolkit"
      }
    }
  ]
}

```

## Verifying that the endpoint is receiving audit logging messages

By default, auditing operates in `non_block` mode to ensure best performance. In `non_block` mode, each audit logging message is sent immediately without waiting to ensure that the previous message was received. If you suspect that audit logging messages are not being received by the endpoint, download the server logs from the Kubernetes console and examine the `apim_microservices` file for messages related to the connection state. Review the messages to validate that the connection was successful or to further diagnose issues.

You can test the connection to a particular endpoint by issuing a "test-connection" call using either the toolkit or a REST API. A response status of `204` indicates that the call was successful and the endpoint is available. The following example contains the payload from a connection test (IDs and users are not real):

```

{
  "id": "8d8eaaa2-1b46-4dcb-cadf-639d7d1f7f20",
  "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event",
  "action": "create",
  "outcome": "success",
  "eventType": "activity",
  "reason": {
    "reasonCode": "200",
    "reasonType": "HTTP"
  },
  "eventTime": "2020-01-30T15:51:18.254Z",
  "initiator": {
    "name": "admin:default-idp-1/tjwatson",
    "typeURI": "service/security/account/user"
  },
  "target": {
    "id": "test-connection",
    "typeURI": "cloud_setting"
  },
  "observer": {
    "id": "target"
  },
  "requestPath": "/api/cloud/settings/audit-endpoint/test-connection",
  "requestData": {
    "url": "/api/cloud/settings/audit-endpoint/test-connection",
    "body": {
      "message": "hello"
    }
  },
  "responseData": {
    "name": "world"
  },
  "attachments": [
    {
      "timestamp": {
        "start": "2020-01-30T15:51:18.160Z",
        "end": "2020-01-30T15:51:18.254Z"
      },
      "cloud_name": "management.example.com",
      "request_id": "8d8eaaa2-1b46-4dcb-cadf-639d7d1f7f20",
      "method": "post",
      "operation": "cloud_setting_auditEndpointTestConnection",
      "summary": "Test connection using one of the auditing endpoints",
      "resource": "cloud_setting",
      "user": {

```

```

    "id": "/api/user-registries/dd49cebf-d3ba-5fee-92eb-f895328bc9ef/d51fa599-a5f3-50e2-a786-fafcc76daca5/users/7ff5fff8-3d4b-3524-8687-524740580d0b",
    "context": "admin",
    "idp_name": "default-idp-1",
    "name": "tjwatson"
  },
  "registration": {
    "url": "/api/cloud/registrations/734c04f7-0fe2-3ffa-b745-81027a3b8c76",
    "type": "toolkit"
  }
}
]
}

```

Tip: To ensure that audit logging messages are not lost due to an unavailable endpoint, verify each endpoint periodically.

- [Configure the audit settings](#)  
Use either the IBM API Connect Cloud Manager UI or the Toolkit CLI to enable auditing of API calls.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configure the audit settings

Use either the IBM® API Connect Cloud Manager UI or the Toolkit CLI to enable auditing of API calls.

### Before you begin

Configure one or more audit endpoints where logging data can be directed, and validate that every endpoint is available by testing the connection as explained in [Configuring the audit log to track API calls](#).

### About this task


You can configure audit settings using the [Cloud Manager UI](#) or the [Toolkit CLI](#).

By default, auditing is disabled. If you enable auditing, the API Connect management node emits audit events as messages and sends them to the endpoint for storage.

### Using the UI to configure audit settings

Use the Cloud Manager user interface to enable auditing of customer API calls.

#### Procedure

1. In the Cloud Manager, click  Settings > Audit Setting > Edit.
2. Click Enable to enable the audit feature, or Disable to disable it.
3. Select an audit mode:
  - Non-blocking - Default value, offers the best performance. Each request is passed immediately without waiting for a response to the previous request. This is susceptible to errors if a request is dependent on the results of an earlier request whose response is not received before the new request is sent.
  - Blocking - If you select this mode, expect a degradation in performance. Each request is allowed to finish execution before the next request is sent. This mode results in fewer errors when requests are dependent on the responses of earlier operations, but the delayed requests slow the network.
4. Add the values to describe an endpoint (you must specify at least one endpoint):
  - URL - Required, must not be empty. The URL of the server where audit logging messages will be sent. The URL must use either HTTP or HTTPS; TCP is not supported.  
If you previously saved the audit settings with an empty URL and experienced problems, you can edit the URL to correct it.
  - TLS client profile - Optional. The URL of the server where the endpoint's TLS client profile is stored.
  - Header name and Header value - Optional. Multiple headers are allowed (click Add to add another header. For each header, specify the header name and the header value.
5. Optional. To specify another endpoint, click Add at the beginning of the "Endpoints" section.
6. Optional. Verify that you can reach the endpoint by clicking Test.
7. Click Save to save your settings.

#### Results

With the microservices nature of API Connect, any changes to the audit settings can take up to five minutes to be fully honored by all of the microservices supporting it.

#### What to do next

If you suspect that audit logging messages are not being received by the endpoint, see [Verifying that the endpoint is receiving audit logging messages](#) for information on validating the connection.

### Using the CLI to configure audit settings

Use the IBM API Connect toolkit CLI to enable auditing of API calls.

#### About this task



Use the following settings to configure auditing with the CLI toolkit:

- **enable** - Boolean, defaults to **false** (no audit). Set to **true** if you want to enable auditing.
- **mode** - Enumerated, defaults to **non\_block**. The audit mode determines how audit requests are timed when they are sent to the endpoint:
  - **non\_block** - Default value, offers the best performance. Each request is passed immediately without waiting for a response to the previous request. This is susceptible to errors if a request is dependent on the results of an earlier request whose response is not received before the new request is sent.
  - **block** - If you select this mode, expect a degradation in performance. Each request is allowed to finish execution before the next request is sent. This mode results in fewer errors when requests are dependent on the responses of earlier operations, but the delayed requests slow the network.
- **endpoints** - Array of strings. You must specify at least one endpoint where the audit logging messages are sent. For each endpoint, you can optionally include a TLS client profile URL and one or more headers. For example, if the endpoint requires an authorization header, you can include it in the endpoint setting.
  - **endpoint\_url** - String. Required, must not be empty. The URL of the server where audit logging messages will be sent. The URL must use either HTTP or HTTPS; TCP is not supported.  
If you previously saved the audit settings with an empty URL and experienced problems, you can edit the URL to correct it.
  - **tls\_client\_profile\_url** - String. Optional. The URL of the server where the endpoint's TLS client profile is stored.
  - **headers** - String. Optional. Multiple headers are allowed. For each header, specify the header name and the header value using the format shown in the example.

The endpoint setting can be lengthy, so uploading the settings in a file helps to avoid errors. Format the settings as a .yaml file (you can choose the file name) and upload the file using the toolkit CLI. The following example enables auditing in non-blocking mode with three endpoints. The first endpoint setting omits both the TLS client profile and the headers, the second endpoint setting includes a TLS client profile and one header, and the third endpoint setting omits the TLS profile but includes two headers.

```
audit_setting:
  enabled: true
  mode: non_block
  endpoints:
    - endpoint:
      endpoint: >-
      https://mutt-audit-endpoint.example_1.com/auditing
    - endpoint:
      endpoint: >-
      https://auditing-endpoint.example_1.com/inputs/64e38c02-acd9-6530-76f8-dd0f336d7b11
      tls_client_profile_url: >-
      https://security.tls_example.com/api/orgs/d8eac736-b4e2-3aa6-c7ff-978a4cd19d0b/tls-client-profiles/19d0f087-b4e2-3aa6-926b-9f6d51d028c3
      headers:
        authorization: Basic c3Bvb246c21tb25rYXRpZXVtYQ==
    - endpoint:
      endpoint: >-
      https://auditing-endpoint.example_2.com/inputs/64e38c02-acd9-6530-76f8-dd0f336d7b11
      headers:
        authorization: Basic c3Bvb246c21tb24=
        x-header: ForYourEyeOnly
```

## Procedure

1. Create a YAML file with the audit settings as shown in the example.  
For information on using input files with the toolkit CLI, see [Reading input from the command line](#).
2. Run the following command to log in to a management server as a member of the cloud administration organization:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm realm
```

Cloud settings are stored on a management server and you must log in to the server before you can access the settings with the CLI. For more information on logging in, see [Logging in to a management server](#).

3. Upload the YAML file with the auditing settings by running the following command:

```
apic cloud-settings:update --server mgmt_endpoint_url my_audit_settings.yaml
```

4. Run the following command to log out of the management server:

```
apic logout --server mgmt_endpoint_url
```

## Results

With the microservices nature of API Connect, any changes to the audit settings can take up to five minutes to be fully honored by all of the microservices supporting it.

## What to do next

If you suspect that audit logging messages are not being received by the endpoint, see [Verifying that the endpoint is receiving audit logging messages](#) for information on validating the connection.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---


# Changing your Cloud Manager password and profile information

You can change your Cloud Manager password and update your profile information.

## Procedure

---

To change your password or profile information, follow these steps:

1. Log in to the Cloud Manager user interface.
2. Click the User icon , and then select My Account.
3. In the Profile section, enter a new email address and First Name and Last Name for the account owner.
4. Click Save.
5. In the Change password section, enter your current password and your new password, and confirm the new password.  
The password must contain six characters at a minimum. It must contain at least three of the following types of characters:
  - Lowercase letters
  - Uppercase letters
  - Numbers
  - Characters from the following list of allowed punctuation and special characters:

```
!
\"
#
$
%
&
\
(
)
*
+
, (comma)
- (dash/hyphen)
. (period)
/
: (colon)
; (semi-colon)
<
=
>
?
@
[
\\
]
^
_ (underscore)
` (back quote/grave accent)
{
| (pipe)
}
~ (tilde)
```

The password cannot contain more than two consecutive repeating characters.

6. Click Save.

## Results

---

Your profile information and password are changed.

## Related concepts

---

- [Configuring and managing your server environment](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Resolving login problems by increasing HTTP header size

You can resolve login problems for the Cloud Manager UI by increasing the maximum HTTP client header size.

### About this task

---

Login attempts to the user interface might fail with error 502 (**Bad Gateway**). This error can occur when logging in from a browser that has a large number of cookies. The error occurs because size of the login request exceeds the maximum HTTP client header size.

The maximum HTTP client header size is limited for security reasons. To workaroud this issue, you can clear the browser cache and cookies, or open an incognito window from the browser, and then retry the login.

Alternatively, you can increase the maximum HTTP client header size. Use caution when increasing the maximum size because larger headers can raise a security risk to your system.

## Procedure

---

1. SSH to your management system or appliance.
2. Enter the following commands:

```
kubectl get deployment -n namespace | grep apim-v2
kubectl edit deployment -n namespace apiconnect-apim-v2-deployment
```

3. In the `env` section of the deployment configuration, change `--max-http-header-size=12000` to a larger value that works for your environment. For example:

```
- name: NODE_OPTIONS
  value: --max-old-space-size=8094 --max-http-header-size=16000
```

4. Save the updated configuration.
5. Run `kubectl get pod -n namespace` a few times until the new `apiconnect-apim-v2` pod is up and running.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Extending the Gateway server behavior

To support your enterprise requirements, you can extend the Gateway servers within IBM® API Connect to provide extra enforcement behavior.

### Before you begin

---

Before you develop your extensions, consider the guidelines in the following topics depending on which type of gateway you are using:

- [Gateway extension guidelines - DataPower Gateway \(v5 compatible\)](#)
- [Gateway extension guidelines - DataPower API Gateway](#)

For information on the different types of gateway, see [API Connect gateway types](#).

### About this task

---

API Connect Gateway servers uses a subset of DataPower® enforcement capabilities. You can supply DataPower Extensions to customize these enforcement capabilities further.

The DataPower extensibility function can be used to perform actions that include schema validation, antivirus scanning, message filtering, authentication, and authorization, token translation, message enrichment, encryption & decryption, digital signing, and validation and message transformation. For more information, see [IBM DataPower Version 7.5 documentation](#).

## Procedure

---

To extend the default enforcement capabilities that are provided in API Connect for the Gateway server, complete the following steps.

1. In your IBM DataPower environment, develop the configuration that you want to add to your Gateway server.
2. Test your enforcement configuration before you add the configuration to the Gateway server.
3. When you complete the IBM DataPower configuration, save the enforcement objects and files as a DataPower exported .zip file.  
The package file is ready to be uploaded to the Gateway server.
4. Copy your exported configuration .zip file to a centralized file system ready for upload to the Gateway server.

### What to do next

---

Upload your extension to the Gateway server; see [Configuring your Gateway server extensions](#).

- [Gateway extension guidelines - DataPower Gateway \(v5 compatible\)](#)  
Before you develop your extensions to the DataPower Gateway (v5 compatible), consider these guidelines.
- [Gateway extension guidelines - DataPower API Gateway](#)  
Before you develop your extensions to the DataPower API Gateway, consider these guidelines.
- [Configuring your Gateway server extensions](#)  
You can add extra IBM DataPower enforcement capabilities to a Gateway service by using API Connect CLI commands to upload a .zip file that defines the required extension behavior, and then enabling the extension in IBM DataPower.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

DataPower Gateway (v5 compatible)

---

## Gateway extension guidelines - DataPower Gateway (v5 compatible)

Before you develop your extensions to the DataPower® Gateway (v5 compatible), consider these guidelines.

Note: These guidelines apply to the DataPower Gateway (v5 compatible). If you are using the DataPower API Gateway, see [Gateway extension guidelines - DataPower API Gateway](#).

For information on the different types of gateway, see [API Connect gateway types](#).

You can download sample gateway extensions from <https://github.com/ibm-apiconnect/dp-extensions>.

You can create DataPower Processing Rules that can extend the enforcement behavior of the IBM® API Connect Gateway server at the following locations:

- **pre-request** extension:
  - Before the Gateway server begins to use the policies in the assembly to process the request.
- **post-request** extension:
  - After the Gateway server processes all of the policies in the assembly up to the proxy policy, if a proxy policy is used.
  - After the Gateway server processes all policies in the assembly, but before any catch logic is processed, if a proxy policy is not used.
- **post-response** extension:
  - After the Gateway server processes all of the remaining policies in the assembly after the proxy policy (including catch logic), but before the response is returned to the client application, if a proxy policy is used.
  - After the Gateway server processes the catch logic, but before the response is returned to the client application, if a proxy policy is not used.
- **post-error** extension:
  - If an error occurs, then before the Gateway server returns the error response to the client application.

To configure the Gateway server to call your extension Processing Rule, you must create an XML file that indicates the extension location and Processing Rule name. For example,

```
<extensions>
<extension location="pre-request">CustomRule1</extension>
<extension location="post-request">CustomRule2</extension>
<extension location="post-response">CustomRule3</extension>
<extension location="post-error">CustomRule4</extension>
</extensions>
```

The `<extension>` element entries are optional for any of the locations. Refer to the Gateway server Extension schema for the XML file syntax.

An extension Processing Rule can be applied to a specific organization or to all of the organizations and Catalogs. If the `<extension>` element does not have a tenant attribute, then the `<extension>` element is applied to all of the organizations. The following example shows an `<extension>` element without a tenant attribute.

```
<extension location="post-error">gen_error_handling</extension>
```

To apply an extension Processing Rule to a specific organization, enter the organization name as the value for the tenant attribute in the `<extension>` element. The following example shows an `<extension>` element with a tenant attribute value of `organization1`.

```
<extension location="post-error" tenant="organization1">organization1_error_handling</extension>
```

If you want to exclude a specific organization in an extension Processing Rule, enter the organization name as the value for the tenant attribute in an empty `<extension>` element. Add the empty `<extension>` element immediately after the custom rule `<extension>` element that you want to exclude the organization from. The following example shows an empty `<extension>` element with a tenant attribute value of `organization1`.

```
<extension location="pre-request">a_specified_CustomRule</extension>
<extension location="pre-request" tenant="organization1"></extension>
```

Important: An extension Processing Rule with a specified tenant attribute takes precedence over an `<extension>` element that applies to all of the organizations. Also, only one extension Processing Rule is applied to each extension location.

This XML file must be saved to the following location and included in your DataPower exported configuration .zip file:

```
local:///ext/extensions.xml
```

There can be only one DataPower exported configuration .zip file added to a Gateway server in API Connect.

## DataPower configuration restrictions

All Processing Rules have read access to the INPUT context.

Processing Rules must not rely on context variables that are created by the IBM Gateway server enforcement configuration, because those configuration variables might change in the future.

All Processing Rules except Before Request can transform or alter the message flowing through the Gateway server. Ensure that the Processing Rule returns the desired message output context back to the Gateway server at the end of processing.

To avoid name conflicts, all DataPower configuration object names prefixed with `webapi` are reserved for IBM use.

The following folders cannot be modified:

- `local:///isp/*`
- `local:///gwapi/*`

As a best practice, avoid adding asynchronous actions in your custom Processing Rules because they increase the use of memory per transaction.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Gateway extension guidelines - DataPower API Gateway

Before you develop your extensions to the DataPower® API Gateway, consider these guidelines.

Note: These guidelines apply to the DataPower API Gateway. If you are using the DataPower Gateway (v5 compatible), see [Gateway extension guidelines - DataPower Gateway \(v5 compatible\)](#).

For information on the different types of gateway, see [API Connect gateway types](#).

---

### Introduction

You apply an extension to a DataPower API Gateway by uploading one or more .cfg files that define the required extension behavior. These .cfg files contain DataPower API Gateway CLI commands. For details of the full set of CLI commands that are available, see the [DataPower API Gateway CLI command documentation](#).

You package the .cfg files in a .zip file, together with any additional required files that are referenced from CLI commands; for example, a .json file that contains the OpenAPI definition for an API. The .cfg files are processed in alphanumeric order by file name.

You upload the extension .zip file to an API Connect Gateway service, and enable the extension, as described in [Configuring your Gateway server extensions](#).

---

### Using CLI commands in a Gateway extension

Although you can use any DataPower API Gateway CLI commands in your extension .cfg files, typically you use commands that modify API Connect objects that have previously been deployed to the DataPower API Gateway.

When API Connect objects are deployed to a DataPower API Gateway, .cfg are created on the Gateway that define the configuration of those objects. By examining those .cfg on the Gateway, you can determine the precise names of API Connect objects that you want to modify, and the current values of the properties of those objects. You can access these .cfg files by completing the following steps:

1. Log in to the DataPower administrative user interface; for the Graphical Interface, select WebGUI rather than Blueprint Console.
2. Switch to the DataPower application domain for API Connect if necessary.
3. Under Files and Administration, click File Management.
4. Expand the temporary: folder to locate the .cfg files. The files for API Connect objects are named according to their containing provider organization and Catalog, as follows:

```
20.provider-org-name_catalog-name_collection.cfg
```

The prefix 20 controls the position of the file in the overall processing sequence, which is in alphanumeric order by file name.

CAUTION:

- Do not modify these files directly in the DataPower API Gateway file system, use the gateway extension mechanism described on this page.
- Do not apply gateway extension modifications to configuration in the local:/config-sequence.cfg file.

---

### Example 1 - change the scope of the rate limit for a Plan

Suppose the DataPower API Gateway configuration for a Plan has the following CLI command:

```
api-plan myorg_sandbox_financial-services_1.0.0_basic
```

The **api-plan** command has the following syntax:

```
api-plan plan_name
```

and creates a Plan of the specified name, or modifies a Plan of the same name if it already exists. By using the specified name, you can add **api-plan** commands in your Gateway extension .cfg files to modify the Plan.

Note: The names of API Connect objects in the DataPower API Gateway configuration are derived from their configuration in API Connect. For example, a Plan name has the following structure:

```
provider-org-name_catalog-name_product-name_product-version_plan-name
```

where:

- *provider-org-name* is the name of the provider organization.
- *catalog-name* is the name of the Catalog in that provider organization.
- *product-name* is the name of the Product that contains the Plan.
- *product-version* is the version of the Product.
- *plan-name* is the name of the Plan.

The default scope to which a Plan rate limit applies is per application. To change this setting on the Plan in this example, so that the rate limit scope is per client ID, add the following command to a .cfg in your Gateway extension:

```
api-plan myorg_sandbox_financial-services_1.0.0_basic
rate-limit-scope per-client-id
exit
```

For more information on configuring Plans by using DataPower API Gateway CLI commands, see [API Plan commands](#).

---

### Example 2 - add a path to an API definition

The .cfg file in this example performs the following actions:

1. Uses the **top** and **configure terminal** commands to reset CLI processing.
2. Uses the **api-operation** command as follows:

- a. Creates a new operation called `bank_branches_get_operation`.
- b. Uses the `reset` command to set all properties to their default values.
- c. Sets the operation method to `GET`.
3. Uses the `api-path` command as follows:
  - a. Creates a new path called `bank_branches_path`.
  - b. Uses the `reset` command to set all properties to their default values.
  - c. Sets the path URL segment to `/details`.
  - d. Adds the previously created operation `bank_branches_get_operation` to the path.
4. Uses the `api-definition` command as follows:
  - a. Modifies the existing API definition called `myorg_sandbox_myapi_1.0.0`.  
 Note: The name of the API definition was derived from its configuration in API Connect when it was published to the DataPower API Gateway, and has the following structure:

```
provider-org-name_catalog-name_api-name_api-version
```

where:

- `provider-org-name` is the name of the provider organization.
- `catalog-name` is the name of the Catalog in that provider organization.
- `api-name` is the name of the API definition in API Connect.
- `api-version` is the version of the API.

- b. Adds the previously created path `bank_branches_path` to the API definition.

Note: The `reset` command is not used here because that would reset all the current settings on the `myorg_sandbox_myapi_1.0.0` API definition, whereas the requirement is to add a new path to the existing configuration of the API definition.

The `.cfg` file is as follows:

```
top; configure terminal

api-operation bank_branches_get_operation
  reset
  method GET
exit

api-path bank_branches_path
  reset
  path "/details"
  operation bank_branches_get_operation
exit

api-definition myorg_sandbox_myapi_1.0.0
  path bank_branches_path
exit
```

For more information on the DataPower API Gateway CLI commands used in this example, see the following pages:

- [Initial login and common commands](#)
- [API Operation commands](#)
- [API Path commands](#)
- [API Definition commands](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring your Gateway server extensions

You can add extra IBM® DataPower® enforcement capabilities to a Gateway service by using API Connect CLI commands to upload a `.zip` file that defines the required extension behavior, and then enabling the extension in IBM DataPower.

### About this task

You upload Gateway server extensions by using the developer toolkit CLI.

### Procedure

1. Log in to your API Connect Management server as an administrator, by using the following command:

```
apic login --server mgmt_endpoint_url --username admin_user_ID --password admin_password --realm admin/identity_provider
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL.
- `admin_user_ID` is the user ID of your administrator account, and is the same as the user ID that you use to log in to the Cloud Manager user interface.
- `admin_password` is the password for your administrator account.
- `identity_provider` is the identity provider that is used to authenticate administrative users.

For example:

```
apic login --server platform-api.myserver.com --username admin --password password --realm admin/myldap
```

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

2. Upload the Gateway extension .zip file by using the following command:

```
apic gateway-extensions:create extension_zip_file --scope org --org admin --gateway-service gateway_service --availability-zone availability-zone --server mgmt_endpoint_url
```

where:

- `extension_zip_file` is the Gateway extension .zip file that you want to upload.
- `gateway_service` is the name of the Gateway service that you want add the extension to.
- `availability-zone` is the name of the availability zone that contains the Gateway service.
- `mgmt_endpoint_url` is the platform API endpoint URL.

For example:

```
apic gateway-extensions:create myextension.zip --scope org --org admin --gateway-service mygatewayservice --availability-zone availability-zone-default --server platform-api.myserver.com
```

This example uses the default supplied availability zone name of `availability-zone-default`, which will be the required value if you have not configured your own availability zones. To check the names of the currently configured availability zones, use the following command:

```
apic availability-zones:list --org admin --server mgmt_endpoint_url
```

For details on configuring availability zones, see [Creating an Availability Zone](#).

To check the names of the currently configured gateway services, use the following command:

```
apic gateway-services:list --org admin --availability-zone availability-zone --server mgmt_endpoint_url
```

You can confirm that the extension has been added to the Gateway service by using the `gateway-extensions:get` command; for example:

```
apic gateway-extensions:get --scope org --org admin --gateway-service mygatewayservice --availability-zone myavailabilityzone --server platform-api.myserver.com --output -
```

(the parameter setting `--output -` writes the details of the Gateway extension object to the command window. You can specify the name of an existing folder to have the details written to a `.yaml` file in that folder.)

For reference details of the `apic`

`gateway-extensions` commands, see [apic gateway-extensions](#).

Note: You cannot upload more than one Gateway extension .zip file to the same Gateway service. If you want add further extensions later, update the original .zip file, then use the `apic`

`gateway-extensions:update` command to replace the previous gateway extensions file with the revised one; for example:

```
apic gateway-extensions:update mynewextension.zip --scope org --org admin --gateway-service mygatewayservice --availability-zone myavailabilityzone --server platform-api.myserver.com
```

3. Apply the extension to each server by restarting the API Connect gateway service object; complete the following steps on **each** Gateway server in the Gateway service:

Note: In Kubernetes environments, the following steps can be accomplished by issuing `kubect1`

`delete pod` on **each** gateway pod. Be sure to wait until each pod gets back to the ready state before deleting the next gateway pod.

- Remove the Gateway server from the load balancing group.
- Disable, then enable the API Connect gateway service object. You can do this by using either the DataPower administrative user interface in a web browser, or by using the DataPower CLI.
  - By using the administrative user interface:
    - Log in to the DataPower administrative user interface; for the Graphical Interface, select WebGUI rather than Blueprint Console.
    - Switch to the DataPower application domain for API Connect if necessary.
    - Search for `API Connect Gateway Service`.
    - Set the Administrative State to disabled.
    - Apply the changes.
    - Set the Administrative State to enabled.
    - Apply the changes.
  - By using the CLI:
    - Enter configuration mode by entering the command `configure`.
    - Navigate to the DataPower application domain for API Connect by entering the command `switch api_connect_domain_name`, where `api_connect_domain_name` is the name of your API Connect application domain.
    - Disable the API Connect gateway service object by entering the following command:

```
apic-gw-service; admin-state disabled; exit
```
    - Enable the API Connect gateway service object by entering the following command:

```
apic-gw-service; admin-state enabled; exit
```
- Ensure that the gateway service object initialization has completed.
- Re-add the Gateway server to the load balancing group.

## Results

The extension is uploaded and applied to each of the servers in the Gateway service, and the associated enforcement capabilities are applied to all incoming API resource requests.

Warning: DataPower Gateway (v5 compatible) only If you upload an extension to the DataPower Gateway (v5 compatible), a status file called `extension_import_response.xml` is written to the `local:/ext` folder in the gateway file system. Do not remove this file, otherwise if your gateway extension is subsequently removed then the referenced objects cannot be reverted to their original state automatically, and you will therefore have to either complete these clean up tasks manually, or re-add the gateway extension to regenerate the `extension_import_response.xml` file and then remove the gateway extension again. If the `extension_import_response.xml` exists and there is a cleanup failure for any object, a failure file is written to the `local:/ext` folder, with the failure details.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Cloud Manager Tutorials

Tutorials for using the Cloud Manager user interface in IBM® API Connect.

## Prerequisites

---

To complete the following tutorials you must be the Cloud Administrator or Cloud Owner of an API Connect instance.

## Overview

---

These tutorials guide you through common tasks performed to set up and maintain the configuration of a cloud.

- [Tutorial: Configuring the Cloud](#)  
This tutorial shows you how to create a basic cloud configuration, with available gateway, analytics and developer portal services.
- [Tutorial: Creating a Provider Organization](#)  
This tutorial shows you how to create a Provider Organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Configuring the Cloud

This tutorial shows you how to create a basic cloud configuration, with available gateway, analytics and developer portal services.

### Before You Begin

---

This task can be completed by users who are assigned one of the following roles:

- Cloud Owner
- Cloud Administrator

You must run the Install Assist installation program to configure the Default Availability Zone that contains the Management service. See [Installing API Connect into a Kubernetes environment](#) and [Deploying the Management OVA file](#).

You will need the following information to complete this tutorial.

- The IP address or FQDN of the gateway, and the port assigned to accept API requests from clients.
- The IP address or FQDN of the gateway, and the port assigned to communicate with the API Management server. This cannot be the same as the port for API requests.
- The FQDN of the Developer Portal service to accept requests from clients. This value was set using the InstallAssist command:

```
apicup subsys set ptl portal-www <portal>.<hostname>.<domainname>
```

- The FQDN of the Developer Portal to communicate with the API Management server. This value was set using the InstallAssist command:

```
apicup subsys set ptl portal-admin <admin>.<hostname>.<domainname>
```

- The FQDN of the Analytics service to accept requests from clients. This value was set using the InstallAssist command:

```
apicup subsys set analyt analytics-client <ac>.<hostname>.<domainname>
```

- The address, port and optional login credentials for an SMTP email server.

Note: The port for the `portal-www`, `portal-admin`, and `analytics-client` services is **443**, and isn't configurable.

---

### About this tutorial

In this tutorial you are going to complete the following lessons:

- [Initial Cloud Manager Console login](#)
- [Configure an Email Server](#)
- [Register a Gateway Service](#)
- [Register a Portal Service](#)
- [Register an Analytics Service](#)
- [Configure a Default Gateway Service](#)

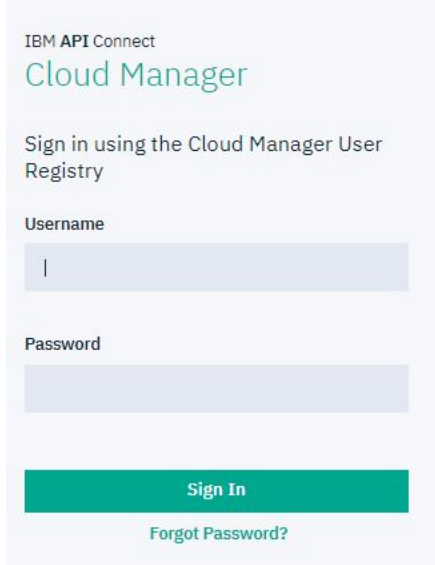


## Initial Cloud Manager Console login

---

Take the following steps to log in to the Cloud Manager user interface for the first time.

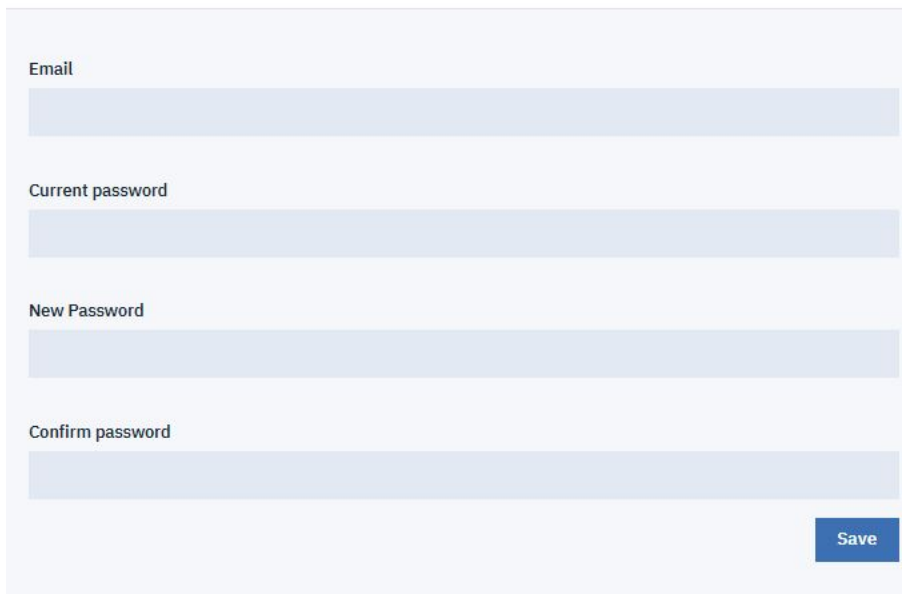
1. In a web browser, enter the management service URL. For example, `https://ManagementService.domain/admin` where `ManagementService.domain` is the fully qualified host name or IP address of the Management service.



The screenshot shows the login page for IBM API Connect Cloud Manager. At the top, it says "IBM API Connect" and "Cloud Manager" in a large green font. Below that, it says "Sign in using the Cloud Manager User Registry". There are two input fields: "Username" and "Password". Below the password field is a green "Sign In" button and a link for "Forgot Password?".

2. Enter the Cloud Administrator user name and password. The default values are `admin` for the user name and `7iron-hide` for the password.
3. You are immediately required to change the admin password, as well as provide an email address for the cloud administrator. Enter the necessary information.  
Note: If you forget your password and request a password reset, the notification email is sent to this email address. This action will use the email server set in the Notifications section of the cloud Settings. This tutorial shows you how to set this configuration.

### Change Password




The screenshot shows the "Change Password" form. It has four input fields: "Email", "Current password", "New Password", and "Confirm password". At the bottom right, there is a blue "Save" button.

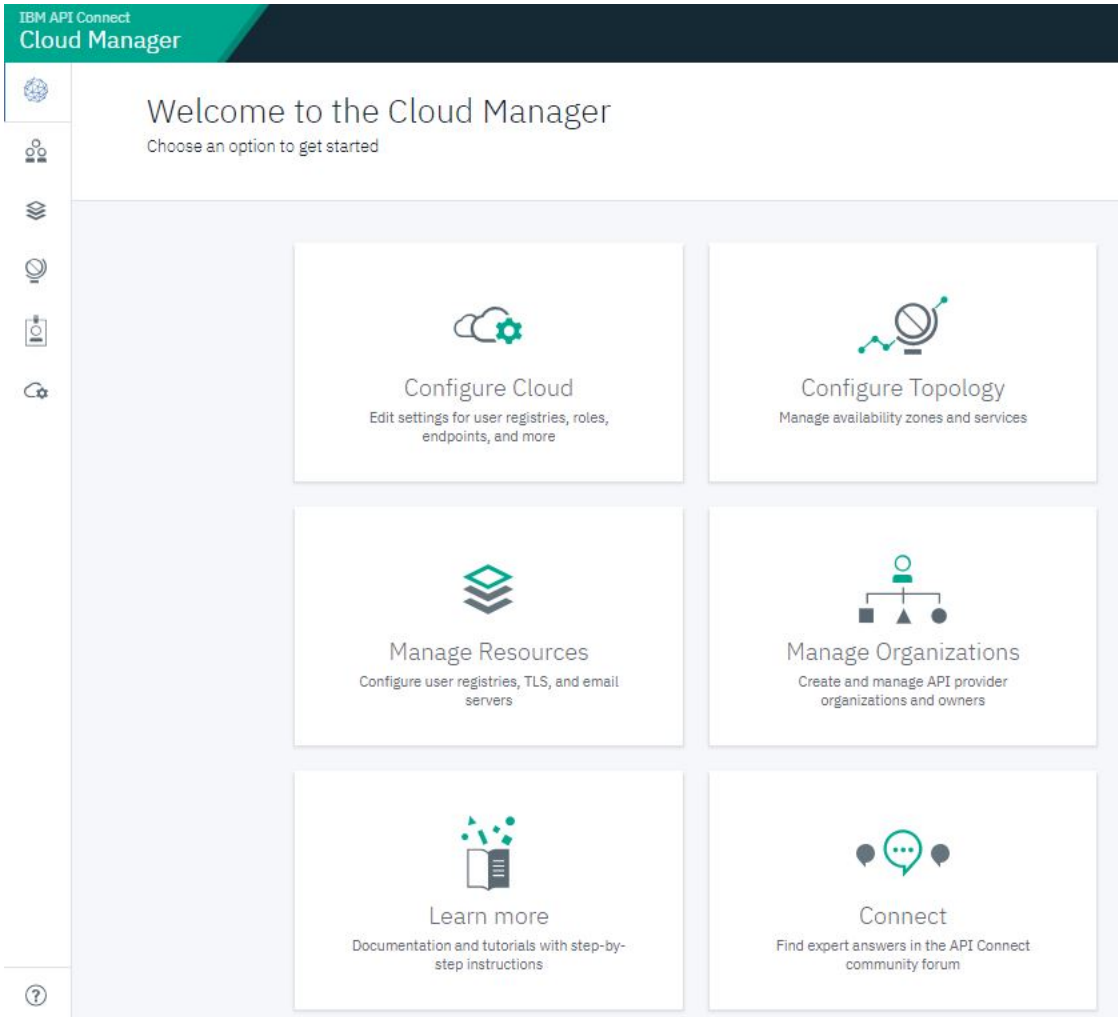
4. Click Save.
5. Log in using the new password.

## Configure an Email Server

---

Take the following steps to configure an email server. This configuration allows you to recover the admin password or receive other important notifications.

1. Click  Resources icon or the Manage Resources tile.



2. Click Notifications.

## Resources


User Registries Create

TITLE	TYPE	VISIBLE TO	
API Manager Local User Registry	Local User Registry	Private	⋮
Cloud Manager Local User Registry	Local User Registry	Private	⋮

3. Click Create.

User Registries Create

Notifications

TITLE	MAIL SERVER
 No items found	

- 4. Enter the appropriate values in the fields These values vary depending on your site.
- 5. Click Save.

## Create Email Server

### Email Server Configuration

**Title**  
Sendgrid

**Name**  
sendgrid

**Address**  
smtp.sendgrid.net

**Port**  
587

**Authenticate User (optional)**

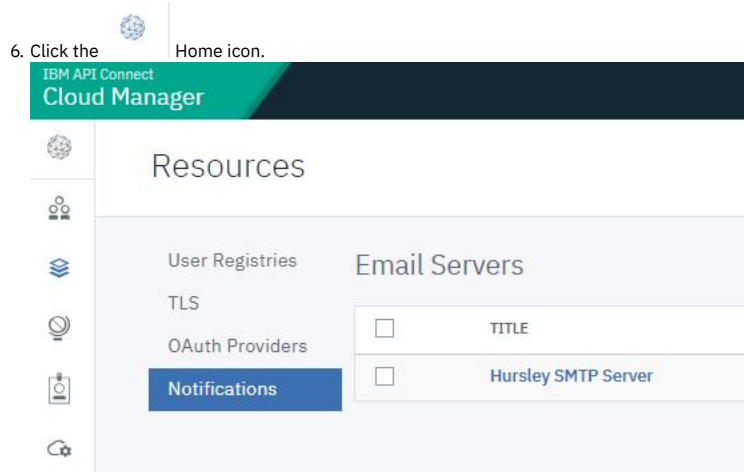
**Authenticate Password (optional)**

**TLS Client Profile (optional)**  
Default TLS client profile

**Test Connection** Test email

Send a test email to confirm that the email server is properly configured.

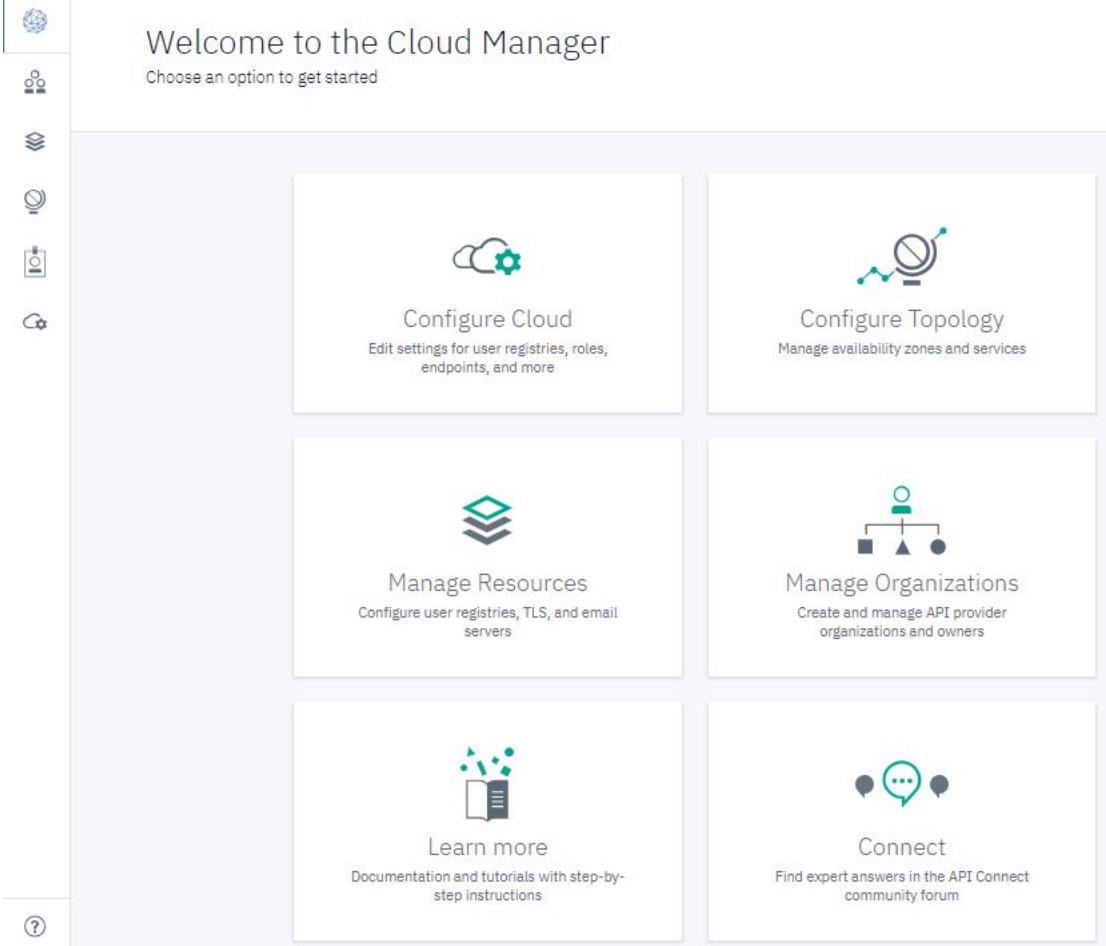
Cancel Save



## Register a Gateway Service

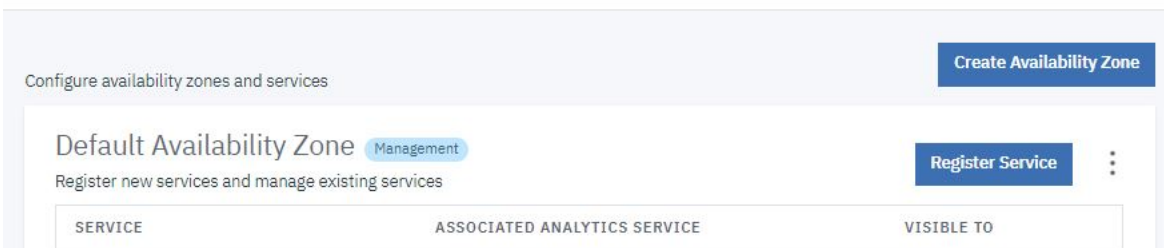
Take the following steps to register a gateway service.

1. Click the Configure Topology tile.



2. Click Register Service.

## Topology



3. Click DataPower Gateway (v5 compatible).

## Select Service Type

**DataPower API Gateway**  
Configure an DataPower API gateway service for securing and enforcing APIs

**DataPower Gateway (Classic)**  
Configure a DataPower gateway service for securing and enforcing APIs

**Portal**  
Configure a developer portal service for API consumers

**Analytics**  
Configure an analytics service to collect API call data

Cancel

#### 4. Take the following steps.

- Enter `gateway_service` in the Title field.
- In the Management Endpoint section, enter the URL of the address and port assigned to the management endpoint in the Endpoint field. This is the port used by the API Management server to connect to the gateway.
- Leave the remaining values as given to set TLS profiles.

## Configure DataPower Gateway Service

**Gateway Details**

**Title**  
gateway\_service

**Name**  
gateway-service

**Summary (optional)**

**Management Endpoint**

**Endpoint**  
https://gateway.example.com:3000

**TLS Client Profile (optional)**  
Default TLS client profile

- In the API Invocation Endpoint section, enter the URL of the gateway address and port assigned to accept API requests from clients in the API Endpoint Base field.
- Do not change the defaults in the Server Name Indication (SNI) fields. Note that your topology may require specific values in these fields.
- Optionally enter a hex value in the OAuth Shared Secret field. This must be a 64-bit hex value that begins with "0x". Providing a shared secret value here enables gateways in a cluster to read the OAuth tokens generated by any member of the cluster using the same secret.
- Click Save.

## API Invocation Endpoint

**API Endpoint Base**

https://gwv5c.example.com:443

**Server Name Indication (SNI)** Add

HOST NAME	TLS SERVER PROFILE	ORDER	DELETE
*	Default TLS server profile		

**OAuth Shared Secret (optional)**

0x

Cancel Save

## Register a Portal Service

---

Take the following steps to register a portal service.

1. Click Register Service.
2. Click Portal.
3. Take the following steps.
  - a. Enter `portal_service` in the Title field.
  - b. In the Management Endpoint section, enter the URL of the portal address and port assigned to communicate with the management server in the Endpoint field.
  - c. Enter the URL of the portal address and port assigned to accept requests from clients in the Portal Website URL field.
  - d. Use the reconfigured profile in the TLS Client Profile field.
  - e. Click Save.

## Configure Portal Service

**Title**  
portal\_service

**Name**  
portal-service

**Summary (optional)**

---

**Management Endpoint**

**Endpoint**  
https://portalsvc.example.com:9443

**TLS Client Profile (optional)**  
Portal Director TLS client profile

---

**Portal Website URL**  
https://portalsvc.example.com:443

## Register an Analytics Service

---

Take the following steps to register an analytics service.

1. Click Register Service
2. Click Analytics.
3. Take the following steps.
  - a. Enter `analytics_service` in the Title field.
  - b. In the Management Endpoint section, enter the URL of the analytics server address and port assigned to accept requests from clients in the Endpoint field.
  - c. Select `Analytics client TLS client profile` in the TLS Client Profile field.
  - d. Click Save.

## Configure Analytics Service

### Analytics Details

**Title**  
analytics\_service

**Name**  
analytics-service

**Summary (optional)**

### Management Endpoint

**Endpoint**  
https://analytics.example.com:9443

**TLS Client Profile (optional)**  
Analytics client TLS client profile

4. Click Associate Analytics Service corresponding to the gateway service listing.

### Default Availability Zone Management

Register new services and manage existing services

SERVICE	ASSOCIATED ANALYTICS SERVICE	VISIBLE TO
gateway_service	<a href="#">Associate Analytics service</a>	Public
analytics_service		Public
portal_service		Public

5. Select analytics\_service\_1. Click Associate.

### Gateway Service

gateway\_service

### Analytics Service

Select the analytics service you would like to associate with the gateway service

ANALYTICS	AVAILABILITY ZONE
<input checked="" type="checkbox"/> analytics_service	Default Availability Zone

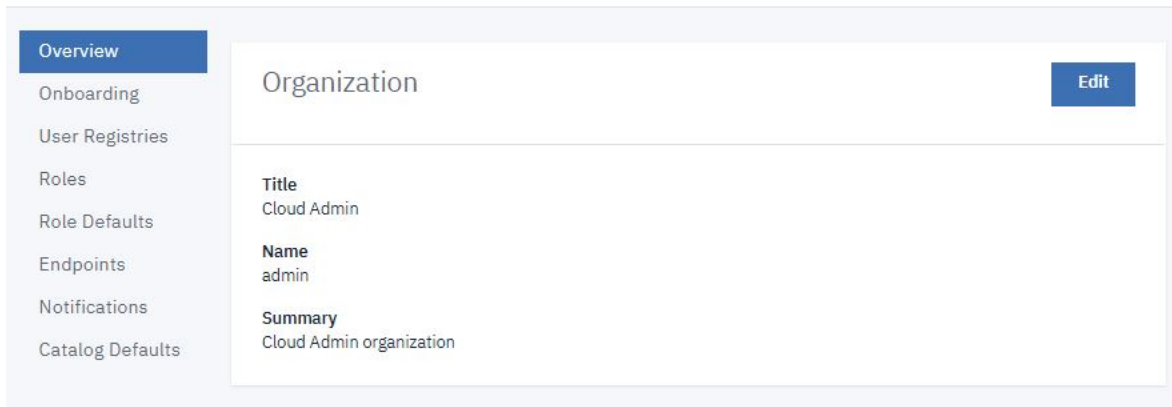


## Configure a Default Gateway Service

Take the following steps to configure a default gateway service for every catalog in the cloud.

1. Click  Settings icon .
2. Click Catalog Defaults.

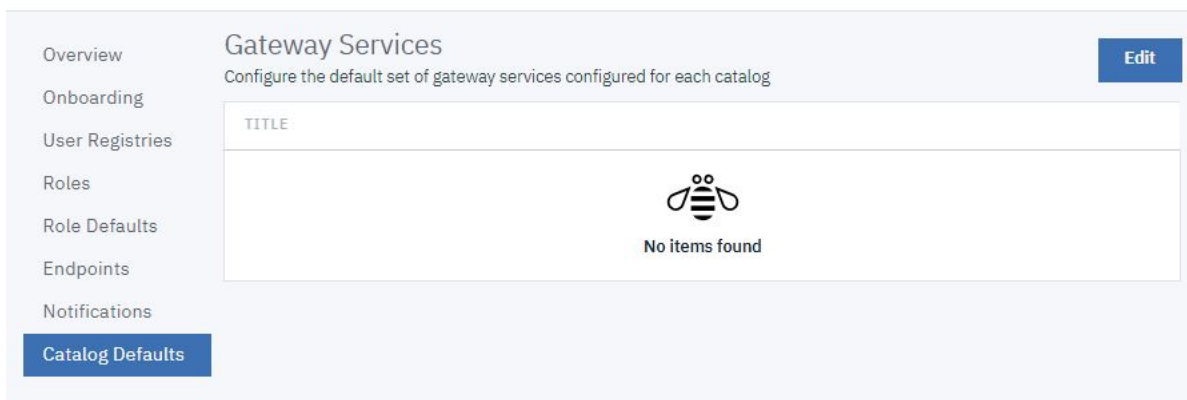
### Settings



The screenshot shows the 'Settings' page with a sidebar on the left containing the following menu items: Overview (selected), Onboarding, User Registries, Roles, Role Defaults, Endpoints, Notifications, and Catalog Defaults. The main content area is titled 'Organization' and includes an 'Edit' button in the top right corner. Below the title, there are three sections: 'Title' with the value 'Cloud Admin', 'Name' with the value 'admin', and 'Summary' with the value 'Cloud Admin organization'.

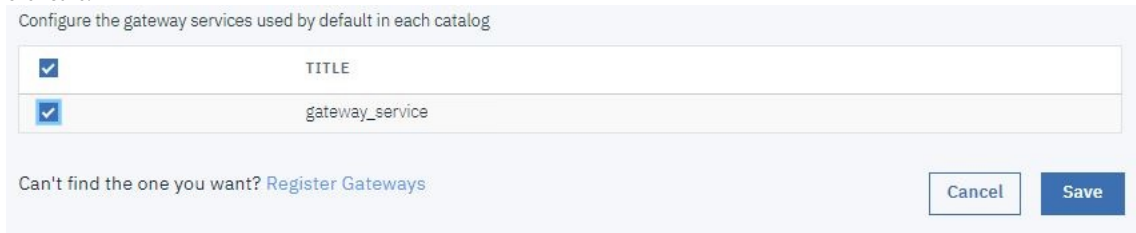
3. Click Edit.

### Settings



The screenshot shows the 'Settings' page with the 'Catalog Defaults' menu item selected in the sidebar. The main content area is titled 'Gateway Services' and includes an 'Edit' button in the top right corner. Below the title, there is a subtitle: 'Configure the default set of gateway services configured for each catalog'. A search bar labeled 'TITLE' is present. Below the search bar, there is a large area with a bee icon and the text 'No items found'.

4. Select an available gateway service.
5. Click Save.



The screenshot shows a dialog box titled 'Configure the gateway services used by default in each catalog'. It contains a table with two rows, each with a checked checkbox and a 'TITLE' column. The first row is empty, and the second row contains the text 'gateway\_service'. Below the table, there is a link: 'Can't find the one you want? [Register Gateways](#)'. At the bottom right, there are two buttons: 'Cancel' and 'Save'.

## What you did in this tutorial

- Set new Cloud Manager Console login password
- Configured a Gateway Service
- Configured a Developer Portal Service
- Configured an Analytics Service
- Configured a Notifications Email Server
- Configured a Catalog Default Gateway Service

## Related information

- [All tutorials](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Tutorial: Creating a Provider Organization

This tutorial shows you how to create a Provider Organization.

## Before You Begin

This task can be completed by users who are assigned one of the following roles:

- Cloud Owner
- Cloud Administrator

You must also complete the following tasks before beginning:

- [Configuring an email server for notifications](#)
- [Setting up notifications](#)

Note that an email server must be configured and an active email server must be selected before a provider organization account can be created.

## About this tutorial

In this tutorial you are going to complete the following lessons:

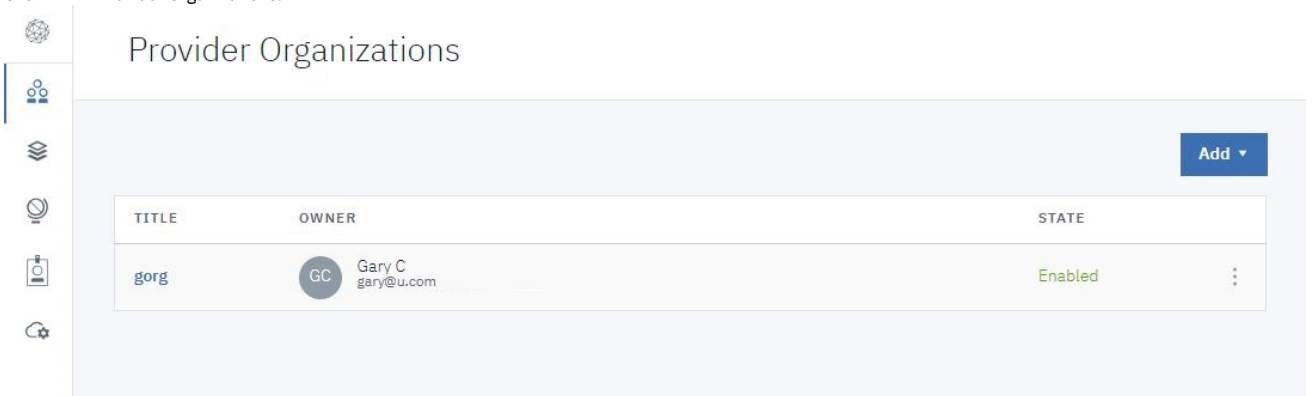
- [Create a Provider Organization](#)

## Create a Provider Organization

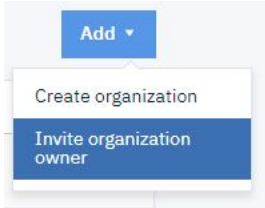
Take the following steps to create a new Provider Organization.

1. Log in to the Cloud Manager.

2. Click  Provider Organizations.

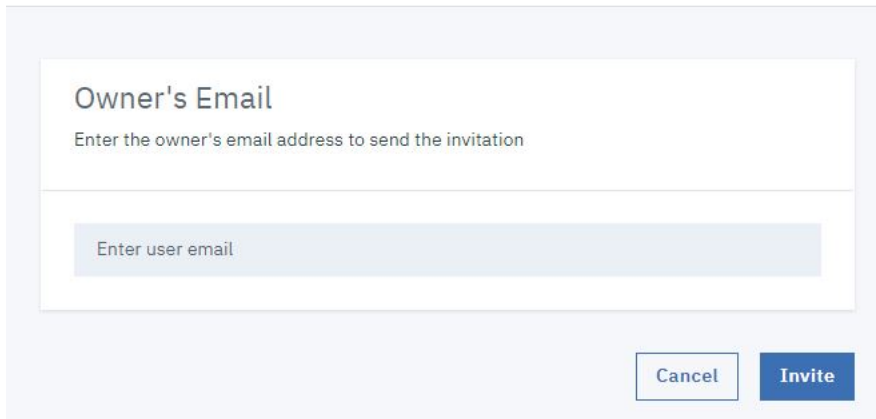


3. Click Add > Invite organization owner.



4. Enter the email address of the organization owner in the Owner's Email field.

## Invite Organization Owner

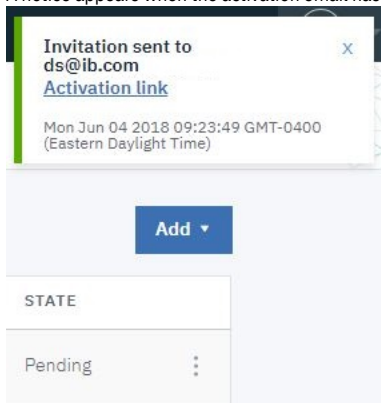


**Owner's Email**  
Enter the owner's email address to send the invitation

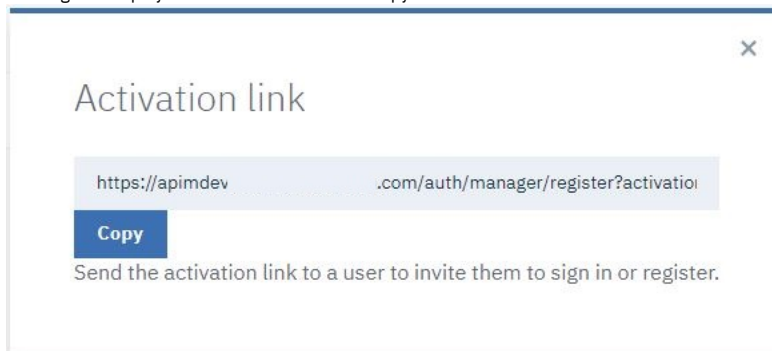
Enter user email

Cancel Invite

5. Click Invite.
6. A notice appears when the activation email has been sent. Click Activation link..



7. A dialog box displays the activation link. Click Copy.



8. Open a new window in your browser. Paste the copied activation link into the Location bar. Optionally, you can click the activation link in the invitation email to open the registration form.
9. Complete fields shown in the form to sign up with the API Manager User Registry. Click Sign up.

IBM API Connect  
API Manager

Sign up with API Manager User Registry

New organization title

Tutorial

Username

tutor

Email

ds@ib.com

First name

D

Last name

S

Password

\*\*\*\*\*

Confirm password

\*\*\*\*\*

[Sign up](#)

Already have an account? [Sign in](#)

10. You see a confirmation of registration. Click Sign in.

IBM API Connect  
API Manager

**Registration completed successfully**

Congratulations, you are now registered.

**Next step**


Log in with your credentials to work in the API Connect cloud.

- Manage your organization
- Manage members for your organization
- Log in with the credentials you registered with


Click the button below to login

[Sign In](#)

11. Enter the username and password you just created. Click Sign in to begin creating APIs and products.

12. Return to the original window of your browser. Click  Provider Organizations to refresh the list. Your new organization is listed.

## Provider Organizations

TITLE	OWNER	STATE
Tutorial	 DS ds@ib.com	Enabled

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a new Provider Organization.

## Related information

- [Importing an API](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Developing your APIs and applications

You develop APIs and LoopBack® applications by using the IBM® API Connect developer toolkit.

The API Connect developer toolkit provides both the API Manager user interface and a command line interface that you can use to develop APIs and LoopBack applications and publish them to API Connect.

You publish APIs by including them in a Product and then publishing the Product. You define your APIs and Products by creating and validating YAML definition files in your local file system. You can then interact with API Connect by using either the API Manager or the toolkit commands.

The developer toolkit is described in detail in the following subsections:

- [Working with the toolkit](#)  
You install the developer toolkit into a Node.js command line environment. You can then use the toolkit commands to interact with IBM API Connect and publish APIs that you have defined in your file system.
- [\[Technical Preview\] Searching for items in API Manager](#)  
Use the search feature in IBM API Connect API Manager to easily locate items such as APIs, Catalogs, applications, and subscriptions.
- [Working with API definitions](#)  
An API definition specifies the complete configuration for an API. You can create and configure your APIs either by using either the API Designer UI application, or by using the browser based API Manager UI.
- [Configuring API security](#)  
You configure security for an API by creating one or more security definitions by using IBM API Connect that specify various aspects of security configuration. You

then select which definitions you want to apply to your API, and to the operations in your API.

- [Working with Products](#)

In IBM API Connect, Plans and APIs are grouped together in Products, with which you can manage the availability and visibility of APIs and Plans.

- [Creating and validating API and Product definitions by using the CLI](#)

The developer toolkit of IBM API Connect provides a command line interface that you can use to create and publish API and Product definitions, and also to validate YAML or JSON definitions.

- [Working with global policies](#)

Use global policies to configure policy assemblies that are called just before, or just after, each of your API assemblies is called. You can upload global policies into each of the gateway services in your Catalogs, and then designate, for each gateway service, which global policy is to be called before an API assembly is called, and which one is to be called after. The designated global policies are applied to all the APIs that are deployed to the associated gateway service.

- [Reference](#)

Reference information for developing your APIs in API Connect, including context variables, and template variables.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Working with the toolkit

You install the developer toolkit into a Node.js command line environment. You can then use the toolkit commands to interact with IBM® API Connect and publish APIs that you have defined in your file system.

The following subsections describe how to install and use the toolkit:

- [Installing the toolkit](#)

You can install the toolkit that provides CLI commands, and the API Designer user interface, for IBM API Connect.

- [Logging in from API Designer](#)

To log in from the API Designer user interface for IBM API Connect, you will need the host name of the management server and the user name and password associated with your API Manager account.

- [Using the developer toolkit command-line tool](#)

The IBM API Connect developer toolkit provides a command-line tool, `apic`, that you can use to perform all API Connect tasks. You can also use the command-line tool to script tasks such as continuous integration and delivery.

- [Viewing application performance metrics](#)

IBM API Connect provides two ways to view application performance metrics for Node.js applications running locally: the built-in metrics dashboard and sending the data to third-party consoles or log files.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Installing the toolkit

You can install the toolkit that provides CLI commands, and the API Designer user interface, for IBM® API Connect.

### About this task

The toolkit is provided as executable files, so no actual installation is necessary, you just need to download the required compressed file and extract the contents.

There are two toolkit options available:

- CLI: provides a command line environment for working with IBM API Connect.
- CLI + LoopBack + Designer: provides a command line environment for working with IBM API Connect, including LoopBack® support, and the API Designer user interface.

To install the toolkit, download the compressed file that is appropriate for your chosen toolkit option and platform, then extract the contents to a chosen location on your local machine. The compressed file contains an executable file for running CLI commands and, if you choose the CLI + LoopBack + Designer option, an executable file for launching the API Designer user interface.

You can download the toolkit compressed file in either of the following ways:

- From IBM Fix Central.
- From either the Cloud Manager or API Manager user interface.

The following table identifies the name of the compressed file that you need to download, depending on your chosen toolkit option and platform:

Table 1. Toolkit file names, by option and platform

Toolkit option	Mac OS X	Linux®	Windows
CLI	toolkit-mac.zip	toolkit-linux.tgz	toolkit-windows.zip
CLI + LoopBack + Designer	toolkit-loopback-designer-mac.zip	toolkit-loopback-designer-linux.tgz	toolkit-loopback-designer-windows.zip

## Procedure

---

To install and run the toolkit, complete the following steps:

1. Download the toolkit compressed file.
  - To download the toolkit from IBM Fix Central, complete the following steps:
    - Open the [IBM Fix Central site](#) in your browser.
    - In the Product selector field, enter `API Connect`, then select IBM API Connect from the drop down list.
    - Select your installed 2018.x.y version from the Installed Version list, then click Continue. If you do not know your installed IBM API Connect version, contact your administrator.
    - In the Text field, enter `toolkit`, then click Continue.
    - Select the required file, as identified in [Table 1](#).  
Note: When you download from IBM Fix Central, the release number is appended to the file name.
    - Click Continue, then follow the instructions to complete the download operation.
  - To download the toolkit from either the Cloud Manager or API Manager user interface, complete the following steps:
    - Cloud Manager or API Manager user interface.
    - Select Help ( ? ) in the navigation.
    - Select Install API Connect CLI & API Designer.
    - Select CLI or CLI + LoopBack + Designer according to your preferred option.
    - Select your platform to download the toolkit compressed file.
    - Close the Install API Connect CLI & API Designer window.

2. Extract the contents of the toolkit compressed file to a folder of your choice.  
The contents of the file depend on the your chosen toolkit option and platform, as follows:

Table 2. Toolkit compressed file contents, by option and platform

Toolkit option	Mac OS X	Linux	Windows
CLI	apic-slim	apic-slim	apic-slim.exe
CLI + LoopBack + Designer	apic api_designer-mac.zip: contains the API Designer user interface application.	apic api_designer-linux	apic.exe api_designer-win.exe

The `apic-slim` or `apic-slim.exe` file is the CLI for IBM API Connect.

The `apic` or `apic.exe` file is the CLI for IBM API Connect including LoopBack support.

Tip: If you are using the CLI option, then if you rename the `apic-slim` file to `apic`, or the `apic-slim.exe` file to `apic.exe`, you can run the CLI commands exactly as documented, copy and paste sample commands from the documentation, and use any command scripts as-is if you later move to the CLI + LoopBack + Designer option.

The `api_designer-platform` file is the API Designer user interface application for the specified platform.

3. Run the CLI.
  - For the Mac OS X or Linux platforms, complete the following steps:
    - Open a terminal instance and navigate to the folder where you extracted the contents of the toolkit compressed file.
    - Make the CLI file an executable file by entering the following command:

```
chmod +x download_name
```

Where `download_name` is the name of the toolkit file that you downloaded, either `apic` or `apic-slim`.

- Run CLI commands as follows:

```
./apic command_name_and_parameters
```

or

```
./apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

- For the Windows platform, complete the following steps:
  - Open a terminal window and navigate to the folder where you extracted the contents of the toolkit compressed file.
  - Run CLI commands as follows:

```
apic command_name_and_parameters
```

or

```
apic-slim command_name_and_parameters
```

For details of the CLI commands, see [apic](#).

Tip: Add the folder location of your CLI file to your PATH variable so that you can run CLI commands from anywhere in your file system.

4. Launch the API Designer user interface by running the application from the location to which you extracted the contents of the toolkit compressed file.

Note:

- To uninstall the API Designer application on a Windows platform with a non Administrator account, complete the following steps:
  - In Windows File Explorer, navigate to the `USER_HOME\AppData\Local\Programs\api-designer` folder.
  - Run the **Uninstall API Designer application** application. Do **not** use the **Add or remove programs** window.
- To uninstall the API Designer application on a Windows platform with an Administrator account, you can either run the **Uninstall API Designer application** application, or you can use the **Add or remove programs** window.

## Results

The IBM API Connect toolkit CLI and, if selected, the API Designer user interface application are installed on your local system.

For information on using the API Designer user interface, see [Developing your APIs and applications](#).

For information on using the toolkit CLI, see [Using the developer toolkit command-line tool](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Logging in from API Designer

To log in from the API Designer user interface for IBM® API Connect, you will need the host name of the management server and the user name and password associated with your API Manager account.

---

### About this task

API Designer provides a secure login to a management server for staging and publishing APIs and Products. You log in to the same host that was configured for API Manager and use your API Manager user name and password. You will need an account on API Manager and membership in a provider organization to access the API Designer.

You can also use the **Explorer** tab to see how your APIs look to a consumer. You can check the descriptions of the different artifacts, and refine any schemas or examples.

Note:

- Configuration changes that are made in API Manager are not available in API Designer concurrently. You will need to log out of API Designer and log back in to access any updates.
- Set `SKIP_IBMID_LOGIN=true` as an environment variable to bypass the IBM ID login screen for API Designer. This applies to users who are working on an internal network, without access to external networks.
- API Designer supports the following OIDC user registries:
  - IBM ID User Registry
  - Google
  - Keycloak
  - Slack
  - Twitter

---

### Procedure

To log in to API Designer, complete the following steps:

1. Download and extract API Designer, as described in [Installing the toolkit](#).
2. Launch the API Designer user interface by running the application from the location where you extracted the contents of the toolkit compressed file.
3. Optional: If you receive the Accept License page, click the URL read the license agreement, then click the button to accept the license. This prompt appears only one time. It does not appear once you have accepted the license on the current system.
4. Optional: If you receive the IBM ID login page, enter your IBM customer ID.  
Note: You won't receive the IBM ID page if you are already authorized, or if you have set the `SKIP_IBMID_LOGIN` environment variable to `true`.
5. Open a directory on your local file system to store the API and Product specifications that you create in API Designer. The system defaults to a previously-accessed directory.  
Note: If you want to use an OpenAPI definition file from elsewhere, downloaded from an external website for example, rather than created by using API Connect, don't copy the file into your API Designer directory.  
Such an API will **not** be visible in the API Designer user interface if it doesn't contain the API Connect specific sections that are required by API Designer.  
  
Instead, copy the OpenAPI definition file to another folder, then import it into API Designer; the import operation adds the necessary API Connect specific sections; see [Adding a REST API by importing an OpenAPI definition file](#).  
  
For more information on the API Connect specific sections in an OpenAPI definition file, see [IBM extensions to the OpenAPI specification](#).
6. If this is the first time you have logged in, establish a cloud connection for API Connect by following these steps:
  - In the dialog window, enter the valid HOST URL of the management server where your API Manager account exists in the form `https://hostname.domain`
  - Click Next.
  - Enter the user name and password for your API Manager account.
  - Select the User Registry from the list.
  - Click Sign In.
7. For subsequent log ins, select the appropriate cloud where you will be working. You can select from any previous cloud connections.
8. When more than one cloud connection is configured, you can switch clouds when developing APIs and Products. See [Switching clouds from API Designer](#).  
Note: If you are using an OIDC user registry that supports single sign-on (SSO), you cannot switch clouds to log in to the same user registry with a different user ID. You must first restart the API Designer application so that you are logged out of the registry, then select the required cloud connection.
9. Click Add Another Cloud to add another cloud connection. Enter the URL, user name and password, and user registry to add the cloud connection.  
Note: If you are using an OIDC user registry that supports single sign-on (SSO), and you are already logged in, you cannot add a new cloud connection to log in to the same user registry with a different user ID. You must first restart the API Designer application so that you are logged out of the registry, then add the new cloud connection.
10. Delete a cloud connection by selecting the trash can icon.
11. API Designer will launch with the organization set to your provider organization in the selected cloud. If you have accounts for multiple provider organizations, it will default to the first organization in alphabetical order.
12. To choose another provider organization, select it from the drop-down list.
  - [Switching clouds from API Designer](#)  
You can switch clouds to work in another cloud in API Designer.
  - [Working offline with API Designer](#)  
You can work offline (without a connection to the cloud) with API Designer to create APIs. Some functionality is missing due to the lack of a cloud connection.

---

### Related information



- [Activating your API Manager user account](#)
- [Creating a provider organization account](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Switching clouds from API Designer

You can switch clouds to work in another cloud in API Designer.

### About this task

---

API Designer allows the flexibility to connect to multiple clouds. For example, your organization may have separate clouds for testing, development, and demos. You select the cloud connection when you log in, but you can also switch clouds when you are developing APIs and Products. Multiple clouds must be configured in order to switch clouds. You must have a valid API Manager account to connect to a cloud.

**Note:** If you are using an OIDC user registry that supports single sign-on (SSO), you cannot switch clouds to log in to the same user registry with a different user ID. You must first restart the API Designer application so that you are logged out of the registry, then select the required cloud connection.

### Procedure

---

To switch clouds for working in API Designer, complete the following steps:

1. Log in to API Designer and select a cloud, or configure a new cloud connection.
2. From the Develop: APIs and Products screen, you can switch to another cloud by selecting it from the Switch Cloud Connection menu.
3. Delete a cloud connection by selecting the trash can icon.

### Related information

---

- [Activating your API Manager user account](#)
- [Creating a provider organization account](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Working offline with API Designer

You can work offline (without a connection to the cloud) with API Designer to create APIs. Some functionality is missing due to the lack of a cloud connection.

### About this task

---

API Designer provides the capability to work in offline mode without a connection to a management server. You can work on APIs with no management server, but some functions will be changed or missing.

The preferred method for working in API Designer is with a Cloud Connection to the management server when logging into API Designer. Working while connected to a management server provides full functionality, with automatic field completion. When working offline, you will need to remember the values to complete the fields that are backfilled from the management server.

Following are the differences when working in offline mode:

- There is no connection to a management server.
- There is no Provider Organization selection.
- Only the Develop APIs and Products tile may be selected. The Learn more and Connect menu tiles are not available.
- When creating an API, the Gateway Type defaults to V5 Gateway and the default policies are loaded.
- Stage and Publish for APIs and Products functions are not available.
- The Sandbox catalog is not available.
- When configuring a Security Definition, there are no selection lists available for the User Registry for a Basic Auth security definition or for the OAuth Provider for an OAuth 2 security definition. You will need to enter the User Registry and OAuth Provider values correctly as text.
- When adding custom Visibility and custom Subscribability values for a product, there is no selection list for the Organization field. You will need to enter the organization name correctly as text.

### Procedure

---

To work in offline mode in API Designer, complete the following steps:

1. Log in to API Designer and open a directory location.
2. To start offline mode, choose Work Offline from the selector in the upper right corner of the Connect to Cloud screen.

3. The Home page appears with the Develop APIs and Products tile. Select Develop APIs and Products to start working in offline mode.
4. To return to online mode, choose Connect to Cloud in the upper right hand corner and select the management server to log in to.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using the developer toolkit command-line tool

The IBM® API Connect developer toolkit provides a command-line tool, **apic**, that you can use to perform all API Connect tasks. You can also use the command-line tool to script tasks such as continuous integration and delivery.

The command line tool is described in detail in the following subtopics. For summary reference material for each of the available commands for the developer toolkit, see [Command-line tool reference for the developer toolkit](#).

- [Overview of the command-line tool](#)  
The IBM API Connect developer toolkit provides commands for cloud administration and API development and management.
- [Logging in to a management server](#)  
You log in to a management server from the command line by using the **apic login** command. The parameters that you supply determine the identity provider that is used to authenticated the supplied user ID, and the scope of the tasks that can be performed after successful log in.
- [Cloud administration commands](#)  
A summary of the cloud administration commands in the IBM API Connect developer toolkit.
- [API development and management commands](#)  
A summary of the core commands in the IBM API Connect developer toolkit.
- [Creating APIs and applications](#)  
You can develop API proxies and API implementations by using the developer toolkit. In the documentation, *API* refers to the API proxy and *application* refers to the API implementation.
- [Publishing APIs and applications](#)  
To publish APIs and applications by using the developer toolkit of IBM API Connect, you set configuration variables to define where you want to publish, log in to the target cloud platform, and then use the appropriate publishing commands.
- [Managing API Products](#)  
Use the **apic products** and **apic apis** commands to manage Products and APIs that have been published to IBM API Connect Catalogs. Use the **--scope space** option to manage Products and APIs that have been published to Spaces within Catalogs.
- [Working with Drafts](#)  
Co-locate your APIs and applications in your local source code control systems to support typical development activities such as commits, branching, merges, continuous integration, and so on. The developer toolkit provides the bridge from the developer's environment to the IBM API Connect runtime services.
- [Reading input from the command line](#)  
If a developer toolkit command takes a file as an input parameter, you can direct the command to read the input directly from the command line rather than supplying a separate file; this can be useful when writing scripts to automate command line operations, for example.
- [Scripting with the toolkit commands](#)  
The IBM API Connect developer toolkit provides commands for cloud administration and API development and management.
- [Working with OpenAPI extensions](#)  
Use the developer toolkit CLI **extensions** commands to manage OpenAPI extensions in your Catalogs or Spaces.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Overview of the command-line tool

The IBM® API Connect developer toolkit provides commands for cloud administration and API development and management.

### Command syntax

---

In general, commands have the following syntax:

```
apic command:sub-command [argument] [options]
```

where

- *command* is the command, usually the thing on which you are acting (for example, product, app, API, Catalogs, and so on).
- *sub-command* is the action to perform.
- *argument* is the argument, where applicable (for example, **catalog**).
- *options* are any number of command-line options, which have the form **--option [value]**. Options also have a short form with a single dash instead of a double dash.

For example, **apic apps:publish --server mgmthost.com**.

For some commands, either the command or sub-command portion is optional. For example:

- **apic products:publish** is equivalent to **apic publish**.
- **apic products:list** is equivalent to **apic products**.

The **create** command has a slightly different syntax:

`apic create:type [options]`

Use the `-h` or `--help` option to view command help.

Note: The language in which the CLI help text, and other command response text, is displayed is determined by the locale setting on your local machine.

## Viewing command line tool help

Display general command-line help information by entering the following command: `apic --help` or `apic -h`. Display help information for a specific `apic` command by entering the following command: `apic command_name --help` or `apic command_name -h`.

## Viewing version information

Display the version of the command-line tool by entering the command: `apic --version`.

## Using configuration variables

You can set the values of commonly-used properties in configuration variables. In general, it's easier and more consistent to set configuration variables instead of specifying them using command-line options.

Note:

You can set a configuration variable locally (the default) to affect only the current LoopBack project, or globally (with command-line option `-g`), to affect all projects. The local value supersedes the global value. You can set local configuration variables only for LoopBack projects. When you set configuration variables for OpenAPI projects, they are always global.

The values of local configuration variables are stored in the `project-root/.apiconnect/config` file, where `project-root` is the project root directory. The values of global configuration variables are stored in the `user-home-dir/.apiconnect/config` file, where `user-home-dir` is the user's home directory.

Use the following commands to work with configuration variables:

- `apic config:get varname` - Get a configuration variable. Use `apic config` to display the values of all local configuration variables or `apic config -g` to display the values of all global configuration variables.
- `apic config:set varname` - Set or update the specified configuration variable.
- `apic config:delete varname` - Delete the specified configuration variable.
- `apic config:clear` - Delete all configuration variables.

You set configuration property values by using the `apic config:set` command. By setting configuration properties (for example `catalog` and `app`), you do not need to supply values for these options when you enter a command.

Additionally, you can use `apic properties` commands to work with configuration properties:

- `apic properties:clear` - Clear the configuration properties.
- `apic properties:create` - Augment the configuration properties with additional name/value pairs.
- `apic properties:delete` - Delete the configuration property.
- `apic properties:get` - Get the configuration property.
- `apic properties:list` - List the configuration properties.
- `apic properties:update` - Update the configuration property.

Note:

If you have an environment variable of the same name as a CLI configuration property then, by default, its value will override the value of the corresponding CLI configuration property for any CLI command at that scope.

For example, if you have defined an environment variable called `SPACE` then, by default, that value will be assumed for the value of the `--space` parameter in the following command, regardless of any `space` configuration property setting:

```
apic products:publish my_product.yaml --scope space
```

To prevent environment variables overriding CLI configuration properties, define an environment variable called `APIC_LOAD_FROM_ENV`, set to the value `false`.

The following table describes the configuration variables:

Table 1. Configuration variables

Variable name	Description	Use instead of (or override with) these flags...
catalog	Default Catalog URI for all commands that manage aspects of a Catalog. Form: <code>mgmt-server/api/catalogs/org-name/catalog-name</code> , where <code>mgmt-server</code> is the management server, <code>org-name</code> is the organization name, and <code>catalog-name</code> is the Catalog name.  Note: The Catalog name <code>apic-dev</code> is reserved for local testing.	<code>--catalog</code> , <code>--organization</code> , <code>--server</code>
cloud	Default management server host name for cloud administration commands. Form: <code>mgmt-server/api/</code> .	<code>--server</code>
consumer	Default URI of an API consumer. Form: <code>mgmt-server/api/consumer-orgs/org-name/catalog-name/consumer-org-name</code> , where <code>mgmt-server</code> is the management server, <code>org-name</code> is the organization name, <code>catalog-name</code> is the Catalog name, and <code>consumer-org-name</code> is the consumer organization name.	<code>--server</code> , <code>--organization</code> , <code>--catalog</code> , <code>--consumer</code>
mode	The default value of the <code>--mode</code> parameter for CLI commands. Set the value to <code>apim</code> or <code>consumer</code> depending on whether you want to run commands on a provider organization or a consumer organization. If you do not set this variable, and do not supply a <code>--mode</code> parameter on a command, the value <code>apim</code> is assumed.	<code>--mode</code>

Variable name	Description	Use instead of (or override with) these flags...
org	Default org URI for all commands that manage organizations. Form: <code>mgmt-server/api/orgs/org-name</code> , where <code>mgmt-server</code> is the management server, <code>org-name</code> is the organization name.  The <code>mgmt-server</code> portion sets the default value of the <code>--server</code> option.  You can append the port number to the server name if it is not the default value of 443.	--organization, --server
space	Default Space URI for all commands that manage aspects of a Space. Form: <code>mgmt-server/api/spaces/org-name/catalog-name/space-name</code> , where <code>mgmt-server</code> is the management server, <code>org-name</code> is the organization name, <code>catalog-name</code> is the Catalog name, and <code>space-name</code> is the Space name.  You can append the port number to the server name if it is not the default value of 443.	--server, --organization, --catalog, --space

To set configuration properties, enter the following command:

```
apic config:set name=value
```

where `name` is the name of the configuration property and `value` the value to assign to it.

For example:

```
apic config:set catalog=https://platform-api.myserver.com/api/catalogs/climbon/sandbox
```

## Scripting commands

It's often helpful to automate a series of `apic` commands in a shell script. Since the `apic` tool first requires you to interactively accept the license, you must first use the following command:

```
apic --accept-license
```

Once you do that, your scripts can run non-interactively.

To disable collection of usage analytics, enter this command:

```
apic --live-help
```

## Related information

- [Command-line tool reference for the developer toolkit](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Logging in to a management server

You log in to a management server from the command line by using the `apic login` command.

The parameters that you supply determine the identity provider that is used to authenticated the supplied user ID, and the scope of the tasks that can be performed after successful log in.

Note: You cannot use the CLI to log in to a management server that uses an OIDC user registry for authentication, because browser access is required for OIDC log in.

To log in to the management server from the command line, enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm realm
```

The parameters for the `apic login` command are as follows:

`mgmt_endpoint_url`

Either the platform API endpoint URL, or the consumer API endpoint URL. Use the platform API endpoint URL if you are logging in as a member of a cloud administration organization or provider organization, and the consumer API endpoint URL if you are logging in as a member of a consumer organization. These endpoint URLs are configured during the installation of API Connect, as described in [Installing the Management subsystem into a Kubernetes environment](#) and [Deploying the Management subsystem in a VMware environment](#). If you have access to the Cloud Manager user interface, you can view the configured endpoint URLs as described in [Viewing platform and UI endpoints](#), ignoring any segments at the end of the displayed URLs. If you are not sure of the endpoint URL, ask your administrator.

`user_id`

The user ID you want to log in with. Depending on the tasks that you want to perform, this user ID might be any of the following:

- A user ID that is a member of the cloud administration organization. This is an ID that you could also use to log in to the Cloud Manager user interface.
- A user ID that is a member of a provider organization. This is an ID that you could also use to log in to the API Manager user interface.
- A user ID that is a member of a consumer organization. This is an ID that you could also use to log in to the Developer Portal.

`password`

The password associated with the supplied user ID.

`realm`

The `realm` parameter specifies the identity provider that is used to authenticated the supplied user ID, and the scope of the tasks that can be performed after successful log in.

The format of the *realm* depends on the type of user, as follows:

- Member of the cloud administration organization:

```
admin/identity_provider
```

To determine the identity provider, see [How to determine the identity provider](#).

- Member of a provider organization:

```
provider/identity_provider
```

To determine the identity provider, see [How to determine the identity provider](#).

- Member of a consumer organization:

```
consumer:provider_org:catalog/identity_provider
```

where *provider\_org* is the name of your provider organization, and *catalog* is the name of the Catalog in that provider organization.

To determine the identity provider, see [How to determine the identity provider](#).

Important: If you log in to the CLI as a member of a consumer organization, you must supply the `--mode=consumer` parameter to the `apic login` command, and to all consumer commands. To avoid having to type the parameter every time, you can set the `mode` configuration variable, by entering the following command:

```
apic config:set mode=consumer
```

You can also use the command interactively; enter `apic login` and you will be prompted for the values. For example:

```
apic login
Enter your API Connect credentials
Server? platform-api.myserver.com
Realm? provider/default-idp-2
Username? myuser
Password?
Logged into myserver.com successfully
```

## How to determine the identity provider

If you want to log in as a member of the cloud administration organization, or as a member of a provider organization, you can help determine which identity provider to use in the *realm* parameter by entering the following command to see a list of all available identity providers (you do not need to be logged in to use this command):

```
apic identity-providers:list --scope scope --server mgmt_endpoint_url --fields name,title
```

where *scope* has the value `admin` or `provider` depending on whether you want to log in as a member of the cloud administration organization, or as a member of a provider organization. The output lists the names and titles of all identity providers, for example:

```
apic identity-providers:list --scope admin --server myserver.com --fields name,title
total_results: 2
results:
- name: default-idp-1
  title: Cloud Manager User Registry
- name: corporate-ldap
  title: Corporate LDAP user registry
```

The `title` value should enable you to determine which identity provider to use; the corresponding `name` value is what you specify in the *realm* parameter.

For any identity providers that were created by your administrator after API Connect was installed, the names will have been determined at creation time.

By default, API Connect creates a local user registry for user login for every context. The identity providers associated with these default registries are as follows:

Registry	Identity provider name
Cloud Manager Local User Registry (for login as a member of the cloud administration organization)	default-idp-1
API Manager Local User Registry (for login as a member of a provider organization)	default-idp-2
Sandbox Catalog User Registry (for login as a member of a consumer organization)	sandbox-idp

If you want to log in as a member of a consumer organization, and you are not using the default Sandbox Catalog User Registry, ask your administrator for the name of your identity provider.

## Logging out

To log out of a management server, use the following command:

```
apic logout --server mgmt_endpoint_url
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Cloud administration commands

A summary of the cloud administration commands in the IBM® API Connect developer toolkit.

## Authenticating

Use the `apic login` command to authenticate to an API Manager service, and the `apic logout` command to remove your local authentication credentials.

Note: When you authenticate successfully, your credentials are stored, in plain text, in the file `Linux .netrc` or `Windows _netrc`. You should therefore set the file permissions in such a way that your credentials are not accessible by others.

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

## Configuring the command-line tool to use TLS certificates

API Manager uses TLS profiles to secure data transmission. For information on how to create a TLS profile in API Manager, see [TLS profiles](#).

To configure the toolkit command-line tool to use certificates to communicate with an API Manager that has TLS profiles enabled, follow these steps:

For more information about the `NODE_EXTRA_CA_CERTS` environment variable, see [Node.js documentation](#).

Note: When using any CLI command that applies directly to the configuration of your IBM API Connect cloud, you must supply the parameter setting `--org admin` for a command that requires an organization name as a parameter.

## Command summary

The following tables summarize `apic` commands for cloud administration.

Table 1. Summary commands for cloud administrators

Command	Description	Sub-commands
<code>apic analytics-services</code>	Analytics Service commands	<ul style="list-style-type: none"> <li><code>apic analytics-services</code> - Analytics Service collection operations.</li> <li><code>apic analytics-services:clear</code> - Clear the Analytics Services.</li> <li><code>apic analytics-services:create</code> - Create a Analytics Service.</li> <li><code>apic analytics-services:delete</code> - Delete the Analytics Service by name or ID.</li> <li><code>apic analytics-services:get</code> - Get the Analytics Service by name or ID.</li> <li><code>apic analytics-services:list</code> - List the Analytics Services.</li> <li><code>apic analytics-services:update</code> - Update the Analytics Service by name or ID.</li> </ul>
<code>apic associates</code>	Associates commands	<ul style="list-style-type: none"> <li><code>apic associates</code> - Associate collection operations.</li> <li><code>apic associates:get</code> - Get the Associate by name or ID.</li> <li><code>apic associates:list</code> - List the Associates.</li> </ul>
<code>apic availability-zones</code>	Availability Zone commands	<ul style="list-style-type: none"> <li><code>apic availability-zones</code> - Availability Zone collection operations.</li> <li><code>apic availability-zones:clear</code> - Clear the Availability Zones.</li> <li><code>apic availability-zones:create</code> - Create a Availability Zone.</li> <li><code>apic availability-zones:delete</code> - Delete the Availability Zone by name or ID.</li> <li><code>apic availability-zones:get</code> - Get the Availability Zone by name or ID.</li> <li><code>apic availability-zones:list</code> - List the Availability Zones.</li> <li><code>apic availability-zones:update</code> - Update the Availability Zone by name or ID.</li> </ul>
<code>apic cloud-settings</code>	Cloud settings commands	<ul style="list-style-type: none"> <li><code>apic cloud-settings:get</code> - Get the Cloud Setting.</li> <li><code>apic cloud-settings:mail-server-configured</code> - indicates whether a mail server has been configured.</li> <li><code>apic cloud-settings:topology</code> - Returns summary details of the configuration of your cloud topology.</li> <li><code>apic cloud-settings:update</code> - Update the Cloud Setting.</li> </ul>
<code>apic configured-gateway-services</code>	Commands for Configured Gateway Services in a Catalog or Space.	<ul style="list-style-type: none"> <li><code>apic configured-gateway-services</code> - Configured Gateway Services operations.</li> <li><code>apic configured-gateway-services:clear</code> - Delete all Configured Gateway Services from a Catalog or Space.</li> <li><code>apic configured-gateway-services:create</code> - Configure a Gateway Service in a Catalog or Space.</li> <li><code>apic configured-gateway-services:delete</code> - Delete a Configured Gateway Service from a Catalog or Space.</li> <li><code>apic configured-gateway-services:get</code> - Obtain the details of a Configured Gateway Service in a Catalog or Space.</li> <li><code>apic configured-gateway-services:list</code> - List the Configured Gateway Services in a Catalog or Space.</li> </ul>
<code>apic credentials</code>	Credential commands	<ul style="list-style-type: none"> <li><code>apic credentials</code> - Application Credential collection operations.</li> <li><code>apic credentials:clear</code> - Clear the Application Credentials.</li> <li><code>apic credentials:create</code> - Create a Application Credential.</li> <li><code>apic credentials:delete</code> - Delete the Application Credential by name or ID.</li> <li><code>apic credentials:get</code> - Get the Application Credential by name or ID.</li> <li><code>apic credentials:list</code> - List the Application Credentials.</li> <li><code>apic credentials:reset</code> - Reset the client id and client secret.</li> <li><code>apic credentials:reset-client-secret</code> - Reset the client secret.</li> <li><code>apic credentials:update</code> - Update the Application Credential by name or ID.</li> <li><code>apic credentials:verify-client-secret</code> - Verify the client secret.</li> </ul>

Command	Description	Sub-commands
<b>apic extensions</b>	Extension commands	<ul style="list-style-type: none"> <li>• <b>apic extensions</b> - Extension collection operations.</li> <li>• <b>apic extensions:document</b> - Get the document for an extension by id, or by name and version.</li> <li>• <b>apic extensions:get</b> - Get an extension by id, or by name and version.</li> <li>• <b>apic extensions:list</b> - List all versions of an extension given the name.</li> <li>• <b>apic extensions:list-all</b> - List all versions of all extensions.</li> </ul>
<b>apic gateway-services</b>	Gateway service commands	<ul style="list-style-type: none"> <li>• <b>apic gateway-services</b> - Gateway Service collection operations.</li> <li>• <b>apic gateway-services:clear</b> - Clear the Gateway Services.</li> <li>• <b>apic gateway-services:create</b> - Create a Gateway Service.</li> <li>• <b>apic gateway-services:delete</b> - Delete the Gateway Service by name or ID.</li> <li>• <b>apic gateway-services:get</b> - Get the Gateway Service by name or ID.</li> <li>• <b>apic gateway-services:list</b> - List the Gateway Services.</li> <li>• <b>apic gateway-services:reset-oauth-secret</b> - Reset the OAuth cryptographic</li> <li>• <b>apic gateway-services:update</b> - Update the Gateway Service by name or ID.</li> </ul>
<b>apic integrations</b>	Integrations commands	<ul style="list-style-type: none"> <li>• <b>apic integrations</b> - Integration collection operations.</li> <li>• <b>apic integrations:clear</b> - Clear the Integrations.</li> <li>• <b>apic integrations:clear-all</b> - Clear all of the Integrations in all of the collections.</li> <li>• <b>apic integrations:create</b> - Create a Integration.</li> <li>• <b>apic integrations:delete</b> - Delete the Integration by name or ID.</li> <li>• <b>apic integrations:get</b> - Get the Integration by name or ID.</li> <li>• <b>apic integrations:list</b> - List the Integrations.</li> <li>• <b>apic integrations:list-all</b> - List all Integrations in all collections.</li> <li>• <b>apic integrations:update</b> - Update the Integration by name or ID.</li> </ul>
<b>apic keystores</b>	Keystores commands	<ul style="list-style-type: none"> <li>• <b>apic keystores</b> - Keystore collection operations.</li> <li>• <b>apic keystores:clear</b> - Clear the Keystores.</li> <li>• <b>apic keystores:create</b> - Create a Keystore.</li> <li>• <b>apic keystores:delete</b> - Delete the Keystore by name or ID.</li> <li>• <b>apic keystores:get</b> - Get the Keystore by name or ID.</li> <li>• <b>apic keystores:list</b> - List the Keystores.</li> <li>• <b>apic keystores:update</b> - Update the Keystore by name or ID.</li> </ul>
<b>apic mail-servers</b>	Mail server commands	<ul style="list-style-type: none"> <li>• <b>apic mail-servers</b> - Mail Server collection operations.</li> <li>• <b>apic mail-servers:clear</b> - Clear the Mail Servers.</li> <li>• <b>apic mail-servers:create</b> - Create a Mail Server.</li> <li>• <b>apic mail-servers:delete</b> - Delete the Mail Server by name or ID.</li> <li>• <b>apic mail-servers:get</b> - Get the Mail Server by name or ID.</li> <li>• <b>apic mail-servers:list</b> - List the Mail Servers.</li> <li>• <b>apic mail-servers:test-connection</b> - Test a mail server connection.</li> <li>• <b>apic mail-servers:update</b> - Update the Mail Server by name or ID.</li> </ul>
<b>apic notification-templates</b>	Notification template commands	<ul style="list-style-type: none"> <li>• <b>apic notification-templates</b> - Notification Template collection operations.</li> <li>• <b>apic notification-templates:get</b> - Get the Notification Template by name or ID.</li> <li>• <b>apic notification-templates:list</b> - List the Notification Templates.</li> <li>• <b>apic notification-templates:list-all</b> - List all Notification Templates in all collections.</li> <li>• <b>apic notification-templates:update</b> - Update the Notification Template by name or ID.</li> </ul>
<b>apic permissions</b>	Permissions commands	<ul style="list-style-type: none"> <li>• <b>apic permissions</b> - Permission collection operations.</li> <li>• <b>apic permissions:get</b> - Get the Permission by name or ID.</li> <li>• <b>apic permissions:list</b> - List the Permissions.</li> <li>• <b>apic permissions:list-all</b> - List all Permissions in all collections.</li> </ul>
<b>apic portal-services</b>	Portal services commands	<ul style="list-style-type: none"> <li>• <b>apic portal-services</b> - Portal Service collection operations.</li> <li>• <b>apic portal-services:clear</b> - Clear the Portal Services.</li> <li>• <b>apic portal-services:create</b> - Create a Portal Service.</li> <li>• <b>apic portal-services:delete</b> - Delete the Portal Service by name or ID.</li> <li>• <b>apic portal-services:get</b> - Get the Portal Service by name or ID.</li> <li>• <b>apic portal-services:list</b> - List the Portal Services.</li> <li>• <b>apic portal-services:update</b> - Update the Portal Service by name or ID.</li> </ul>
<b>apic services</b>	Services commands	<ul style="list-style-type: none"> <li>• <b>apic services</b> - Service collection operations.</li> <li>• <b>apic services:clear</b> - Clear the Services.</li> <li>• <b>apic services:clear-all</b> - Clear all Services in all collections.</li> <li>• <b>apic services:create</b> - Create a Service.</li> <li>• <b>apic services:delete</b> - Delete a Service.</li> <li>• <b>apic services:get</b> - Get the Service by name and version.</li> <li>• <b>apic services:list</b> - List the Services.</li> <li>• <b>apic services:list-all</b> - List all Services in all collections.</li> <li>• <b>apic services:update</b> - Update the Service by name and version.</li> </ul>

Command	Description	Sub-commands
<code>apic space-settings</code>	Space settings commands	<ul style="list-style-type: none"> <li>• <code>apic space-settings:get</code> - Get the Space Setting.</li> <li>• <code>apic space-settings:update</code> - Update the Space Setting.</li> </ul>
<code>apic spaces</code>	Space commands	<ul style="list-style-type: none"> <li>• <code>apic spaces</code> - Space collection operations.</li> <li>• <code>apic spaces:clear</code> - Clear the Spaces.</li> <li>• <code>apic spaces:create</code> - Create a Space.</li> <li>• <code>apic spaces:delete</code> - Delete the Space by name or ID.</li> <li>• <code>apic spaces:get</code> - Get the Space by name or ID.</li> <li>• <code>apic spaces:list</code> - List the Spaces.</li> <li>• <code>apic spaces:transfer-owner</code> - Transfer owner to a new member.</li> <li>• <code>apic spaces:update</code> - Update the Space by name or ID.</li> </ul>
<code>apic tls-client-profiles</code>	TLS client profile commands	<ul style="list-style-type: none"> <li>• <code>apic tls-client-profiles</code> - TLS Client Profile collection operations.</li> <li>• <code>apic tls-client-profiles:clear</code> - Clear the TLS Client Profiles.</li> <li>• <code>apic tls-client-profiles:clear-all</code> - Clear all of the TLS Client Profiles in the collection.</li> <li>• <code>apic tls-client-profiles:create</code> - Create a TLS Client Profile.</li> <li>• <code>apic tls-client-profiles:delete</code> - Delete the TLS Client Profile by name or ID.</li> <li>• <code>apic tls-client-profiles:get</code> - Get the TLS Client Profile by name or ID.</li> <li>• <code>apic tls-client-profiles:list</code> - List the TLS Client Profiles.</li> <li>• <code>apic tls-client-profiles:list-all</code> - Lists all of the TLS Client Profiles in a collection.</li> <li>• <code>apic tls-client-profiles:update</code> - Update the TLS Client Profile by name or ID.</li> </ul>
<code>apic tls-server-profiles</code>	TLS server profile commands	<ul style="list-style-type: none"> <li>• <code>apic tls-server-profiles</code> - TLS Server Profile collection operations.</li> <li>• <code>apic tls-server-profiles:clear</code> - Clear the TLS Server Profiles.</li> <li>• <code>apic tls-server-profiles:clear-all</code> - Clear all of the TLS Server Profiles in the collection.</li> <li>• <code>apic tls-server-profiles:create</code> - Create a TLS Server Profile.</li> <li>• <code>apic tls-server-profiles:delete</code> - Delete the TLS Server Profile by name or ID.</li> <li>• <code>apic tls-server-profiles:get</code> - Get the TLS Server Profile by name or ID.</li> <li>• <code>apic tls-server-profiles:list</code> - List the TLS Server Profiles.</li> <li>• <code>apic tls-server-profiles:list-all</code> - List all of the TLS Server Profiles in the collection.</li> <li>• <code>apic tls-server-profiles:update</code> - Update the TLS Server Profile by name or ID.</li> </ul>
<code>apic truststores</code>	Truststore commands	<ul style="list-style-type: none"> <li>• <code>apic truststores</code> - Truststore collection operations.</li> <li>• <code>apic truststores:clear</code> - Clear the Truststores.</li> <li>• <code>apic truststores:create</code> - Create a Truststore.</li> <li>• <code>apic truststores:delete</code> - Delete the Truststore by name or ID.</li> <li>• <code>apic truststores:get</code> - Get the Truststore by name or ID.</li> <li>• <code>apic truststores:list</code> - List the Truststores.</li> <li>• <code>apic truststores:update</code> - Update the Truststore by name or ID.</li> </ul>
<code>apic user-registries</code>	User registry commands	<ul style="list-style-type: none"> <li>• <code>apic user-registries</code> - User Registry collection operations.</li> <li>• <code>apic user-registries:clear</code> - Clear the User Registries.</li> <li>• <code>apic user-registries:create</code> - Create a User Registry.</li> <li>• <code>apic user-registries:delete</code> - Delete the User Registry by name or ID.</li> <li>• <code>apic user-registries:execute</code> - Execute a User Registry operation.</li> <li>• <code>apic user-registries:get</code> - Get the User Registry by name or ID.</li> <li>• <code>apic user-registries:list</code> - List the User Registries.</li> <li>• <code>apic user-registries:search</code> - search for users in a user registry.</li> <li>• <code>apic user-registries:test-connection</code> - Test a User Registry connection.</li> <li>• <code>apic user-registries:update</code> - Update the User Registry by name or ID.</li> </ul>
<code>apic user-registry-settings</code>	User registry settings commands	<ul style="list-style-type: none"> <li>• <code>apic user-registry-settings:get</code> - Get the User Registry Setting.</li> <li>• <code>apic user-registry-settings:update</code> - Update the User Registry Setting.</li> </ul>
<code>apic users</code>	Users commands	<ul style="list-style-type: none"> <li>• <code>apic users</code> - User collection operations.</li> <li>• <code>apic users:clear</code> - Clear the Users.</li> <li>• <code>apic users:create</code> - Create a User.</li> <li>• <code>apic users:delete</code> - Delete the User by name or ID.</li> <li>• <code>apic users:get</code> - Get the User by name or ID.</li> <li>• <code>apic users:list</code> - List the Users.</li> <li>• <code>apic users:request-password-reset</code> - send a password reset link.</li> <li>• <code>apic users:search-provider</code> - Search for provider users from an organization.</li> <li>• <code>apic users:update</code> - Update the User by name or ID.</li> </ul>
<code>apic webhooks</code>	Webhooks commands	<ul style="list-style-type: none"> <li>• <code>apic webhooks</code> - Webhooks collection operations.</li> <li>• <code>apic webhooks:get</code> - Get the Webhooks by name or ID.</li> <li>• <code>apic webhooks:list</code> - List the Webhooks.</li> <li>• <code>apic webhooks:update</code> - Update the Webhooks by name or ID.</li> </ul>



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API development and management commands

A summary of the core commands in the IBM® API Connect developer toolkit.

### Authenticating

Use the `apic login` command to authenticate to an API Manager service, and the `apic logout` command to remove your local authentication credentials.

Note: When you authenticate successfully, your credentials are stored, in plain text, in the file `Linux .netrc` or `Windows _netrc`. You should therefore set the file permissions in such a way that your credentials are not accessible by others.

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

### Configuring the command-line tool to use TLS certificates

API Manager uses TLS profiles to secure data transmission. For information on how to create a TLS profile in API Manager, see [TLS profiles](#).

To configure the toolkit command-line tool to use certificates to communicate with an API Manager that has TLS profiles enabled, follow these steps:

For more information about the `NODE_EXTRA_CA_CERTS` environment variable, see [Node.js documentation](#).

### Creating and managing local files

You create and work with API and Product definition YAML files locally before you stage them to API Manager.

To create a local API definition file, use the `apic create:api` command. To create a local Product definition file, use the `apic create:product` command.

Use the `apic apis` and `apic products` commands to list API Manager artifacts of the specified type.

To validate the syntactical correctness of a local API or Product definition file, use the `apic validate` command.

To create a draft API in API Manager from a local API or Product definition file, use the `apic draft-apis:create` and `apic draft-products:create` commands, respectively.

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before an API is validated, created in draft, staged, or published. For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

### Working with Catalogs and Spaces

To create a Catalog, use the `apic catalog:create` command. To view information on a Catalog, use the `apic catalog:get` command; to list all Catalogs contained in organizations that the currently authenticated user is a member of, use the `apic catalogs` command.

You can use a *Space* to partition a Catalog so multiple teams can manage Products and APIs independently in a single Catalog. A Space is conceptually like a sub-catalog, except that Products and APIs in all Spaces in a given Catalog are published to the same developer portal. For more information about Spaces, see [Using syndication in IBM API Connect](#).

To enable Spaces for a Catalog, use the following command:

```
apic catalogs:set catalog_name --spaces enabled
```

Use the toolkit `apic spaces` commands to create and manage Spaces:

- `apic spaces` - List Spaces contained in a Catalog.
- `apic:spaces create` - Create a Space in a Catalog.
- `apic:spaces get` - Get information on a Space in a Catalog.
- `apic:spaces set` - Set information on a Space in a Catalog.
- `apic:spaces delete` - Delete a Space in a Catalog.

### Command summary

The following tables summarize `apic` commands for API development and management.

Table 1. Summary of general-purpose commands

Command	Description	Sub-commands
<code>apic config</code>	List and manage configuration variables. For more information, see <a href="#">Using configuration variables</a> . With no sub-command, lists values of defined configuration variables.	<ul style="list-style-type: none"><li>• <code>apic config</code> - Manage configuration variables</li><li>• <code>apic config:clear</code> - Delete all configuration variables</li><li>• <code>apic config:delete</code> - Delete a configuration variable</li><li>• <code>apic config:get</code> - Get a configuration variable</li><li>• <code>apic config:list</code> - List the application and global configuration variables</li><li>• <code>apic config:set</code> - Set or update configuration variables</li></ul>

Command	Description	Sub-commands
<b>apic create</b>	Create a draft API or Product definition YAML file.	<ul style="list-style-type: none"> <li>• <b>create:api</b> - Create a draft API OpenAPI definition YAML file</li> <li>• <b>create:product</b> - Create a Product definition YAML file</li> </ul>
<b>apic extensions</b>	Manage OpenAPI extensions in a Catalog. With no sub-command, lists the extensions in the production Catalog.	<ul style="list-style-type: none"> <li>• <b>apic extensions</b> - Extension collection operations.</li> <li>• <b>apic extensions:clear</b> - Delete all versions of an extension given the name.</li> <li>• <b>apic extensions:clear-all</b> - Delete all versions of all extensions.</li> <li>• <b>apic extensions:clone</b> - Download all versions of all extensions to your local drive.</li> <li>• <b>apic extensions:create</b> - Create an extension.</li> <li>• <b>apic extensions:delete</b> - Delete an extension by id, or by name and version.</li> <li>• <b>apic extensions:document</b> - Get the document for an extension by id, or by name and version.</li> <li>• <b>apic extensions:get</b> - Get an extension by id, or by name and version.</li> <li>• <b>apic extensions:list</b> - List all versions of an extension given the name.</li> <li>• <b>apic extensions:list-all</b> - List all versions of all extensions.</li> <li>• <b>apic extensions:update</b> - Update an extension by id, or by name and version.</li> </ul>
<b>apic lb</b>	Create LoopBack® project and project artifacts. With no sub-command, creates a new LoopBack project.	<ul style="list-style-type: none"> <li>• <b>acl</b> - Add access control list specification</li> <li>• <b>app</b> - Create a new LoopBack project (default)</li> <li>• <b>boot-script</b> - Add boot script</li> <li>• <b>datasource</b> - Create data source</li> <li>• <b>export-api-def</b> - Generate OpenAPI definitions from models</li> <li>• <b>middleware</b> - Add middleware function</li> <li>• <b>model</b> - Create model.</li> <li>• <b>oracle</b> - Utility to help install LoopBack Oracle connector</li> <li>• <b>property</b> - Add property to existing model</li> <li>• <b>relation</b> - Add relation between models</li> <li>• <b>remote-method</b> - Add remote method</li> <li>• <b>soap</b> - Generate models based on SOAP web service</li> <li>• <b>swagger</b> - Generate LoopBack project from an OpenAPI definition</li> </ul>
<b>apic login</b>	Log in to API Manager.	<p>None. Specify server and credentials with the required flags:</p> <ul style="list-style-type: none"> <li>• <b>-p, --password password</b></li> <li>• <b>-r, --realm realm</b>. Determines the identity provider that is used to authenticated the supplied user ID, and the scope of the tasks that can be performed after successful log in.</li> <li>• <b>-s, --server mgmt_endpoint_url</b>. Use the value of the platform API endpoint URL for this entry.</li> <li>• <b>-u, --username user_name</b></li> </ul> <p>For full details on how to log in to your management server from the CLI, see <a href="#">Logging in to the management server</a>.</p>
<b>apic logout</b>	Log out from API Manager.	<p>None. Specify server with the required flag:</p> <ul style="list-style-type: none"> <li>• <b>-s, --server mgmt_service</b>.</li> </ul> <p>You can append the port number to the server name if it is not the default value of 443.</p>
<b>apic validate</b>	Validate API or Product definition YAML file.	None

Table 2. Summary of commands to manage APIs, Products, and Catalogs

Command	Description	Sub-commands
<b>apic apis</b>	List and manage APIs that are staged or published to Catalog or Space. Default sub-command is <b>list-all</b> .	<ul style="list-style-type: none"> <li>• <b>apic apis</b> - API collection operations</li> <li>• <b>apic apis:get</b> - Get the OpenAPI definition YAML file for an API by name and version, or by ID</li> <li>• <b>apic apis:list</b> - List an API by name</li> <li>• <b>apic apis:list-all</b> - List all APIs in a Catalog or Space</li> <li>• <b>apic apis:update</b> - Update an API by name</li> </ul>
<b>apic apps</b>	List and manage developer applications that are registered in a consumer organization. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic apps</b> - Application collection operations</li> <li>• <b>apic apps:clear</b> - Clear all applications</li> <li>• <b>apic apps:create</b> - Create an application</li> <li>• <b>apic apps:delete</b> - Delete an application by name or ID</li> <li>• <b>apic apps:get</b> - Get an application object by name</li> <li>• <b>apic apps:list</b> - List all the applications in a consumer organization</li> <li>• <b>apic apps:update</b> - Update an application by name or ID</li> </ul>
<b>apic catalogs</b>	List and manage Catalogs in a provider organization. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic catalogs:clear</b> - Clear all Catalogs from provider organization</li> <li>• <b>apic catalogs:create</b> - Create a Catalog</li> <li>• <b>apic catalogs:delete</b> - Delete a Catalog by name or ID</li> <li>• <b>apic catalogs:get</b> - Get a Catalog object by name or ID</li> <li>• <b>apic catalogs:list</b> - List all the Catalogs in a provider organization</li> <li>• <b>apic catalogs:transfer-owner</b> - Transfer the ownership of a Catalog to another user</li> <li>• <b>apic catalogs:update</b> - Update a Catalog by name or ID</li> </ul>

Command	Description	Sub-commands
<b>apic catalog-settings</b>	Manage the configuration settings for a Catalog	<ul style="list-style-type: none"> <li>• <b>apic catalog-settings:get</b> - Get the Catalog settings.</li> <li>• <b>apic catalog-settings:update</b> - Update the Catalog settings.</li> </ul>
<b>apic drafts</b>	List and manage the draft API and Product definitions in a provider organization. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic drafts</b> - Draft collection operations</li> <li>• <b>apic drafts:clear</b> - Delete all draft APIs and Products in a provider organization</li> <li>• <b>apic drafts:list</b> - List the draft APIs and Products in a provider organization</li> </ul>
<b>apic draft-apis</b>	List and manage the draft API definitions in a provider organization. Default sub-command is <b>list-all</b> .	<ul style="list-style-type: none"> <li>• <b>apic draft-apis</b> - Draft API collection operations</li> <li>• <b>apic draft-apis:clear</b> - Delete all versions of a draft API by name</li> <li>• <b>apic draft-apis:clear-all</b> - Delete all draft APIs in a provider organization</li> <li>• <b>apic draft-apis:create</b> - Create a draft API</li> <li>• <b>apic draft-apis:delete</b> - Delete a Draft API by name and version</li> <li>• <b>apic draft-apis:get</b> - Get the OpenAPI definition YAML file for an API by name and version</li> <li>• <b>apic draft-apis:list</b> - List all versions of a draft API by name</li> <li>• <b>apic draft-apis:list-all</b> - List all draft APIs in a provider organization</li> <li>• <b>apic draft-apis:update</b> - Update a draft API by name and version, supplying the revised OpenAPI definition file</li> <li>• <b>apic draft-apis:validate</b> - Validates a draft API</li> </ul>
<b>apic draft-products</b>	List and manage the draft Product definitions in a provider organization. Default sub-command is <b>list-all</b> .	<ul style="list-style-type: none"> <li>• <b>apic draft-products</b> - Draft Product collection operations</li> <li>• <b>apic draft-products:clear</b> - Delete all versions of a draft Product by name</li> <li>• <b>apic draft-products:clear-all</b> - Delete all draft Products in a provider organization</li> <li>• <b>apic draft-products:create</b> - Create a draft Product</li> <li>• <b>apic draft-products:delete</b> - Delete a draft Product by name and version</li> <li>• <b>apic draft-products:get</b> - Get the Draft Product by name and version</li> <li>• <b>apic draft-products:list</b> - List the Draft Products</li> <li>• <b>apic draft-products:list-all</b> - List all draft Products in a provider organization</li> <li>• <b>apic draft-products:publish-all</b> - Publishes the Draft Product</li> <li>• <b>apic draft-products:update</b> - Update a draft Product by name and version, supplying the revised definition file</li> <li>• <b>apic draft-products:validate</b> - Validates a draft Product</li> </ul>
<b>apic members</b>	List and manage the members of a provider organization, consumer organization, Catalog, or Space. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic members</b> - Member operations</li> <li>• <b>apic members:clear</b> - Delete all members</li> <li>• <b>apic members:create</b> - Create a member</li> <li>• <b>apic members:delete</b> - Delete a member by name or ID</li> <li>• <b>apic members:get</b> - Get a member object by name or ID</li> <li>• <b>apic members:list</b> - List all members</li> <li>• <b>apic members:update</b> - Update the member by name or ID</li> </ul>
<b>apic member-invitations</b>	List and manage member invitations. A member invitation is created when a user is invited to be a member of a provider organization, consumer organization, Catalog, or Space. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic member-invitations</b> - Member invitation collection operations.</li> <li>• <b>apic member-invitations:clear</b> - Delete all member invitations.</li> <li>• <b>apic member-invitations:create</b> - Create a member invitation.</li> <li>• <b>apic member-invitations:delete</b> - Delete a member invitation by name or ID.</li> <li>• <b>apic member-invitations:get</b> - Get a member invitation object by name or ID.</li> <li>• <b>apic member-invitations:list</b> - List all member invitations.</li> <li>• <b>apic member-invitations:update</b> - Update a member invitation by name or ID.</li> </ul>

Command	Description	Sub-commands
<b>apic orgs</b>	List and manage provider organizations, and the admin organization. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic orgs</b> - Organization collection operations</li> <li>• <b>apic orgs:clear</b> - Delete all organizations</li> <li>• <b>apic orgs:create</b> - Create an organization</li> <li>• <b>apic orgs:delete</b> - Delete an organization by name or ID</li> <li>• <b>apic orgs:get</b> - Get an organization object by name or ID</li> <li>• <b>apic orgs:list</b> - List all organizations</li> <li>• <b>apic orgs:transfer-owner</b> - Transfer ownership of an organization</li> <li>• <b>apic orgs:update</b> - Update an organization by name or ID</li> </ul>
<b>apic org-settings</b>	Manage settings for provider organizations, and the admin organization.	<ul style="list-style-type: none"> <li>• <b>apic org-settings:get</b> - Get the settings object for an organization</li> <li>• <b>apic org-settings:update</b> - Update the settings for an organization</li> </ul>
<b>apic policies</b>	List and manage policies in a Catalog. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic policies</b> - Policy collection operations</li> <li>• <b>apic policies:clear</b> - Clear the Policies</li> <li>• <b>apic policies:clear-all</b> - Clear all Policies in all collections</li> <li>• <b>apic policies:create</b> - Create a Policy</li> <li>• <b>apic policies:delete</b> - Delete a Policy</li> <li>• <b>apic policies:get</b> - Get the Policy by name and version</li> <li>• <b>apic policies:list</b> - List the Policies</li> <li>• <b>apic policies:list-all</b> - List all Policies in all collections</li> <li>• <b>apic policies:update</b> - Update the Policy by name and version</li> </ul>
<b>apic products</b>	List and manage Products that are staged or published to Catalog or Space. Default sub-command is <b>list-all</b> .	<ul style="list-style-type: none"> <li>• <b>apic products</b> - Product collection operations</li> <li>• <b>apic products:clear</b> - Delete all versions of a Product by name</li> <li>• <b>apic products:clear-all</b> - Delete all Products</li> <li>• <b>apic products:delete</b> - Delete a Product by name and version</li> <li>• <b>apic products:execute-migration-target</b> - Products execute migration target operations</li> <li>• <b>apic products:get</b> - Get a Product object by name and version</li> <li>• <b>apic products:list</b> - List all versions of a Product by name</li> <li>• <b>apic products:list-all</b> - List all Products</li> <li>• <b>apic products:publish</b> - Publish a Product to a Catalog or Space, supplying the Product definition YAML file</li> <li>• <b>apic products:replace</b> - Replace a Product with another Product</li> <li>• <b>apic products:set-migration-target</b> - Set the target Product for migrating subscriptions from a Product</li> <li>• <b>apic products:supersede</b> - Supersede a Product with another Product</li> <li>• <b>apic products:update</b> - Update a Product by name and version, supplying the revised Product definition YAML file</li> </ul>
<b>apic identity-providers</b>	View information about identity providers.	<ul style="list-style-type: none"> <li>• <b>apic identity-providers</b> - Identity provider operations</li> <li>• <b>apic identity-providers:list</b> - List the identity providers.</li> </ul>
<b>apic spaces</b>	List and manage Spaces contained in a Catalog. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic spaces</b> - Space collection operations</li> <li>• <b>apic spaces:clear</b> - Delete all Spaces from a Catalog</li> <li>• <b>apic spaces:create</b> - Create a Space in a Catalog</li> <li>• <b>apic spaces:delete</b> - Delete a Space from a Catalog, by name or ID</li> <li>• <b>apic spaces:get</b> - Get a Space object by name or ID</li> <li>• <b>apic spaces:list</b> - List all the Spaces in a Catalog</li> <li>• <b>apic spaces:transfer-owner</b> - Transfer the ownership of a Space to another user</li> <li>• <b>apic spaces:update</b> - Update a Space by name or ID</li> </ul>
<b>apic subscriptions</b>	List and manage subscriptions in a Product or a Catalog. Default sub-command is <b>list</b> .	<ul style="list-style-type: none"> <li>• <b>apic subscriptions</b> - Subscription collection operations</li> <li>• <b>apic subscriptions:clear</b> - Clear the Subscriptions</li> <li>• <b>apic subscriptions:create</b> - Create a Subscription</li> <li>• <b>apic subscriptions:delete</b> - Delete the Subscription by name or ID</li> <li>• <b>apic subscriptions:get</b> - Get the Subscription by name or ID</li> <li>• <b>apic subscriptions:list</b> - List the Subscriptions</li> <li>• <b>apic subscriptions:update</b> - Update the Subscription by name or ID</li> </ul>

Table 3. Summary of other commands for API developers and managers

Command	Description	Sub-commands
---------	-------------	--------------

Command	Description	Sub-commands
<code>apic consumer-orgs</code>	Manage consumer organizations	<ul style="list-style-type: none"> <li>• <code>apic consumer-orgs</code> - Consumer Organization collection operations.</li> <li>• <code>apic consumer-orgs:clear</code> - Clear the Consumer Organizations.</li> <li>• <code>apic consumer-orgs:create</code> - Create a Consumer Organization.</li> <li>• <code>apic consumer-orgs:delete</code> - Delete the Consumer Organization by name or ID.</li> <li>• <code>apic consumer-orgs:get</code> - Get the Consumer Organization by name or ID.</li> <li>• <code>apic consumer-orgs:list</code> - List the Consumer Organizations.</li> <li>• <code>apic consumer-orgs:transfer-owner</code> - Transfer owner to a new member.</li> <li>• <code>apic consumer-orgs:update</code> - Update the Consumer Organization by name or ID.</li> </ul>
<code>apic consumer-org-settings</code>	Manage consumer organization settings	<ul style="list-style-type: none"> <li>• <code>apic consumer-org-settings:delete</code> - Delete the Consumer Organization Setting.</li> <li>• <code>apic consumer-org-settings:get</code> - Get the Consumer Organization Setting.</li> <li>• <code>apic consumer-org-settings:update</code> - Update the Consumer Organization Setting.</li> </ul>
<code>apic groups</code>	Manage groups	<ul style="list-style-type: none"> <li>• <code>apic groups</code> - Group collection operations.</li> <li>• <code>apic groups:clear</code> - Clear the Groups.</li> <li>• <code>apic groups:create</code> - Create a Group.</li> <li>• <code>apic groups:delete</code> - Delete the Group by name or ID.</li> <li>• <code>apic groups:get</code> - Get the Group by name or ID.</li> <li>• <code>apic groups:list</code> - List the Groups.</li> <li>• <code>apic groups:update</code> - Update the Group by name or ID.</li> </ul>
<code>apic invitations</code>	Manage invitations. An invitation is created when a user is invited to be the owner of a provider organization, consumer organization, Catalog, or Space.	<ul style="list-style-type: none"> <li>• <code>apic invitations</code> - Invitation collection operations.</li> <li>• <code>apic invitations:clear</code> - Delete all invitations.</li> <li>• <code>apic invitations:create</code> - Create an invitation.</li> <li>• <code>apic invitations:delete</code> - Delete an invitation by name or ID.</li> <li>• <code>apic invitations:get</code> - Get the details of an invitation by name or ID.</li> <li>• <code>apic invitations:list</code> - List all invitations.</li> <li>• <code>apic invitations:update</code> - Update an invitation by name or ID.</li> </ul>
<code>apic member-invitations</code>	Manage member invitations. A member invitation is created when a user is invited to be a member of a provider organization, consumer organization, Catalog, or Space.	<ul style="list-style-type: none"> <li>• <code>apic member-invitations</code> - Member invitation collection operations.</li> <li>• <code>apic member-invitations:clear</code> - Delete all member invitations.</li> <li>• <code>apic member-invitations:create</code> - Create a member invitation.</li> <li>• <code>apic member-invitations:delete</code> - Delete a member invitation by name or ID.</li> <li>• <code>apic member-invitations:get</code> - Get the details of a member invitation by name or ID.</li> <li>• <code>apic member-invitations:list</code> - List all member invitations.</li> <li>• <code>apic member-invitations:update</code> - Update a member invitation by name or ID.</li> </ul>
<code>apic members</code>		<ul style="list-style-type: none"> <li>• <code>apic members</code> - Member collection operations.</li> <li>• <code>apic members:clear</code> - Clear the Members.</li> <li>• <code>apic members:create</code> - Create a Member.</li> <li>• <code>apic members:delete</code> - Delete the Member by name or ID.</li> <li>• <code>apic members:get</code> - Get the Member by name or ID.</li> <li>• <code>apic members:list</code> - List the Members.</li> <li>• <code>apic members:update</code> - Update the Member by name or ID.</li> </ul>

Command	Description	Sub-commands
<b>apic registration</b>		<ul style="list-style-type: none"> <li>• <b>apic registrations</b> - Registration collection operations.</li> <li>• <b>apic registrations:clear</b> - Clear the Registrations.</li> <li>• <b>apic registrations:create</b> - Create a Registration.</li> <li>• <b>apic registrations:delete</b> - Delete the Registration by name or ID.</li> <li>• <b>apic registrations:get</b> - Get the Registration by name or ID.</li> <li>• <b>apic registrations:list</b> - List the Registrations.</li> <li>• <b>apic registrations:update</b> - Update the Registration by name or ID.</li> </ul>
<b>apic role-defaults</b>		<ul style="list-style-type: none"> <li>• <b>apic role-defaults</b> - Role Default collection operations.</li> <li>• <b>apic role-defaults:clear</b> - Clear the Role Defaults.</li> <li>• <b>apic role-defaults:create</b> - Create a Role Default.</li> <li>• <b>apic role-defaults:delete</b> - Delete the Role Default by name or ID.</li> <li>• <b>apic role-defaults:get</b> - Get the Role Default by name or ID.</li> <li>• <b>apic role-defaults:list</b> - List the Role Defaults.</li> <li>• <b>apic role-defaults:list-all</b> - List all Role Defaults in all collections.</li> <li>• <b>apic role-defaults:update</b> - Update the Role Default by name or ID.</li> </ul>
<b>apic roles</b>		<ul style="list-style-type: none"> <li>• <b>apic roles</b> - Role collection operations.</li> <li>• <b>apic roles:clear</b> - Clear the Roles.</li> <li>• <b>apic roles:create</b> - Create a Role.</li> <li>• <b>apic roles:delete</b> - Delete the Role by name or ID.</li> <li>• <b>apic roles:get</b> - Get the Role by name or ID.</li> <li>• <b>apic roles:list</b> - List the Roles.</li> <li>• <b>apic roles:update</b> - Update the Role by name or ID.</li> </ul>
<b>apic tasks</b>		<ul style="list-style-type: none"> <li>• <b>apic tasks</b> - Task collection operations.</li> <li>• <b>apic tasks:clear</b> - Clear the Tasks.</li> <li>• <b>apic tasks:create</b> - Create a Task.</li> <li>• <b>apic tasks:delete</b> - Delete the Task by name or ID.</li> <li>• <b>apic tasks:get</b> - Get the Task by name or ID.</li> <li>• <b>apic tasks:list</b> - List the Tasks.</li> <li>• <b>apic tasks:update</b> - Update the Task by name or ID.</li> </ul>
<b>apic user-registries</b>		<ul style="list-style-type: none"> <li>• <b>apic user-registries</b> - User Registry collection operations.</li> <li>• <b>apic user-registries:clear</b> - Clear the User Registries.</li> <li>• <b>apic user-registries:create</b> - Create a User Registry.</li> <li>• <b>apic user-registries:delete</b> - Delete the User Registry by name or ID.</li> <li>• <b>apic user-registries:execute</b> - Execute a User Registry operation.</li> <li>• <b>apic user-registries:get</b> - Get the User Registry by name or ID.</li> <li>• <b>apic user-registries:list</b> - List the User Registries.</li> <li>• <b>apic user-registries:search</b> - search for users in a user registry.</li> <li>• <b>apic user-registries:test-connection</b> - Test a User Registry connection.</li> <li>• <b>apic user-registries:update</b> - Update the User Registry by name or ID.</li> </ul>

Command	Description	Sub-commands
<code>apic user-registry-settings</code>		<ul style="list-style-type: none"> <li>• <code>apic user-registry-settings:delete</code> - Delete the User Registry Setting.</li> <li>• <code>apic user-registry-settings:get</code> - Get the User Registry Setting.</li> <li>• <code>apic user-registry-settings:update</code> - Update the User Registry Setting.</li> </ul>
<code>apic users</code>		<ul style="list-style-type: none"> <li>• <code>apic users</code> - User collection operations.</li> <li>• <code>apic users:clear</code> - Clear the Users.</li> <li>• <code>apic users:create</code> - Create a User.</li> <li>• <code>apic users:delete</code> - Delete the User by name or ID.</li> <li>• <code>apic users:get</code> - Get the User by name or ID.</li> <li>• <code>apic users:list</code> - List the Users.</li> <li>• <code>apic users:request-password-reset</code> - send a password reset link.</li> <li>• <code>apic users:search-provider</code> - Search for provider users from an organization.</li> <li>• <code>apic users:update</code> - Update the User by name or ID.</li> </ul>

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating APIs and applications

You can develop API proxies and API implementations by using the developer toolkit. In the documentation, *API* refers to the API proxy and *application* refers to the API implementation.

LoopBack® is a high-performance Node.js interaction-tier framework for APIs and micro-services. The [LoopBack framework](#) can be downloaded and installed using the **CLI + LoopBack + Designer** option for downloading the toolkit. See [Installing the toolkit](#). To create a new LoopBack project, use the command `apic lb`.

You can use the developer toolkit or LoopBack to create language-independent APIs by using [OpenAPI](#) to proxy to an existing backend implementation or to augment applications developed in other languages or frameworks such as [Express®](#), Java™, [Swift](#), Go, and others.

## Creating development artifact definitions

Use the `apic create` command to create development artifacts, by using the following commands:

Command	Description
<code>apic create:api</code>	Create an OpenAPI definition.
<code>apic create:api --wsdl filename</code>	Create a SOAP API definition from a WSDL definition file, or a .zip file that contains the WSDL definition files for a service. The name and version of the generated API are obtained from the WSDL file. If you upload a .zip file, you can include in the .zip file an options file to specify additional directives. For details, see <a href="#">Using an options file when importing a WSDL service</a> .
<code>apic create:product</code>	Create an API Product definition.

Note: You can create an API or Product from an OpenAPI template file by using the `--template template-name` option.

You can also create Product and API definitions non-interactively by providing the `--title` option. This option sets several values that you can also customize with additional options; for example:

```
apic create:api --title Routes
apic create:product --title "Climb On"
```

You can also create the API and Product definitions at the same time:

```
apic create:api --title Routes --product "Climb On"
apic create:api --wsdl globalweather.wsdl --product "Weather Forecasting"
```

Alternatively, you can create APIs and then reference them when you create a new Product; for example:

```
apic create:api --title Routes
apic create:api --title Ascents
apic create:product --title "Climb On" --apis "routes.yaml ascents.yaml"
```

## Validating development artifact definitions

After you edit development artifacts or before you publish artifacts, best practice is to validate them; for example:

```
apic validate routes.yaml # Validate an API
apic validate climb-on.yaml # Validate the Product and its APIs
apic validate climb-on.yaml --product-only # Validate the Product only (do not validate the referenced APIs)
```

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the draft API is created with the `apic drafts:validate` command. For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

## Developing LoopBack applications

After you've created a LoopBack application with the `apic lb app` command, you can add additional functionality to the application with the commands listed in the following table. Run these commands from the project root directory of the application. APIs created with LoopBack can be imported into API Manager.

Command	Description
<code>apic lb acl</code>	Define an <a href="#">access control list</a> for accessing LoopBack models.
<code>apic lb app</code>	Create a LoopBack application with the <a href="#">Application generator</a> .
<code>apic lb bluemix</code>	Add IBM Cloud deployment artifacts to a LoopBack application. See <a href="#">IBM Cloud configuration generator</a> .
<code>apic lb boot-script</code>	Create a new <a href="#">boot script</a> .
<code>apic lb datasource</code>	<a href="#">Data source generator</a>
<code>apic lb export-api-def</code>	Export an OpenAPI and a Product definition from LoopBack models.
<code>apic lb middleware</code>	Define and register <a href="#">Express middleware</a> to define the phase of execution.
<code>apic lb model</code>	Add a new model to a LoopBack application: <a href="#">Model generator</a>
<code>apic lb oracle</code>	Install and troubleshoot the Oracle connector: <a href="#">Oracle installer</a>
<code>apic lb property</code>	Add <a href="#">additional properties</a> to a Loopback model.
<code>apic lb relation</code>	Add <a href="#">relationships</a> between LoopBack models.
<code>apic lb remote-method</code>	Add <a href="#">remote methods</a> to a LoopBack model.
<code>apic lb soap</code>	Generate a model from a SOAP web service:
<code>apic lb swagger</code>	Generate a LoopBack application project from an OpenAPI definition.

Note: These commands are annotated with "Stability: prototype" because IBM® is looking for feedback on them before certifying them for production.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Publishing APIs and applications

To publish APIs and applications by using the developer toolkit of IBM® API Connect, you set configuration variables to define where you want to publish, log in to the target cloud platform, and then use the appropriate publishing commands.

See the following sections for more information:

- [Setting configuration variables](#)
- [Logging in to API Connect](#)
- [Publishing APIs](#)

## Setting configuration variables

The `apic config` command provides global and project-based configuration variables that specify the target Catalog for publishing APIs and applications. The values of these variables are stored in `~/.apiconnect/config` (for global variables) and `project-dir/.apiconnect` (for project variables). For a full list of configuration variables, see [Overview of the command-line tool](#).

Set the `catalog` configuration variable to the URI of an API Connect Catalog to define a default Catalog target for all commands managing Catalogs. The `catalog` URI has the form:

```
https://mgmt_endpoint_url/api/catalogs/org_name/catalog_name
```

where `mgmt_endpoint_url` is the platform API endpoint URL, `org_name` is the provider organization name, and `catalog_name` is the Catalog name. The `mgmt_endpoint_url` portion sets the default value of the `--server` option, the `org_name` portion sets the default value of the `--org` option, and the `catalog_name` portion sets the default value of the `--catalog` option; you can override any of these values by including the corresponding option in a command.

Set the `space` configuration variable to the URI of an API Connect Space, to define a default Space target for all commands that manage Spaces. The `space` URI has the form:

```
https://mgmt_endpoint_url/api/spaces/org_name/catalog_name/space_name
```

where `mgmt_endpoint_url` is the platform API endpoint URL, `org_name` is the provider organization name, `catalog_name` is the Catalog name, and `space_name` is the name of the Space. The `mgmt_endpoint_url` portion sets the default value of the `--server` option, the `org_name` portion sets the default value of the `--org` option, the `catalog_name` portion sets the default value of the `--catalog` option, and the `space_name` sets the default value of the `--space` option; you can override any of these values by including the corresponding option in a command.

Although setting these configuration variables is not required, doing so simplifies commands that interact with API Connect clouds by providing default values for frequently used command-line options.

Here is an example of publishing with and without the `catalog` configuration variable set.

Without the configuration variable set:

```
apic products publish climb-on.yaml --server mgmthost.com --org climbon --catalog sandbox
```

With the configuration variable set:

```
apic config:set catalog=https://platform-api.myserver.com/api/catalogs/climbon/sandbox
catalog: https://platform-api.myserver.com/api/catalogs/climbon/sandbox
apic products publish climb-on.yaml
```



You can override default values provided by the `catalog` configuration variable by providing one of the standard options with a different value. For example, use the `--catalog` option with the `apic products publish` command to specify the `qa` Catalog:

```
apic products publish climb-on.yaml --catalog qa
```

Don't forget about global configuration variables. If you use the same Catalog as the default target for multiple projects, set the value globally:

```
apic config:set --global catalog=https://platform-api.myserver.com/api/catalogs/climb-on/sandbox
```

Note:

If you have an environment variable of the same name as a CLI configuration property then, by default, its value will override the value of the corresponding CLI configuration property for any CLI command at that scope.

For example, if you have defined an environment variable called `SPACE` then, by default, that value will be assumed for the value of the `--space` parameter in the following command, regardless of any `space` configuration property setting:

```
apic products:publish my_product.yaml --scope space
```

To prevent environment variables overriding CLI configuration properties, define an environment variable called `APIC_LOAD_FROM_ENV`, set to the value `false`.

## Logging in to API Connect

Use the `apic login` and `apic logout` commands to manage your authentication. For details, see [Logging in to a management server](#).

## Publishing APIs

Publishing APIs to API Catalogs in API Connect clouds enables you to socialize the APIs by using the developer portal and secure them by using the Gateway.

An *API Product* (or simply *Product*) is used to compose APIs for publishing. API Product managers can use it to bundle one or more APIs together, control the visibility of the Product in the developer portal (for example, only allow partners x, y, and z to view and subscribe to the Product), and define Plans to provide consumption options. The Products that reference the APIs and define the consumption Plans are also the primary unit of lifecycle management for APIs.

Use the `apic products publish` command (equivalent to `apic products:publish`) to publish API Products to an API Connect cloud. The following example demonstrates how to create APIs composed by a Product, and publishing the Product and its APIs to a Catalog:

```
apic create:api --title Routes
apic create:api --title Ascents
apic create:product --title "Climb On" --apis "routes.yaml ascents.yaml"
apic config:set catalog=https://platform-api.myserver.com/api/catalogs/climb-on/sandbox
apic login --username some-user --password some-password --server platform-api.myserver.com --realm provider/default-idp-2
apic products publish climb-on.yaml
```

For full details on how to log in to your management server from the CLI, see [Logging in to a management server](#).

Add the `--stage` option to `apic publish` to stage the Product into a Catalog instead of publishing it. Products in a Catalog can be in the following states: staged, published, deprecated, or retired. For example:

```
apic products publish --stage climb-on.yaml
```

You can use a *Space* to partition a Catalog so multiple teams can manage Products and APIs independently in a single Catalog. A Space is conceptually like a sub-catalog, except that Products and APIs in all Spaces in a given Catalog are published to the same developer portal. For more information about Spaces, see [Using syndication in IBM API Connect](#).

If default configuration values have been set for the Space, Catalog, organization and management server, the following publish command could be used:

```
apic products publish --scope space product.yaml
```

where *product* is the name of the product that you want to publish.

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published with the `apic publish` command. For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Managing API Products

Use the `apic products` and `apic apis` commands to manage Products and APIs that have been published to IBM® API Connect Catalogs. Use the `--scope space` option to manage Products and APIs that have been published to Spaces within Catalogs.

## Product management command summary

The following table summarizes the commands available for managing API Products. Usage examples follow.

Example command	Description
<pre>apic config:set catalog=https://platform- api.myserver.com/api/catalogs/myorg/sandbox</pre>	Set the default Catalog.

Example command	Description
<code>apic login --username some-user --password some-password --server platform-api.myserver.com --realm provider/my-identity-provider</code>	Login into the management server. For full details on how to log in to your management server from the CLI, see <a href="#">Logging in to the management server</a> .
<code>apic create:api --title Routes --product "ClimbOn"</code>	Create the Product and API.
<code>apic products:publish climbon.yaml</code>	Publish the Product to the default catalog. Add <code>--stage</code> argument to stage the Product.
<code>apic products:list-all --scope catalog</code>	List the Products in the default catalog.
<code>apic products:get climbon:1.0.0 --output -</code>	Display the Product's properties. Omit the output argument to download to a file.
<code>apic apis:list-all --scope catalog</code>	List the APIs in the Catalog.
<code>apic apis:get routes:1.0.0 --output -</code>	Get the API properties. Omit the output argument to download to a file.
<code>apic products:replace climbon:1.0.0 mapfile.txt</code>	Replace the Product indicated on the command line with the staged or published Product designated in the mapfile. This retires the replaced Product.
<code>apic products:supersede climbon:1.0.0 mapfile.txt</code>	Supersede the Product indicated on the command line with the staged or published Product designated in the mapfile. This deprecates the superseded Product.
<code>apic products:delete climbon:1.0.0</code>	Delete the Product from the Catalog. The Product must be either retired or deprecated.

## Using the products and apis commands through a full lifecycle

This example shows a complex lifecycle where a new version of a Product and API replaces the original version at run time.

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published with the `apic publish` command. For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

Set the default Catalog and login to the mgmthost.com API Connect cloud

```
apic config:set catalog=https://platform-api.myserver.com/api/catalogs/climbon/sandbox
apic login --username some-user --password some-password --server platform-api.myserver.com --realm my-identity-provider
```

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

Create and publish an initial version

```
apic create:api --title Routes --version 1.0.0 --filename routes100.yaml
apic create:product --title "Climb On" --version 1.0.0 --apis routes100.yaml --filename climbon100.yaml
apic products:publish climbon100.yaml
```

Create a new version to fix a bug in the API, stage it to the Catalog

```
apic create:api --title Routes --version 1.0.1 --filename routes101.yaml
apic create:product --title "Climb On" --version 1.0.1 --apis routes101.yaml --filename climbon101.yaml
apic products:publish --stage climbon101.yaml
```

Inspect the Catalog

```
apic products:list-all --scope catalog
```

This produces a list of all Products in the catalog. Copy the ID of the `climbon:1.0.1` Product. The ID resembles the following example.

```
https://server/api/catalogs/3eca6046-3c27-4ad/ab8772bf-0f65-45/products/8f854623-bda1-4f9
```

Create a mapping file

For example, if you are replacing a Product, the mapping file specifies the Product that you want to replace, and the mapping of the Plans from the source Product to the target Product. For example, if you are replacing `climbon:1.0.0` with `climbon:1.0.1` then `product_url` property specifies the URL of `climbon:1.0.0`. This file takes the following form:

```
product_url: https://server/api/catalogs/{id}/products/{id}
plans:
- source: {source_plan_name_1}
  target: {target_plan_name_1}
- source: {source_plan_name_2}
  target: {target_plan_name_2}
  .
  .
  .
```

Note:

- The source and target Plan names can be the same or different.
- Every Plan in the Product you are replacing must be mapped to a Plan in the replacement Product.

"Hot-replace" version 1.0.0 with 1.0.1

```
apic products:replace climbon:1.0.1 PRODUCT_PLAN_MAPPING_FILE
```

After it is replaced in this way, a Product is retired. It is no longer active.

Note: The Product specified on the command line is the replacement Product. For example, if you are replacing `climbon:1.0.0` with `climbon:1.0.1` then the Product specified on the command line is `climbon:1.0.1`.

Supersede version 1.0.0 with 1.0.1

Rather than hot-replacing an existing Product with a new one (usually an updated version), you can supersede the existing Product with a new one. When you replace a Product, all external application subscriptions to that Product are automatically moved to the new Product. When you supersede a Product, the subscriptions are not automatically moved to the new Product, some action must be taken to subscribe external applications to the superseding Product.

The command to supersede a Product uses the same mapping file as the command to replace a Product. The name of the superseding Product is given on the command line.

```
apic products:supersede climbon:1.0.1 PRODUCT_PLAN_MAPPING_FILE
```

Once superseded, a Product is marked deprecated, meaning that no further subscriptions to the Product are allowed, although it is still active and existing subscriptions still work.

Note: The Product specified on the command line is the superseding Product. For example, if you are replacing `climbon:1.0.0` with `climbon:1.0.1` then the Product specified on the command line is `climbon:1.0.1`. The mapping file specifies the URL of the Product you are superseding.

Set a migration target

It is possible to set a migration target for the existing Product. This helps migration.

Use a command like the following to set the migration target.

```
apic products:set-migration-target climbon:1.0.0 PRODUCT_PLAN_MAPPING_FILE
```

Note: The Product specified on the command line is the Product from which you want to migrate subscriptions. The mapping file specifies the target Product for the subscriptions.

Once a migration target is set, external application developers will be able to migrate their application subscriptions through the Developer Portal with ease. It is also possible to execute a subscription migration using the following command.

```
apic products:execute-migration-target climbon:1.0.0
```

Delete a Product

When the replaced or superseded Product is no longer needed, it can be deleted.

```
apic products:delete climbon:1.0.0
```

## Additional Product and API operations

In addition to the lifecycle management capabilities, you can download Products and APIs in a catalog or space using the `clone` sub-command:

Command	Description
<code>apic products:clone</code>	Download all Products and their APIs from the catalog or space.
<code>apic apis:clone</code>	Download all APIs from the catalog or space.

It can also be useful to clear all Products and their APIs from a Catalog, particularly for a development Catalog (you must provide the name of the Catalog as the value of the `--confirm` parameter):

```
apic products:clear --confirm catalog_name
```

where `catalog_name` is the name of the Catalog.

You can use a *Space* to partition a Catalog so multiple teams can manage Products and APIs independently in a single Catalog. A Space is conceptually like a sub-catalog, except that Products and APIs in all Spaces in a given Catalog are published to the same developer portal. For more information about Spaces, see [Using syndication in IBM API Connect](#).

To manage the Products and APIs that have been published to a Space, include the `--scope space` option with the `apic products` and `apic apis` commands. For example, to list the Products that are contained in a Space called `flights`, use the following command:

```
apic products --scope space --space flights --catalog production --org climbonorg --server platform-api.myserver.com
```

## Changing the lifecycle state of a Product in a Catalog or Space

To directly change the lifecycle state of a Product that has previously been staged or published to a Catalog or Space, complete the following steps:

- Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic products:update product_name:version --server mgmt_endpoint_url --org organization --scope scope --catalog catalog [--space space] -
```

where:

- `product_name` is the name of the Product whose lifecycle state you want to change.
- `version` is the version of that Product.
- `mgmt_endpoint_url` is the platform API endpoint URL.
- `organization` is the name of the provider organization in which the Product was previously staged or published.
- `scope` has one of the following values:
  - `catalog` if the Catalog does not have Spaces enabled.
  - `space` if the Catalog has Spaces enabled. If you specify `space` for the `--scope` parameter you must also supply the `--space` parameter.
- `catalog` is the name of the Catalog in which the Product was previously staged or published.
- (optional) `space` is the name of the Space. The `--space` parameter is required if the Catalog has Spaces enabled, in which case you must also include `--scope space` in the command.

The command returns:

```
Reading PRODUCT_FILE arg from stdin
```

- Enter the following data, followed by a new line:

```
state: new_state
```

where `new_state` is the state that you want to change the Product to, and must have one of the following values.

- `staged`
  - `published`
  - `deprecated`
  - `retired`
  - `archived`
- Press **CTRL D** to terminate the input. If the command is successful, the lifecycle state change is confirmed. For example:

```
apic products:update finance:1.0.0 --server https://myserver.com --org development --scope catalog --catalog sandbox -
Reading PRODUCT_FILE arg from stdin
state: published
finance:1.0.0 [state: published] https://myserver.com/api/catalogs/dce12994-a6a1-487b-83b6-c73bd8498799/006827d5-
9e82-427a-abe6-be5b126210f7/products/0f0af980-f505-4f36-b09c-d7b1c9c1a2f2
```

Note:

The command will not succeed if the lifecycle state change you are attempting is not permitted.

For example, the following lifecycle state changes are permitted:

- staged to published
- deprecated to retired

The following lifecycle state changes are not permitted:

- published to staged
- retired to published

For full details, see [The Product lifecycle](#).

To list all Products in a Catalog or Space, together with their lifecycle states, use the following command:

```
apic products:list-all --server mgmt_endpoint_url --org organization --scope scope --catalog catalog [--space space]
```

For example:

```
apic products:list-all --server https://myserver.com --org development --scope catalog --catalog sandbox
graphql-services:1.0.0 [state: staged] https://myserver.com/api/catalogs/dce12994-a6a1-487b-83b6-c73bd8498799/006827d5-
9e82-427a-abe6-be5b126210f7/products/7652d568-b396-4bfa-bf71-2f18cea63737
finance:1.0.0 [state: published] https://myserver.com/api/catalogs/dce12994-a6a1-487b-83b6-c73bd8498799/006827d5-
9e82-427a-abe6-be5b126210f7/products/0f0af980-f505-4f36-b09c-d7b1c9c1a2f2
```

To find out the lifecycle state of a specific Product, use the following command:

```
apic products:get product_name:version --server mgmt_endpoint_url --org organization --scope scope --catalog catalog [--space
space] --fields state --output -
```

For example:

```
apic products:get finance:1.0.0 --server https://myserver.com --org development --scope catalog --catalog sandbox --fields
state --output -
state: published
```

## Example scripts

A set of example toolkit scripts that demonstrate how to create and manage organizations, users, apps, Products and APIs are available at <https://github.com/ibm-apiconnect/example-toolkit-scripts>.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with Drafts

Co-locate your APIs and applications in your local source code control systems to support typical development activities such as commits, branching, merges, continuous integration, and so on. The developer toolkit provides the bridge from the developer's environment to the IBM® API Connect runtime services.

API Connect provides an online development capability called *Drafts* where you can define API and Product definitions. The `apic drafts` commands enable synchronization of Product and API artifacts between local source code control systems and drafts.

You can use `drafts` to clear and list your drafts. For example:

```
apic config:set catalog=https://platform-api.myserver.com/api/catalogs/climbon/sandbox # set the default Catalog
apic login --username some-user --password some-password --server platform-api.myserver.com --realm provider/myldap # login
into the management server

apic create:api --title routes --product ClimbOn
apic drafts # list what is in drafts
apic drafts:clear --confirm drafts # clear drafts collection
apic drafts:list # list the available drafts
```

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Reading input from the command line

If a developer toolkit command takes a file as an input parameter, you can direct the command to read the input directly from the command line rather than supplying a separate file; this can be useful when writing scripts to automate command line operations, for example.

## Reading input from a file

Many developer toolkit commands that create or update new objects in API Connect take a file as an input parameter. For example:

- Register a new gateway service:  

```
apic gateway-services:create [flags] GATEWAY_SERVICE_FILE
```
- Update a provider organization:  

```
apic orgs:update [flags] ORG_FILE
```
- Create a new Catalog:  

```
apic catalogs:create [flags] CATALOG_FILE
```
- Create a new consumer organization:  

```
apic consumer-orgs:create [flags] CONSUMER_ORG_FILE
```
- Register a new developer application:  

```
apic apps:create [flags] APP_FILE
```
- Create a new email server:  

```
apic mail-servers:create [flags] MAIL_SERVER_FILE
```
- Update the list of applications that subscribe to a Product:  

```
apic subscriptions:update [flags] SUBSCRIPTION_FILE
```

You can format the input file as a YAML file (the default) or as a JSON file. To use a JSON file as input, include the `--format json` parameter.

## Reading input from the command line

As an alternative to supplying an explicit file name to the command, you can supply a hyphen (-) in place of the file name parameter; the command will then read the file content directly from the command line as shown in the following examples.

- Register a new DataPower® Gateway (v5 compatible) gateway service:  

```
apic gateway-services:create --server platform-api.myserver.com --org admin --availability-zone availability-zone-default -
Reading GATEWAY_SERVICE_FILE arg from stdin
name: dpgw-service
title: DataPower gateway service, compatible with v5
gateway_service_type: datapower-gateway
endpoint: 'https://mygwhost.com:3000'
api_endpoint_base: 'https://mygwhost.com:9443'
sni:
  - host: '*'
    tls_server_profile_url: https://platform-api.myserver.com/api/orgs/75203636-f038-4287-a732-24af4bf7059d/tls-server-
profiles/3c0a0e93-6aa4-4288-b09c-eccf4901b104
visibility:
  type: public
```

Note: You can obtain the `tls_server_profile_url` property value for a TLS server profile by using the following command, which lists the URLs for all TLS server profiles:

```
apic tls-server-profiles --org organization_name --server mgmt_endpoint_url
```

- Create a new provider organization:  

```
apic orgs:create --server platform-api.myserver.com -
Reading ORG_FILE arg from stdin
name: development
title: Development organization
owner_url: https://platform-api.myserver.com/api/user-registries/1bbbd414-22f1-47cf-8eac-2050530d29a7/c082d755-866e-4394-
959f-fbb264c6c3a1/users/2db491c7-2d0f-4f60-a8ae-f38d07481f21
```

Note: You can obtain the `owner_url` property value by using the following command, which returns the `url` property for a specific user:

```
apic users:get username --user-registry user_registry_name --server mgmt_endpoint_url --org organization_name --fields url
--output -
```
- Create a new Catalog:  

```
apic catalogs:create --server platform-api.myserver.com --org myorg -
Reading CATALOG_FILE arg from stdin
name: production
title: Production Catalog
summary: Catalog containing APIs in production use
```
- Create a new consumer organization:  

```
apic consumer-orgs:create --server platform-api.myserver.com --catalog sandbox --org myorg -
Reading CONSUMER_ORG_FILE arg from stdin
name: finance-apps
title: Developers of finance applications
```

```
owner_url: https://platform-api.myserver.com/api/user-registries/a1fb8159-fda1-410f-93ab-a0ea5fd04535/70ce7ec3-e83f-4c87-9f3c-4662ad108bbc/users/fddd5df5-c178-4a34-ab92-0a48344d5c9b
```

Note: You can obtain the `owner_url` property value by using the following command, which returns the `url` property for a specific user:

```
apic users:get username --user-registry user_registry_name --server mgmt_endpoint_url --org organization_name --fields url --output -
```

- Register a new developer application:

```
apic apps:create --server consumer-api.myserver.com --org my-consumer-org --mode consumer -
Reading APP_FILE arg from stdin
name: finance
title: Finance application
summary: Mobile app for personal finance management
```

- Create a new email server:

```
apic mail-servers:create --org admin --server platform-api.myserver.com -
Reading MAIL_SERVER_FILE arg from stdin
title: My email server
name: my-email-server
host: smtp.myemail.com
port: 20
credentials:
  username: me@myemail.com
  password: password
```

- Create a new subscription to subscribe an application to a Product:

```
apic subscriptions:create --app myapp --catalog sandbox --org myorg --server platform-api.myserver.com --consumer-org my-consumer-org -
Reading SUBSCRIPTION_FILE arg from stdin
product_url: https://platform-api.myserver.com/api/catalogs/960733c1-d7f7-4b90-9bc7-69dbf4ce31/ca34537d-adf4-4320-8495-a7feaa62d679/products/638015c6-a8b0-452e-841f-2ac441ab6962
plan: default-plan
```

Note: You can obtain the `product_url` property value for a Product by using the following command, which lists the URLs for all versions of a Product given the Product name:

```
apic products:list product_name --scope catalog --catalog catalog_name --org provider_org_name --server mgmt_endpoint_url
```

---

## Related concepts

- [Overview of the command-line tool](#)

---

## Related reference

- [Logging in to a management server](#)

---

## Related information

- [Command-line tool reference](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Scripting with the toolkit commands

The IBM® API Connect developer toolkit provides commands for cloud administration and API development and management.

---

## Scripting commands

It's often helpful to automate a series of `apic` commands in a shell script. Since the `apic` tool first requires you to interactively accept the license, you must first use the following command:

```
apic --accept-license
```

Once you do that, your scripts can run non-interactively.

Toolkit command scripts allow you to automate many of the tasks required to set up and maintain your API products. A set of example scripts are available at <https://github.com/ibm-apiconnect/example-toolkit-scripts>.

These scripts demonstrate how to perform the following tasks.

- Create a Provider Organization
- Configure a catalog to use one or more gateways
- Create draft APIs and Products
- Publish Products
- Replace an existing published product with a new version
- Delete a product

- Create a Consumer Organization
- Create a new consumer app
- Subscribe the new app to an existing published product
- Stage a new product version
- Supersede the existing published product with the new one
- Migrate existing subscriptions from the old product to the new product

## Related information

- [Command-line tool reference for the developer toolkit](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with OpenAPI extensions

Use the developer toolkit CLI **extensions** commands to manage OpenAPI extensions in your Catalogs or Spaces.

You can extend the OpenAPI specification by adding either a JSON or YAML extension schema to an API, depending on the version of IBM® API Connect you are using. An extension is imported into a Catalog, or as Space in a Catalog, then added to the API schema.

The commands described here are for importing OpenAPI extensions into a Catalog or Space, and viewing and managing them. After you have imported an OpenAPI extension, you can reference it in your API definitions; for details, see [Referring to an extension in an API definition](#).

Note: If Spaces are enabled in a Catalog, an OpenAPI extension that you import into one Space is imported into **all** Spaces; you cannot import an OpenAPI extension into an individual Space in the Catalog. Any subsequent updates are also applied to all Spaces.

Command name	Action	Syntax
<b>extensions clear</b>	<p>The <b>extensions:clear</b> command deletes all versions of an extension given the extension name.</p> <p>The parameters are as follows:</p> <p><b>--scope</b> Specifies whether you want to clear the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p><b>--catalog</b> or <b>-c &lt;catalog_name&gt;</b> Specifies a single Catalog with the Catalog name.</p> <p><b>--space</b> Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p><b>--configured-gateway-service &lt;service_name&gt;</b> Specifies the name of the gateway service.</p> <p><b>--confirm &lt;catalog_name&gt;</b> Confirms the clear of the extension.</p> <p><b>&lt;extension_name&gt;</b> Specifies the name of the existing extension. Note: This is the extension name, not the file name.</p> <p><b>--org</b> or <b>-o &lt;organization_name&gt;</b> Specifies a single organization with the organization name.</p> <p><b>--server</b> or <b>-s &lt;management_server_endpoint&gt;</b> Specifies the server endpoint.</p>	<pre>apic extensions:clear --scope catalog space --catalog_  -c catalog_name --server   -s management_server_endpoint --configured-gateway-service gw_service_name --confirm catalog_name --org_  -o org_name [--space space]</pre> <p>Example:</p> <pre>apic extensions:clear --scope catalog --catalog catalog1 extension1 --server endpoint1 --configured-gateway-service gw_service1 --confirm catalog1 --org my_org</pre> <p>This example deletes all versions of the <b>extension1</b> extension that are in <b>catalog1</b> of the <b>my_org</b> organization, and are paired with <b>endpoint1</b>.</p>

Command name	Action	Syntax
<b>extensions clear-all</b>	<p>The <b>extensions:clear-all</b> command deletes all of the extensions in the specified Catalog or Space.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clear the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p> <p>--confirm &lt;catalog_name&gt; Confirms the clear of the extension.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p>	<pre>apic extensions:clear-all --scope catalog space --catalog -c catalog_name --server -s management_server_endpoint --configured-gateway-service gw_service_name --confirm catalog_name --org -o org_name [--space space]</pre> <p>Example:</p> <pre>apic extensions:clear-all --scope catalog --catalog catalog1 --server endpoint1 --configured-gateway-service gw_service1 --confirm catalog1 --org my_org</pre> <p>This example deletes all the extensions that are in <b>catalog1</b> of the <b>my_org</b> organization, and are paired with <b>endpoint1</b>.</p>
<b>extensions clone</b>	<p>The <b>extensions:clone</b> command creates a local definition file for each extension.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clone the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>org</b> Sets the scope to the organization.</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p>	<pre>apic extensions:clone --scope catalog space --catalog -c catalog_name --org -o org_name [-- space space] --server -s management_server_endpoint --configured- gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:clone --scope catalog --catalog catalog1 --org my_org --server endpoint1 --configured- gateway-service gw_service_1</pre> <p>This example clones the extensions that are in <b>catalog1</b> of the <b>my_org</b> organization, and are paired with <b>endpoint1</b>.</p>



Command name	Action	Syntax
<b>extensions create</b>	<p>The <b>extensions:create</b> command creates an extension in a Catalog or Space, given an extension definition file.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clear the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>&lt;extension_file&gt; Specifies the name of the extension file that contains the new information for your extension.</p> <p>Note: This is the OpenAPI file name, not the name of the extension.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p>	<pre>apic extensions:create --scope catalog space extension_file --server   -s management_server_endpoint --catalog   -c catalog_name [--space space] --org   -o org_name --configured-gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:create --scope catalog filename.yaml -- server endpoint1 --catalog catalog1 --org my_org --configured-gateway-service gw_service_1</pre> <p>This example creates an extension in <b>catalog1</b> of the <b>my_org</b> organization, paired with <b>endpoint1</b>, using the <b>filename.yaml</b> OpenAPI extension definition file.</p>
<b>extensions delete</b>	<p>The <b>extensions:delete</b> command deletes a specific version of an extension.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clear the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>&lt;extension_name&gt;:&lt;version_number&gt; Specifies the name and version number of the extension. Note: <i>extension_name</i> is the extension name, not the file name.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p>	<pre>apic extensions:delete --scope catalog space extension_name:version_number --catalog   -c catalog_name --org   -o org_name [-- space space] --server   -s management_server_endpoint --configured- gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:delete --scope catalog myextension:1.0.0 --catalog catalog1 --org orgmain --server endpoint1 --configured-gateway- service gw_service_1</pre> <p>This example deletes <b>myextension</b> version <b>1.0.0</b> that is in <b>catalog1</b> of the <b>orgmain</b> organization, and is paired with <b>endpoint1</b>.</p>

Command name	Action	Syntax
<b>extensions get</b>	<p>The <b>extensions:get</b> command retrieves the definition file for a specific version of an extension.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clone the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>org</b> Sets the scope to the organization.</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>&lt;extension_name&gt;:&lt;version_number&gt; Specifies the name and version number of the extension. Note: <i>extension_name</i> is the extension name, not the file name.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p>	<pre>apic extensions:get --scope catalog space extension_name:version_number --catalog   -c catalog_name --org   -o org_name [-- space space] --server   -s management_server_endpoint --configured- gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:get --scope catalog myextension:1.0.0 - -catalog catalog1 --org orgmain --server endpoint1 --configured-gateway- service gw_service_1</pre> <p>This example gets <b>myextension</b> version 1.0.0 that is in <b>catalog1</b> of the <b>orgmain</b> organization, and is paired with <b>endpoint1</b>.</p>
<b>extensions list</b>	<p>The <b>extensions:list</b> command lists all versions of an extension given the extension name.</p> <p>The parameters are as follows:</p> <p>--scope Specifies whether you want to clone the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>org</b> Sets the scope to the organization.</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p>--server or -s &lt;management_server_endpoint&gt; Specifies the server endpoint.</p> <p>--catalog or -c &lt;catalog_name&gt; Specifies a single Catalog with the Catalog name.</p> <p>--space Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p>&lt;extension_name&gt; Specifies the name of the existing extension. Note: This is the extension name, not the file name.</p> <p>--org or -o &lt;organization_name&gt; Specifies a single organization with the organization name.</p> <p>--configured-gateway-service &lt;service_name&gt; Specifies the name of the gateway service.</p>	<pre>apic extensions:list --scope catalog space extension_name --catalog   -c catalog_name --org   -o org_name [-- space space] --server   -s management_server_endpoint --configured- gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:list --scope catalog my_extension -- -catalog catalog1 --org my_org --server endpoint1 --configured-gateway-service gw_service_1</pre> <p>This example lists all versions of the extension <b>my_extension</b> that are paired with <b>endpoint1</b>, are in <b>catalog1</b>, and in the <b>my_org</b> organization.</p>

Command name	Action	Syntax
<b>extensions list-all</b>	<p>The <b>extensions:list-all</b> command lists all of the extensions that are available in a Catalog or Space. This is the default command if you enter only <b>apic extensions</b>.</p> <p>The parameters are as follows:</p> <p><b>--scope</b> Specifies whether you want to clone the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>org</b> Sets the scope to the organization.</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p><b>--server</b> or <b>-s</b> <i>&lt;management_server_endpoint&gt;</i> Specifies the server endpoint.</p> <p><b>--catalog</b> or <b>-c</b> <i>&lt;catalog_name&gt;</i> Specifies a single Catalog with the Catalog name.</p> <p><b>--space</b> Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p><b>--org</b> or <b>-o</b> <i>&lt;organization_name&gt;</i> Specifies a single organization with the organization name.</p> <p><b>--configured-gateway-service</b> <i>&lt;service_name&gt;</i> Specifies the name of the gateway service.</p>	<pre>apic extensions:list-all --scope catalog space --catalog -c catalog_name --org -o org_name [--space space] --server -s management_server_endpoint --configured-gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:list-all --scope catalog --catalog catalog1 --org my_org --server endpoint1 --configured-gateway-service gw_service1</pre> <p>This example lists the extensions that are paired with <b>endpoint1</b>, are in <b>catalog1</b>, and in the <b>my_org</b> organization.</p>
<b>extensions update</b>	<p>The <b>extensions:update</b> command replaces the information of an existing version of an extension with the information in an external file.</p> <p>The parameters are as follows:</p> <p><b>--scope</b> Specifies whether you want to clear the extensions in a Catalog or Space. The following options are available:</p> <p><b>catalog</b> The command is to be applied to Catalog</p> <p><b>space</b> The command is to be applied to a Space in a Catalog. When Spaces are enabled in a Catalog, you must set the scope to <b>space</b> and supply the <b>--space</b> parameter.</p> <p><b>--server</b> or <b>-s</b> <i>&lt;management_server_endpoint&gt;</i> Specifies the server endpoint.</p> <p><b>--catalog</b> or <b>-c</b> <i>&lt;catalog_name&gt;</i> Specifies a single Catalog with the Catalog name.</p> <p><b>--space</b> Specifies the name of a Space in a Catalog. The <b>--space</b> parameter is required if the Catalog has Spaces enabled, in which case you must also include <b>--scope space</b> in the command.</p> <p><b>--org</b> or <b>-o</b> <i>&lt;organization_name&gt;</i> Specifies a single organization with the organization name.</p> <p><b>--configured-gateway-service</b> <i>&lt;service_name&gt;</i> Specifies the name of the gateway service.</p> <p><b>&lt;extension_name&gt;:&lt;version_number&gt;</b> Specifies the name and version number of the extension. Note: <i>extension_name</i> is the extension name, not the file name.</p> <p><b>&lt;extension_file&gt;</b> Specifies the name of the extension file that contains the new information for your extension. Note: This is the OpenAPI file name, not the name of the extension.</p>	<pre>apic extensions:update --scope catalog space extension_name:version_number extension_file --catalog -c catalog_name --org -o org_name [--space space] --server -s management_server_endpoint --configured-gateway-service gw_service_name</pre> <p>Example:</p> <pre>apic extensions:update --scope catalog extension1:1.0.0 extension2.yaml --catalog catalog1 --org my_org --server endpoint1 --configured-gateway-service gw_service1</pre> <p>This example updates <b>extension1</b> version <b>1.0.0</b> that is paired with <b>endpoint1</b>, in <b>catalog1</b>, and in the <b>my_org</b> organization, from the file <b>extension2.yaml</b>.</p>

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Viewing application performance metrics

IBM® API Connect provides two ways to view application performance metrics for Node.js applications running locally: the built-in metrics dashboard and sending the data to third-party consoles or log files.

To view application performance metrics, you have to run your app with `apic start`. The metrics dashboard is available automatically. To use third-party consoles, you need to set an environment variable and have the console configured properly.

- [Viewing the application metrics dashboard](#)  
You can view the application performance metrics for a Node application by using IBM API Connect.
- [Viewing application metrics using third-party consoles](#)  
You can monitor your LoopBack® applications by obtaining metrics data. You can send the metrics data to a third-party console, a log file, or syslog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Viewing the application metrics dashboard

You can view the application performance metrics for a Node application by using IBM® API Connect.

### Before you begin

---

To view application metrics for a Node application, the application project must either have:

- A main script file named `server.js`, `app.js`, or `index.js`.
- Or a `package.json` file that has a `main` property specifying the main application script file, for example:

```
"main": "server/myserver.js",
```

LoopBack application projects created with `apic loopback` automatically meet this requirement.

### About this task

---

When you run an application locally, you can view application performance metrics based on [Node application metrics](#) in the application metrics dashboard.

To view application metrics:

### Procedure

---

1. In the application project root directory, enter this command to run the application:

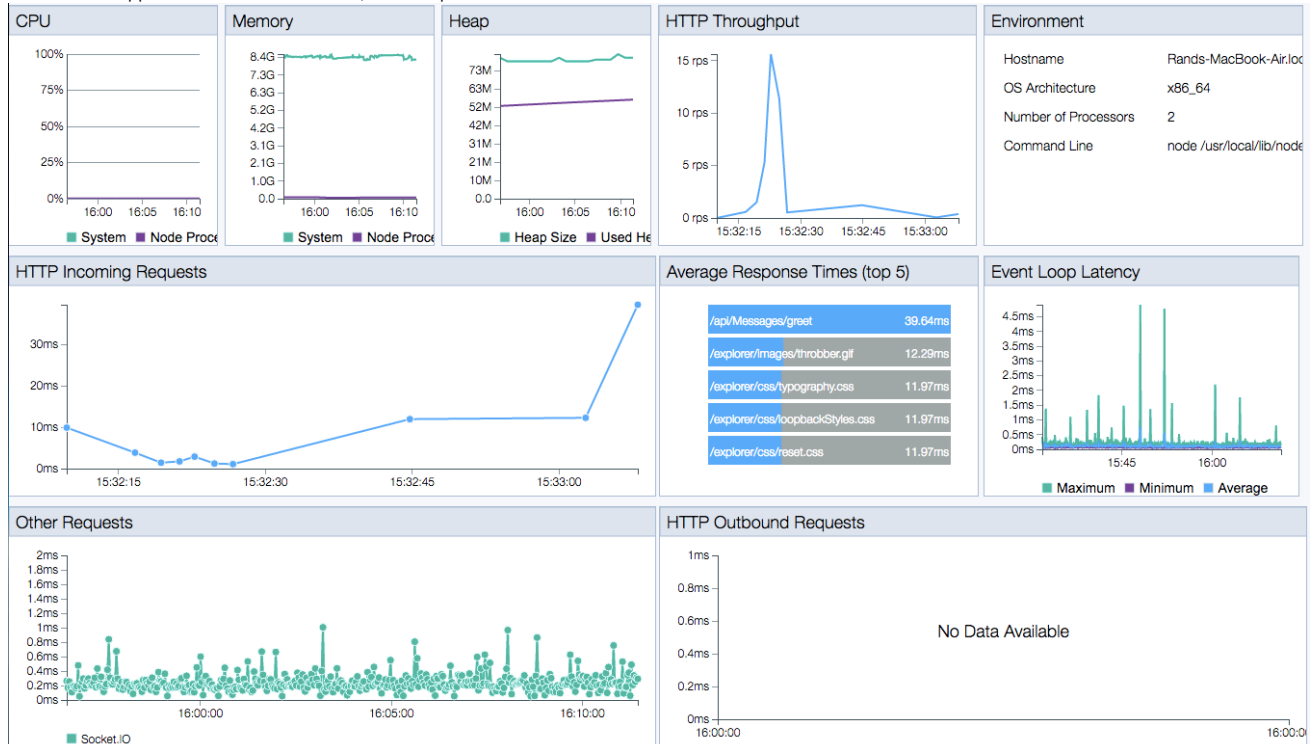
```
$ apic start
```

The console will display the following message:

```
Service <project-name> started on port 4001. Access the application dashboard at http://127.0.0.1:4001/appmetrics-dash  
Service <project-name> started on port 4002.
```

2. Open your browser to <http://127.0.0.1:4001/appmetrics-dash>.

You'll see the application metrics dashboard; for example:



The metrics are:

- **CPU** - Percentage of CPU time spent on the Node process.
- **Memory** - Amount of memory used by the Node process and available in the system.
- **Heap** - Amount of heap memory used and available.
- **HTTP Throughput** (requests per second) versus time.
- **HTTP Incoming Requests** - Response time for HTTP requests versus time.
- **Average Response Time** for the top five routes.
- **Event Loop Latency** (minimum / maximum / average) - Amount of time spent for an event loop "tick."
- **Other Requests**
- **HTTP Outbound Requests**

For more information on these metrics, see [Understanding the Application Metrics for Node.js dashboard](#).

Note: Node Report diagnostics are not currently available. The Node Report button is not operational in this release.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Viewing application metrics using third-party consoles

You can monitor your LoopBack® applications by obtaining metrics data. You can send the metrics data to a third-party console, a log file, or syslog.

You enable monitoring for an application by setting the **STRONGLOOP\_METRICS** environment variable. The value of the environment variable is a metrics URL that specifies the destination for the metrics data.

Enter the following command:

```
apic props:set STRONGLOOP_METRICS=metrics_url --remote --service app_name --organization org_name --server management_cluster_hostname_or_address
```

where:

- *metrics\_url* is the metrics URL that specifies the logging destination.
- *app\_name* is the value of the Name property of the App that references the collective to which the application is published.
- *org\_name* is the provider organization to which the application is published.
- *management\_cluster\_hostname\_or\_address* is the IP address or host name of the Management cluster to which the application is published.

The following sections provide details of the possible logging destinations.

## StatsD

StatsD is a simple protocol for log information together with a simple daemon (server) that aggregates and summarizes application metrics. The client communicates with the StatsD server by using the StatsD protocol, and the daemon then generates aggregate metrics and relays them to a graphing or monitoring backend. For more information on StatsD, see [StatsD, what it is and how it can help you](#).

The [StatsD](#) Node server has the following capabilities:

- Includes a built-in [Graphite](#) backend that can be local or [hosted](#).
- Supports other [backends](#) such [Zabbix](#), [DataDog](#), and so on.
- Supports custom backends.

Other metrics consumers, such as [DataDog](#), have agents that support the StatsD-protocol.

To use StatsD, set a metrics URL of the following form:

```
statsd://[host[:port]][/scope]
```

where:

- *host* is the name of the host where the StatsD server is running; the default value is `localhost`.
- *port* is the TCP port that the StatsD server is using; the default value is `8125`.
- *scope* is a string to scope or identify metrics; for example this might be the name of the application or module.

For example:

```
apic props:set STRONGLOOP_METRICS=statsd://myhost:1234/app1 --remote --service myApp --organization myOrg --server myhost.com
```

Example output:

```
my-app.cpu.user:0.00907|g
my-app.cpu.system:0.01664|g
my-app.cpu.total:0.0257|g
my-app.heap.used:10698193|g
my-app.heap.total:27423366|g
my-app.loop.count:127|c
```

## Hosted Graphite

---

To use Graphite, set a metrics URL of the following form:

```
graphite://[host[:port]]
```

where:

- *host* is the name of the host where the Graphite server is running; the default value is `localhost`.
- *port* is the TCP port the Graphite server is using; the default value is `2003`.

The metrics data is forwarded to hosted Graphite.

For example:

```
apic props:set STRONGLOOP_METRICS=graphite://myhost.com:1234 --remote --service myApp --organization myOrg --server myhost.com
```

## Splunk

---

To use Splunk, set a metrics URL of the following form:

```
splunk://[host]:port
```

where:

- *host* is the name of the host where the Splunk server is running; the default value is `localhost`.
- *port* is the TCP port that the Splunk server is using; you must provide a value because the protocol has no assigned port.

The metrics data is written to Splunk by using a UDP key-value protocol

For example:

```
apic props:set STRONGLOOP_METRICS=splunk://myhost.com:1234 --remote --service myApp --organization myOrg --server myhost.com
```

## Log file

---

To send metrics information to a log file, set a metrics URL of the following form:

```
log:[file]
```

where *file* is the name of the log file that you want to send the metrics information to. If you omit the file name, metrics information is sent to the console (stdout).

For example:

```
apic props:set STRONGLOOP_METRICS=log:myapp.log --remote --service myApp --organization myOrg --server myhost.com
```

## Syslog

---

To write metrics information using syslog, set a metrics URL of the following form:

```
syslog:[?[application=appName] [&priority=level]]
```

where:

- *appName* is any string; the default value is `statsd`.
- *level* is any of the following values:
  - LOG\_DEBUG
  - LOG\_INFO (the default)

- LOG\_NOTICE
- LOG\_WARNING
- LOG\_CRIT


For example:

```
apic props:set STRONGLOOP_METRICS=syslog:?application=myApp&priority=LOG_WARNING --remote --service myApp --organization myOrg --server myhost.com
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## [Technical Preview] Searching for items in API Manager

Use the search feature in IBM® API Connect API Manager to easily locate items such as APIs, Catalogs, applications, and subscriptions.

Search is available from any page in API Manager and can be accessed by typing in the search field in the page banner, or by clicking  Search in the navigation list.

### This feature is provided as a Technical Preview

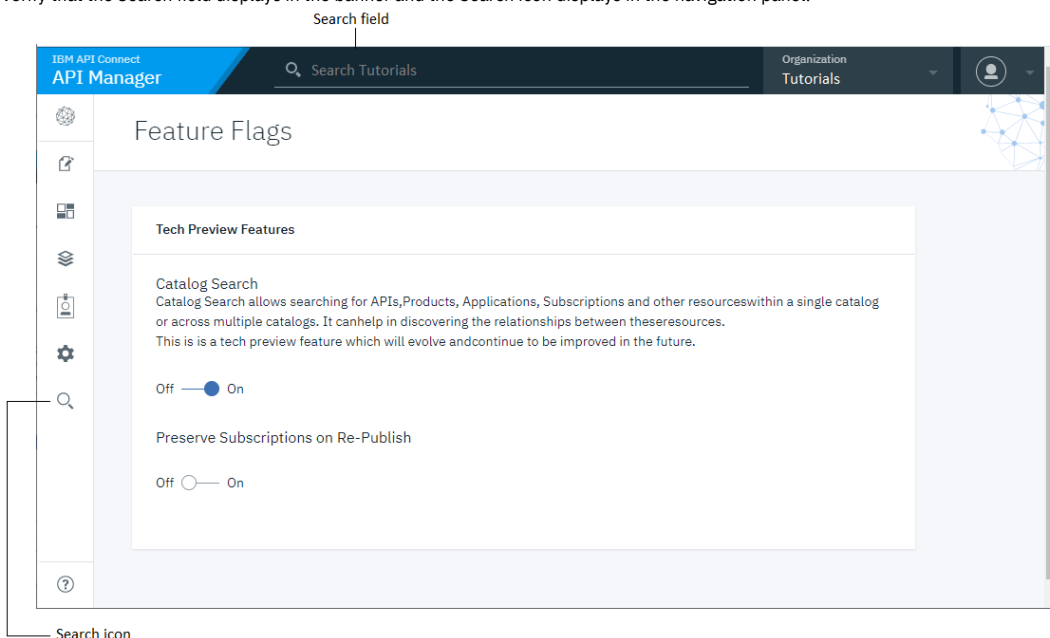
As such, the feature might not be fully functional in your environment and support is limited. In addition, the feature's appearance and functionality are subject to change both in this release (V2018) and in later releases of API Connect. Review the following disclaimer before enabling the Search technical preview:

Notice: Technology Preview Code (TPC) may be included or distributed with the Program or updates to it but are not part of the Program. TPC is licensed under the same terms as the Program, except as provided below. TPC will be identified as such in the Notices File (or in an updated Notices File accompanying the updates). Some or all of the TPC may not be made generally available by IBM as or in a product. Licensee is permitted to use TPC only for internal use for evaluation purposes and not for use in a production environment. The Notices File may limit this evaluation use to an evaluation period. If so, at the end of such evaluation period Licensee must cease using and uninstall the TPC. IBM provides the TPC without obligation of support and "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF TITLE, NON-INFRINGEMENT OR NON-INTERFERENCE AND ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.


### Enabling Search

Enable Search for your API Connect user account by completing the following steps:

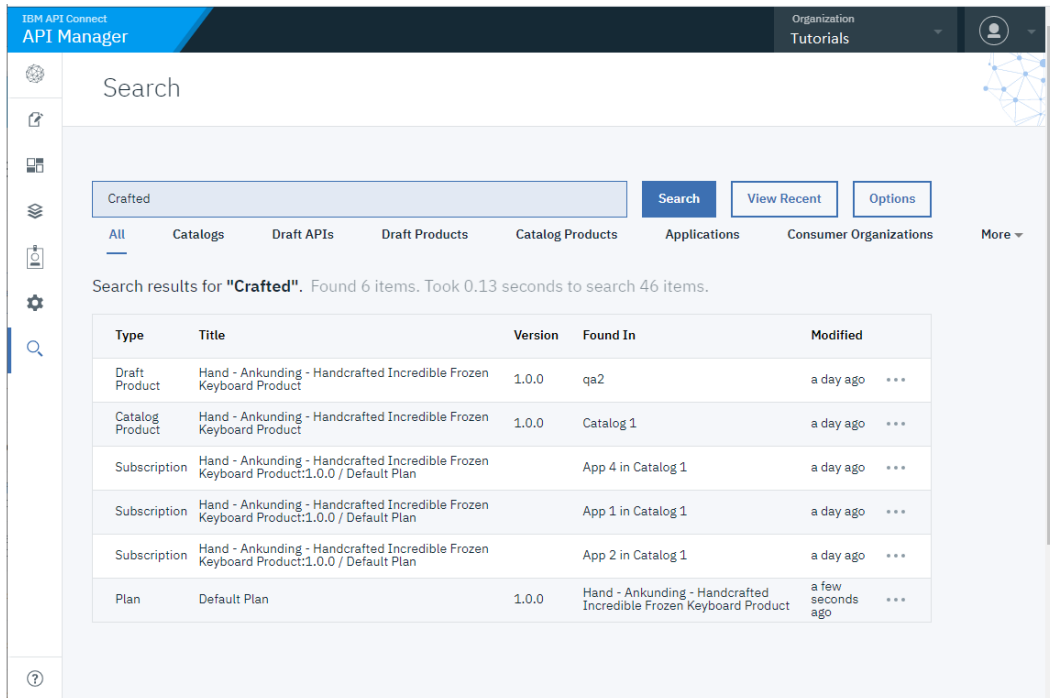
1. Log in to API Manager.
2. In the browser's address bar, append `/ff` to the URL and press Enter to display the Feature Flags page.  
For example: `https://example.com/my-org/ff`
3. On the Feature Flags page, click the Off - On toggle for the "Catalog Search" option to set it to On.
4. When the "Enable Catalog Search" message displays, click Confirm.
5. Verify that the Search field displays in the banner and the Search icon displays in the navigation panel.



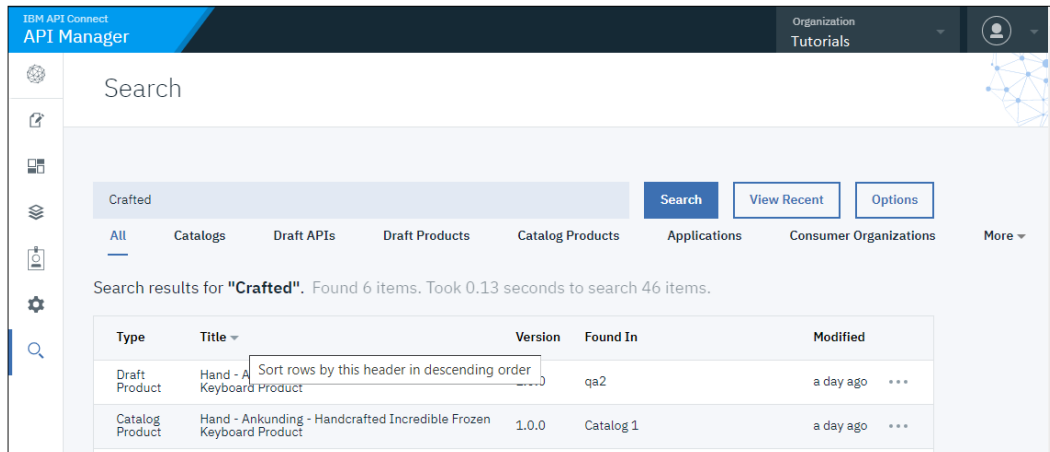
### Using Search

The Search feature is available on all pages of the API Manager. To start searching, just type a string into the Search field and press Enter. Whenever you press Enter in the Search field, the Search page displays with results and additional options. If you press Enter in an empty Search field, or click  Search in the navigation list, then the Search page displays a list all items. If you type a string before pressing Enter, the page displays the results from the corresponding search.

Search results include all items that contain the search string in a text field. For example, suppose you searched for "Crafted". The results might look like the following image, with items that don't match the search string exactly, but do include the string in a text field such as a title or a name:

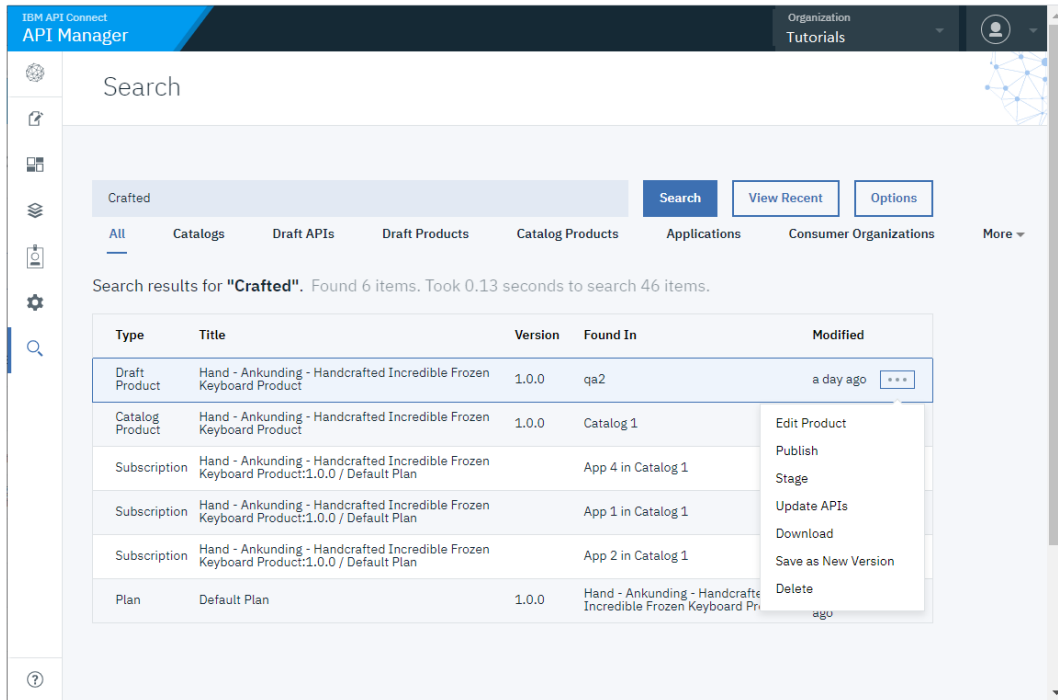


By default, the search returns all types of items that contain the string. Sort results on a particular column by clicking the column header.



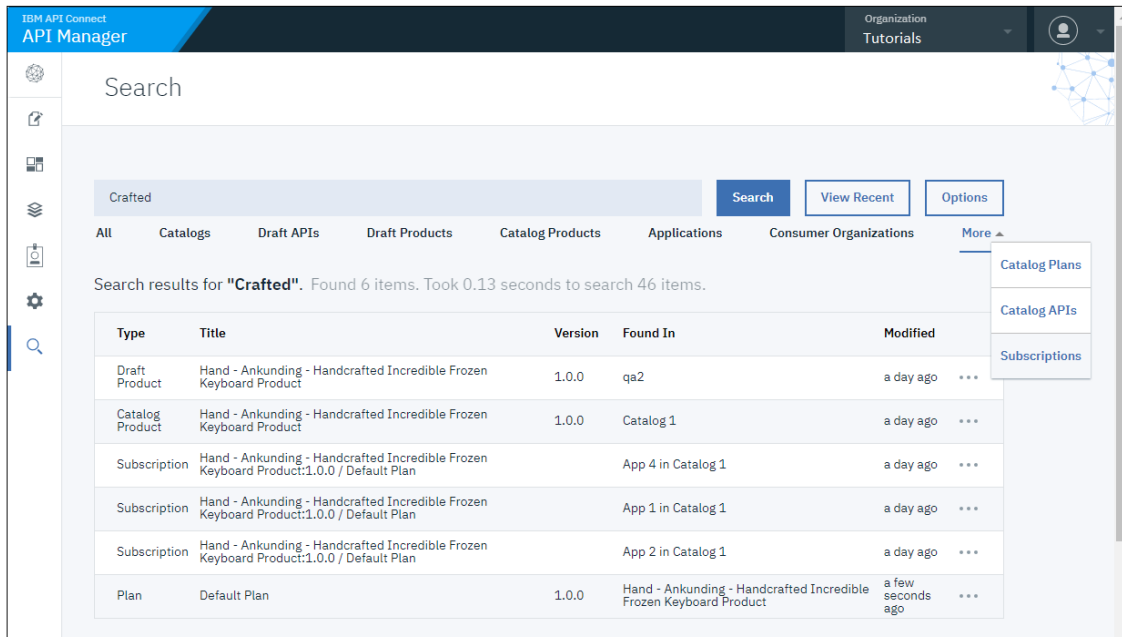
When you find an item you want to work with, click the ... actions menu next to the item's "Modified" date to see what you can do. The list of possible actions depends on the item's type.



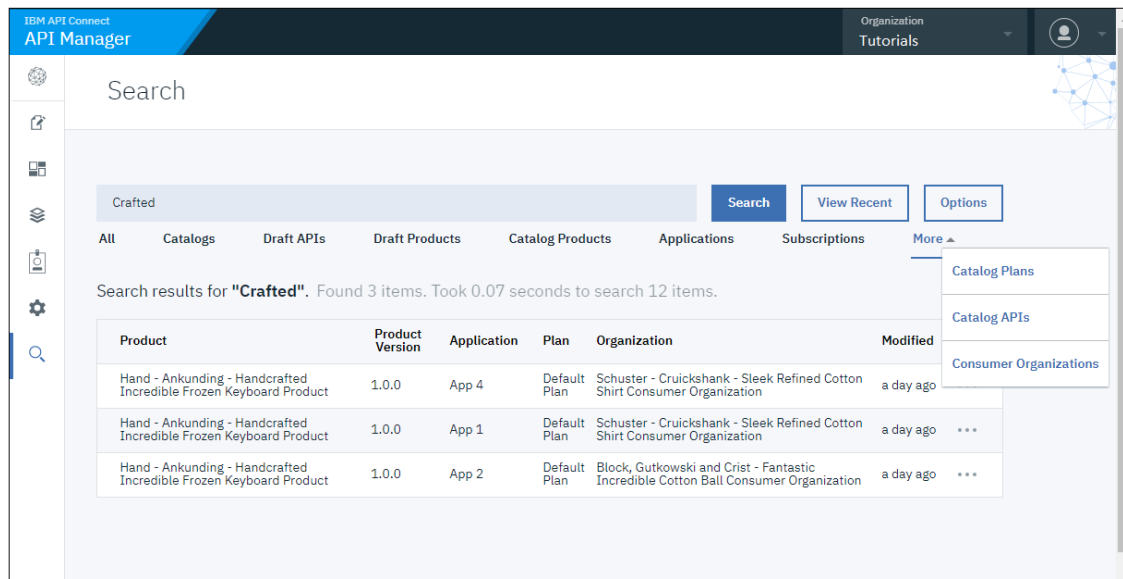


You can limit the search results by selecting an item type from the list that displays before the results. Click More to view additional types. If you select a type from the More menu, it replaces a type in the list.

In the following example, the results of the search for "Crafted" are filtered with the Subscriptions type.



Notice that "Subscriptions" now displays in the default list, and that "Consumer Organizations" moved to More menu.



If you want to refine the search further, click Options. On the Options panel, click Catalogs to Search and select which catalogs to search (the default is all catalogs). You can select up to 5 catalogs for a limited search.

You can also choose whether only exact matches (the complete string) are included in results, or items containing similar strings are also included.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with API definitions

An API definition specifies the complete configuration for an API. You can create and configure your APIs either by using either the API Designer UI application, or by using the browser based API Manager UI.

The following topics provide full details for creating and configuring your APIs.

- [Creating an API definition](#)  
You can create API definitions by using the API Manager or the command line interface in IBM® API Connect.
- [Editing an API definition](#)  
Use this topic to specify configuration settings for an API. You can do this when you create the initial definition, or you can do it later by editing an existing definition.
- [Activating an API](#)  
After you have created an API definition you can activate it to make it available for testing.
- [Testing an API with the assembly test tool](#)  
IBM API Connect provides a test environment for you to ensure that your APIs are defined and implemented correctly.
- [Testing an API with the Local Test Environment](#)  
Use the Local Test Environment to test APIs on your local machine, without the need to connect to an API Connect management server. The Local Test Environment is a lightweight API Manager running on your local machine. It allows you to rapidly test APIs locally.
- [Staging an API](#)  
The graphical wizard provides an option that adds the API to a Product and stages the Product in a Catalog. When a Product is in the staged state, it is not yet visible to, or subscribable by, any developers. The syndication feature in IBM API Connect means that if Spaces are enabled for a Catalog, Products can be staged only to a Space within that Catalog.
- [Publishing an API](#)  
The graphical wizard provides an option that adds the API to a Product and publishes the Product in a Catalog. Publishing a draft Product makes the APIs in that Product visible on the Developer Portal for use by application developers. The syndication feature in IBM API Connect means that if Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog.
- [Downloading an API definition](#)  
You can download the details of your API definitions so that you can store them for later recovery.
- [Deleting an API definition](#)  
You can delete an API definition that is no longer required.
- [Using an options file when importing a WSDL service](#)  
When you create an API definition, or add a target WSDL service to an API definition, by importing a .zip file, you can specify additional directives by including an options file in the .zip file.
- [API policies and logic constructs](#)  
Policies and logic constructs are pieces of configuration that control a specific aspect of processing in the Gateway server during the handling of an API invocation at run time.
- [Tags in API Connect](#)  
There are a number of forms of tagging in IBM API Connect.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating an API definition

You can create API definitions by using the API Manager or the command line interface in IBM® API Connect.

### Before you begin

---

An API is a set of functions that provide some business or technical capability and can be called by applications by using a defined protocol. Applications are typically mobile or web applications, and they use the HTTP protocol. An API definition is composed of paths, and can be one of the following types:

#### REST API definition

A REST API is a defined set of interactions that uses the HTTP protocol, typically by using JSON or XML as the data format that is exchanged. For example, a data request might use an HTTP GET method, and a data record might use an HTTP POST method. The choice of data format depends on the type of application that is calling the API. JSON is commonly used for web pages or mobile applications that present a user interface (by using JavaScript or HTML), whereas XML is often used for machine-to-machine scenarios.

You can create REST APIs by using LoopBack® functions to create models and data sources. These are then used by your REST API definition and exposed to your users.

Alternatively, you can expose and secure your existing APIs by using a [Proxy](#) or [Invoke](#) policy.

In either case, you can configure your API definition either by using the API Manager, or by writing an OpenAPI definition file and publishing it using either API Manager or the command line interface.

#### SOAP API

You can create SOAP API definitions that are based on an existing Web Services Description Language (WSDL) file. You can use this facility to benefit from the capabilities that are provided by API Connect, which include analytics and mapping between variables. You can also expose the API by using the Developer Portal for any existing SOAP services in your organization, including any SOAP services that are part of a service-oriented architecture (SOA) or Enterprise Service Bus (ESB) infrastructure.

You can create SOAP API definitions through either the command line interface, or through API Manager.

### About this task

---

You can create definitions for REST APIs or SOAP APIs.

- In the API Manager, you can add a REST API definition either by composing the API definition, and its operations, from scratch, or by importing an OpenAPI definition file. You can also use the tooling to quickly create a proxy API that calls an existing endpoint.
- If you have an existing SOAP service that you want to expose more widely, you can add a SOAP API to IBM® API Connect. You can use the Developer Portal to publicize the SOAP service to the developers. If a developer wants to use the SOAP API, you can use API Connect to manage their sign-up and access to the service, and track the usage of that API.

### Procedure

---

1. Quickly create an API definition by using one of the following wizards:

- [Creating a REST proxy API from a target service](#)
- [Creating a REST proxy API from an existing OpenAPI service](#)
- [Creating a new REST OpenAPI definition](#)
- [Creating a REST proxy API from an existing WSDL service](#)
- [Adding a REST API by importing an OpenAPI definition file](#)
- [Creating a SOAP proxy API from an existing WSDL service](#)

2. After you create your API definition, you can define it in more detail by following the instructions in [Editing an API definition](#).

#### • [Creating a REST proxy API from a target service](#)

If you have an existing REST service that you want to expose through an IBM API Connect API definition, you can create a proxy API and specify the target endpoint by using the API Manager.

#### • [Creating a REST proxy API from an existing OpenAPI service](#)

If you have an existing OpenAPI described target service that you want to expose through an IBM API Connect API definition, you can create a proxy API and specify the target endpoint.

#### • [Creating a REST proxy API from an existing WSDL service](#)

If you have a SOAP service defined in a WSDL file, you can use the WSDL file to create an API Connect REST proxy API that calls that SOAP service.

#### • [Creating a SOAP proxy API from an existing WSDL service](#)

If you have a SOAP service defined in a WSDL file, you can use the WSDL file to create an API Connect SOAP proxy API that calls that SOAP service.

#### • [Creating a new REST OpenAPI definition](#)

You can create and edit draft REST API definitions by using the API Manager or the API Designer user interface in IBM API Connect.

#### • [Adding a REST API by importing an OpenAPI definition file](#)

You can use an OpenAPI definition file to import a REST API into IBM API Connect.

#### • [Adding a SOAP API by importing a zip file](#)

You can use an OpenAPI definition file to import a SOAP API into IBM API Connect.

### Related concepts

---

- [Using the developer toolkit command-line tool](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a REST proxy API from a target service

If you have an existing REST service that you want to expose through an IBM® API Connect API definition, you can create a proxy API and specify the target endpoint by using the API Manager.

### About this task




You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

Create proxy REST APIs in minutes by using the API Manager tool to create an API from target service. This procedure creates a REST proxy that routes all traffic to a target REST API or service endpoint.

### Procedure

To compose a proxy API from a target service, complete the following steps.

1. In the navigation pane, click  Develop, and then click Add  API.  
The Add API: Create screen is displayed.
2. Select From target service, and click Next.
3. Click Next. Specify the API summary in the Info screen. You can fine-tune the API after it is created.
  - The Title can include special characters but should be kept short so that it can be easily displayed in the user interface.
  - The Name is entered automatically. The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. To view the CLI commands to manage draft APIs, see [apic draft-apis](#).
  - The Version corresponds to the value of the `info.version` property of the API's OpenAPI definition. The `version.release.modification` version numbering scheme is recommended; for example `1.0.0`.
  - The Base path is the URL segment of the API and does not include the host name or any additional segments for paths or operations. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
  - The optional Description helps to identify the API.
4. Specify the existing REST API endpoint that you want to call in the Target Service URL field.
5. Click Next. In the Secure section, configure the API security that you require.
  - Secure using Client ID - Select this option to require an Application to provide a Client ID (API Key). This causes the `x-IBM-Client-Id` parameter to be included in the request header for the API. If selected, you can then select whether to limit the API calls on a per key (per Client ID) basis:
    - Limit API calls on a per key basis - If selected, you must configure the rate limit that you require. Rate limits control the maximum number of calls allowed within a time period (hour, minute, month or day). For example, 100 calls per hour.  
For information about security options in IBM API Connect, see [Configuring API security](#).
  - CORS - Select this option to enable cross-origin resource sharing (CORS) support for your API. This allows your API to be accessed from another domain.
6. Click Next to create your API definition.  
The Summary panel displays messages as the definition is created, and the selected security options and rate limits are enforced.
7. Select one of the following options:
  - To further configure your API, click Edit API. For details, see [Editing an API definition](#).
  - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

### Results

You created a proxy API from an existing target service.

### What to do next

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

### Related tasks

- [Defining Paths for a REST API](#)

### Related reference

- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a REST proxy API from an existing OpenAPI service

If you have an existing OpenAPI described target service that you want to expose through an IBM® API Connect API definition, you can create a proxy API and specify the target endpoint.

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.



To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

Create proxy REST APIs in minutes by using the API Manager tool to create an API from existing OpenAPI service. This procedure creates a REST proxy that routes all traffic to a target REST API or service endpoint.

### Procedure

---


To compose a proxy API from an existing OpenAPI service, complete the following steps.

1. In the navigation pane, click  Develop, and then click Add  API.  
The Add API: Create screen is displayed.
2. Select From existing OpenAPI service, and click Next.
3. To upload the service information from an OpenAPI file, you can either drag and drop your file, or browse and select the file that you want to use. Valid file types are YAML, YML, etc.
  - Click Browse and browse for the OpenAPI file.
  - Drag and drop the file into the active area.

After you upload the file, the OpenAPI specification is evaluated and a message displays whether it is valid.

4. Click Next. Specify the API summary in the Info screen. You can fine-tune the API after it is created.
  - The Title can include special characters but should be kept short so that it can be easily displayed in the user interface.
  - The Name is entered automatically. The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. To view the CLI commands to manage draft APIs, see [apic draft-apis](#).
  - The Version corresponds to the value of the **info.version** property of the API's OpenAPI definition. The **version.release.modification** version numbering scheme is recommended; for example 1.0.0.
  - The Base path is the URL segment of the API and does not include the host name or any additional segments for paths or operations. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
  - The optional Description helps to identify the API.
5. Click Next. In the Secure section, configure the API security that you require.
  - Secure using Client ID - Select this option to require an Application to provide a Client ID (API Key). This causes the **X-IBM-Client-Id** parameter to be included in the request header for the API. If selected, you can then select whether to limit the API calls on a per key (per Client ID) basis:
    - Limit API calls on a per key basis - If selected, you must configure the rate limit that you require. Rate limits control the maximum number of calls allowed within a time period (hour, minute, month or day). For example, 100 calls per hour.For information about security options in IBM API Connect, see [Configuring API security](#).
  - CORS - Select this option to enable cross-origin resource sharing (CORS) support for your API. This allows your API to be accessed from another domain.
6. Optional: (API Manager UI only) Select Activate API if you want to immediately use the API for further development and testing.

Note:

- When you select the Activate API option, API Connect automatically completes the following actions:
    - Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title *api\_title* auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#)
    - Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).
  - You cannot use the Activate API option if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to use the Activate API option they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
  - To use the Activate API option, you must be assigned a role that has the **Product:Manage** and **Subscription:Manage** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).
7. Click Next to create your API definition.  
The Summary panel displays messages as the definition is created, and the selected security options and rate limits are enforced.  
  
If you selected Activate API, the wizard populates an API Endpoint URL that you can use in testing. If you have also selected Secure using Client ID, the wizard displays a Client ID and Client Secret you can use.
  8. Select one of the following options:
    - To further configure your API, click Edit API. For details, see [Editing an API definition](#).
    - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

### Results

---

You created a proxy API from an existing OpenAPI service.

## What to do next

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

## Related tasks

- [Defining Paths for a REST API](#)

## Related reference

- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Creating a REST proxy API from an existing WSDL service

If you have a SOAP service defined in a WSDL file, you can use the WSDL file to create an API Connect REST proxy API that calls that SOAP service.

## About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.


To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

If the WSDL file is a stand-alone file with no external dependencies, you can load the .wsdl file from a directory to create the REST API definition.

If the WSDL file references other WSDL files or references XSD files containing XML schema definitions, you must create a .zip archive of the WSDL file and its dependent documents, and then load the .zip file from a directory to add the REST API definition.


## Procedure

To add a REST API definition by loading a WSDL file, complete the following steps:

1. In the navigation pane, click  Develop, and then click Add API.  
The Add API: Create screen is displayed.
2. On the Add API pane, select From existing WSDL service (REST Proxy). Click Next.  
The Create API from Existing WSDL Service (REST Proxy) window opens.
3. To upload the service information from a stand-alone .wsdl file, or a .zip file that contains a WSDL file and its dependent documents, you can either drag and drop your file, or browse and select the file that you want to use.  
After you upload the file, the WSDL is evaluated and a message displays whether the WSDL is valid.  
  
If you upload a .zip file, you can include in the .zip file an options file to specify additional directives. For details, see [Using an options file when importing a WSDL service](#).
4. Click Next. At the Select Service panel, select a WSDL service from the imported file.
5. Click Next. Specify the API summary in the Info screen. You can fine-tune the API after it is created.
  - The Title can include special characters but should be kept short so that it can be easily displayed in the user interface.
  - The Name is entered automatically. The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. To view the CLI commands to manage draft APIs, see [apic draft-apis](#).
  - The Version corresponds to the value of the `info.version` property of the API's OpenAPI definition. The `version.release.modification` version numbering scheme is recommended; for example `1.0.0`.
  - The Base path is the URL segment of the API and does not include the host name or any additional segments for paths or operations. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
  - The optional Description helps to identify the API.
6. Click Next. In the Secure section, configure the API security that you require.
  - Secure using Client ID - Select this option to require an Application to provide a Client ID (API Key). This causes the `x-IBM-Client-Id` parameter to be included in the request header for the API. If selected, you can then select whether to limit the API calls on a per key (per Client ID) basis:
    - Limit API calls on a per key basis - If selected, you must configure the rate limit that you require. Rate limits control the maximum number of calls allowed within a time period (hour, minute, month or day). For example, 100 calls per hour.  
For information about security options in IBM® API Connect, see [Configuring API security](#).
    - CORS - Select this option to enable cross-origin resource sharing (CORS) support for your API. This allows your API to be accessed from another domain.
7. Optional: (API Manager UI only) Select Activate API if you want to immediately use the API for further development and testing.

Note:

- When you select the Activate API option, API Connect automatically completes the following actions:
  - Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title `api_title` auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#)

- Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).
  - You cannot use the Activate API option if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to use the Activate API option they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
  - To use the Activate API option, you must be assigned a role that has the **Product:Manage** and **Subscription:Manage** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).
8. Click Next to create your API definition.  
The Summary panel displays messages as the definition is created, and the selected security options and rate limits are enforced.
- If you selected Activate API, the wizard populates an API Endpoint URL that you can use in testing. If you have also selected Secure using Client ID, the wizard displays a Client ID and Client Secret you can use.
9. Select one of the following options:
- To further configure your API, click Edit API. For details, see [Editing an API definition](#).
  - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

## Results


---

You created a REST API definition by using an existing WSDL file.

## What to do next

---

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

You can also manage the Product lifecycle, and control who can see and subscribe to your Product, by opening the Sandbox Catalog associated with your API in the Manage page of the API Manager UI. 

## Related concepts

---

- [Developing your APIs and applications](#)

## Related reference

---

- [API and Product definition template examples](#)

## Related information

---

- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Creating a SOAP proxy API from an existing WSDL service

If you have a SOAP service defined in a WSDL file, you can use the WSDL file to create an API Connect SOAP proxy API that calls that SOAP service.

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

You can add a SOAP API to expose an existing SOAP service by supplying the WSDL file that defines that existing service in one of the following ways:


- If the WSDL file is a stand-alone file with no external dependencies, you can load the .wsdl file from a directory to create the SOAP API definition.
- You can provide one or more WSDL files in a single .zip file that contains the WSDL files and any necessary schemas.  
If the WSDL file references other WSDL files or references XSD files containing XML schema definitions, you must create a .zip archive of the WSDL file and its dependent documents, and then load the .zip file from a directory to add the SOAP API definition.

The service must support Web Services Basic Profile Version 1.1 - Second Edition.

## Procedure

---


To add a SOAP API definition by loading a WSDL file, complete the following steps:

1. In the navigation pane, click  Develop, and then click Add API.  
The Add API: Create screen is displayed.
2. Select From existing WSDL service (SOAP Proxy). Click Next.  
The Create API from Existing WSDL Service (SOAP Proxy) window opens.
3. To upload the service information from a stand-alone .wsdl file, or a .zip file that contains a WSDL file and its dependent documents, you can either drag and drop your file, or browse and select the file that you want to use.  
After you upload the file, the WSDL is evaluated and a message displays whether the WSDL is valid.

If you upload a .zip file, you can include in the .zip file an options file to specify additional directives. For details, see [Using an options file when importing a WSDL service](#).

4. Click Next. At the Select Service panel, select a WSDL service from the imported file.
5. Click Next. Specify the API summary in the Info screen. You can fine-tune the API after it is created.
  - The Title can include special characters but should be kept short so that it can be easily displayed in the user interface.
  - The Name is entered automatically. The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. To view the CLI commands to manage draft APIs, see [apic draft-apis](#).
  - The Version corresponds to the value of the `info.version` property of the API's OpenAPI definition. The `version.release.modification` version numbering scheme is recommended; for example 1.0.0.
  - The Base path is the URL segment of the API and does not include the host name or any additional segments for paths or operations. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
  - The optional Description helps to identify the API.
6. Click Next. In the Secure section, configure the API security that you require.
  - Secure using Client ID - Select this option to require an Application to provide a Client ID (API Key). This causes the `X-IBM-Client-Id` parameter to be included in the request header for the API. If selected, you can then select whether to limit the API calls on a per key (per Client ID) basis:
    - Limit API calls on a per key basis - If selected, you must configure the rate limit that you require. Rate limits control the maximum number of calls allowed within a time period (hour, minute, month or day). For example, 100 calls per hour.For information about security options in IBM® API Connect, see [Configuring API security](#).
  - CORS - Select this option to enable cross-origin resource sharing (CORS) support for your API. This allows your API to be accessed from another domain.
7. Optional: (API Manager UI only) Select Activate API if you want to immediately use the API for further development and testing.

Note:

- When you select the Activate API option, API Connect automatically completes the following actions:
    - Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title `api_title` auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#).
    - Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).
  - You cannot use the Activate API option if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to use the Activate API option they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
  - To use the Activate API option, you must be assigned a role that has the `Product:Manage` and `Subscription:Manage` permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).
8. Click Next to create your API definition.  
The Summary panel displays messages as the definition is created, and the selected security options and rate limits are enforced.  
  
If you selected Activate API, the wizard populates an API Endpoint URL that you can use in testing. If you have also selected Secure using Client ID, the wizard displays a Client ID and Client Secret you can use.
  9. Select one of the following options:
    - To further configure your API, click Edit API. For details, see [Editing an API definition](#).
    - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

## Results


---

You created a SOAP API definition by using an existing WSDL file.

## What to do next

---

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

You can also manage the Product lifecycle, and control who can see and subscribe to your Product, by opening the Sandbox Catalog associated with your API in the  Manage page of the API Manager UI.

## Related concepts

---

- [Developing your APIs and applications](#)

## Related reference

---

- [API and Product definition template examples](#)

## Related information

---



- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a new REST OpenAPI definition

You can create and edit draft REST API definitions by using the API Manager or the API Designer user interface in IBM® API Connect.




### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

**API Manager UI only:** To complete this task you must be assigned a role that has the **Api-Drafts:Edit** permission. The pre-supplied Developer role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).

### Procedure

To create a new REST OpenAPI definition, complete the following steps.

1. In the navigation pane, click  Develop, and then click Add  API.  
The Add API: Create screen is displayed.
2. Select New OpenAPI .
3. Click Next. Specify the API summary in the Info screen. You can fine-tune the API after it is created.
  - The Title can include special characters but should be kept short so that it can be easily displayed in the user interface.
  - The Name is entered automatically. The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. To view the CLI commands to manage draft APIs, see [apic draft-apis](#).
  - The Version corresponds to the value of the `info.version` property of the API's OpenAPI definition. The `version.release.modification` version numbering scheme is recommended; for example `1.0.0`.
  - The Base path is the URL segment of the API and does not include the host name or any additional segments for paths or operations. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
  - The optional Description helps to identify the API.
4. Click Next. In the Secure section, configure the API security that you require.
  - Secure using Client ID - Select this option to require an Application to provide a Client ID (API Key). This causes the `X-IBM-Client-Id` parameter to be included in the request header for the API. If selected, you can then select whether to limit the API calls on a per key (per Client ID) basis:
    - Limit API calls on a per key basis - If selected, you must configure the rate limit that you require. Rate limits control the maximum number of calls allowed within a time period (hour, minute, month or day). For example, 100 calls per hour.  
For information about security options in IBM API Connect, see [Configuring API security](#).
  - CORS - Select this option to enable cross-origin resource sharing (CORS) support for your API. This allows your API to be accessed from another domain.
5. Click Next to create your API definition.  
The Summary panel displays messages as the definition is created, and the selected security options and rate limits are enforced.
6. Select one of the following options:
  - To further configure your API, click Edit API. For details, see [Editing an API definition](#).
  - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

### Results

You successfully created a REST API definition. For API Designer, the specifications for the APIs and Products are stored in the directory that you specified when you logged in. For API Manager, the specifications for the APIs and Products are stored on the management server.

### What to do next

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

### Related tasks

- [Defining Paths for a REST API](#)
- [Configuring API security](#)

### Related reference

- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding a REST API by importing an OpenAPI definition file

You can use an OpenAPI definition file to import a REST API into IBM® API Connect.

### Before you begin

---

Your file must conform to version 2.0 of the OpenAPI specification. The format of the file can be JSON or YAML.

Note: Products that contain an API with a Swagger property using **regex** that include lookahead assertions, such as "(?" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!
The multipart 'openapi' field contains an OpenAPI definition with validation errors.
definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0, col:
0)
400
```

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.



**API Manager UI only:** To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

If you are using the API Designer UI and you want to use an OpenAPI definition file from elsewhere, downloaded from an external website for example, rather than created by using API Connect, use the import mechanism described here rather than copying the file into your local API Designer directory; the import operation adds API Connect specific sections that are required by API Designer.

### Procedure

---

To add a REST API by importing an OpenAPI file, complete the following steps:

1. In the navigation pane, click  Develop, and then click Add  API. The Add API: Create screen is displayed.
2. In the Import section, select Existing OpenAPI, and click Next.
3. To import a file from your local file system, you can either drag and drop your files, or click Browse and select the file that you want to use. The following file types are supported if they contain a valid OpenAPI definition: .json, .yaml, and .yaml. The wizard checks the validity of the YAML, and displays a message indicating successful validation.

Note:

- If you are using the API Manager UI, the import operation fails if the file defines an API that has the same name and version as an existing API definition. However, if you are using the API Designer UI, API definitions are uniquely identified by the file name in your local file system; therefore, if you import two different files that define the same API name and version, two API definitions with the same name and version are created in API Connect, with no error.
- Any validation error messages are displayed in English only, and are not translated.

4. Click Next.

5. Optional: (API Manager UI only) Select Activate API if you want to immediately use the API for further development and testing.


Note:

- When you select the Activate API option, API Connect automatically completes the following actions:
  - Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title *api\_title* auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#)
  - Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).
- You cannot use the Activate API option if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to use the Activate API option they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
- To use the Activate API option, you must be assigned a role that has the **Product:Manage** and **Subscription:Manage** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

6. Click Next. The Import API Summary panel indicates that the YAML file is loaded and valid.

If you selected Activate API, the wizard populates an API Endpoint URL, and displays a Client ID and Client Secret that you can use.

7. Select one of the following options:

- To further configure your API, click Edit API. For details, see [Editing an API definition](#).
- If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

### Results

---

When the API definition has been imported, it is shown in the list of API definitions in the APIs and Products page. For API Designer, the specifications for the APIs and Products are stored in the directory that you specified when you logged in. For API Manager, the specifications for the APIs and Products are stored on the management

server.

## What to do next

---

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

## Related information

---

- [IBM API Connect overview](#)
- [Managing your APIs](#)
- [OpenAPI Specification](#)
- [What is OpenAPI?](#)
- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding a SOAP API by importing a zip file

You can use an OpenAPI definition file to import a SOAP API into IBM® API Connect.

### Before you begin

---

The format of the file must be .zip, and the file must contain the WSDL definition and the YAML file that defines the API. The YAML file must conform to version 2.0 of the OpenAPI specification.

Note: Products that contain an API with a Swagger property using **regex** that include lookahead assertions, such as "(?" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!  
The multipart 'openapi' field contains an OpenAPI definition with validation errors.  
definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0, col:  
0)  
400
```

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

**API Manager UI only:** To complete this task, you must be assigned a role that has the **Api-Drafts:Edit**, **Settings:View**, and **App:View** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).


If you are using the API Designer UI and you want to use an OpenAPI definition file from elsewhere, downloaded from an external website for example, rather than created by using API Connect, use the import mechanism that is described here rather than copying the file into your local API Designer directory; the import operation adds API Connect specific sections that are required by API Designer.


For more information on how to create a .zip file, see [Downloading an API definition](#).

## Procedure

---

To add a SOAP API by importing a .zip file, complete the following steps:

1. In the navigation pane, click  Develop, and then click Add > API.  
The Add API: Create screen is displayed.
2. In the Import section, select Existing OpenAPI, then click Next.
3. To import a file from your local file system, you can either drag and drop your files, or click Browse and select the file that you want to use.  
The wizard checks the validity of the WSDL and YAML, and displays a message that indicates a successful validation.  
Note:
  - If you are using the API Manager UI, the import operation fails if the file defines an API that has the same name and version as an existing API definition. However, if you are using the API Designer UI, API definitions are uniquely identified by the file name in your local file system; therefore, if you import two different files that define the same API name and version, two API definitions with the same name and version are created in API Connect, with no error.
  - Any validation error messages are displayed in English only, and are not translated.
4. Click Next.
5. Optional: (API Manager UI only) Select Activate API if you want to immediately use the API for further development and testing.  
Note:
  - When you select the Activate API option, API Connect automatically completes the following actions:
    - Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title *api\_title* auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#)

- Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).
  - You cannot use the Activate API option if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to use the Activate API option they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
  - To use the Activate API option, you must be assigned a role that has the **Product:Manage** and **Subscription:Manage** permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).
6. Click Next. The Import API Summary panel indicates that the file is loaded and valid.  
If you selected Activate API, the wizard populates an API Endpoint URL, and displays a Client ID and Client Secret that you can use.
7. Select one of the following options:
- To further configure your API, click Edit API. For details, see [Editing an API definition](#).
  - If you do not want to configure your API further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task. For details on how to configure your API later, see [Editing an API definition](#).

## Results

---

When the API definition has been imported, it is shown in the list of API definitions in the Develop page. For API Designer, the specifications for the APIs and Products are stored in the directory that you specified when you logged in. For API Manager, the specifications for the APIs and Products are stored on the management server.

## What to do next

---

APIs are made available to application developers by including them in a Product, and then publishing that Product to a Catalog. For more information, see [Working with Products](#) and [Working with Catalogs](#).

## Related information

---

- [IBM API Connect overview](#)
- [Managing your APIs](#)
- [OpenAPI Specification](#)
- [What is OpenAPI?](#)
- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---


## Editing an API definition

---

Use this topic to specify configuration settings for an API. You can do this when you create the initial definition, or you can do it later by editing an existing definition.

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI. During the initial creation of an API, the API wizard guides you to enter the minimum configuration settings, and then provides an Edit API icon. You can select it continue with advanced configuration, or click  to exit the configuration. When you click Edit API, you can use the API Setup page to specify additional configuration. This topic describes how to specify additional configuration.

When you select Edit API, the Design view of the Develop API editor is displayed. You can configure your definition by editing the different sections in the page navigation side bar of the Develop API editor. The following steps take you through these sections. You can also write your API directly in the source by using the Source view, and you can configure your policy assembly by using the Assemble view

**Note:** If you make an update in any section of the Develop API editor, you must click Save in that section to ensure that your changes are updated in the API definition.

## Procedure

---

1. Access the API Setup page in one of the following ways:
  - During the initial creation of an API, the API wizard guides you to enter the minimum configuration settings, and then provides an Edit API icon. When you select that icon, the API Setup page is displayed. Proceed to the next step.
  - When you want to edit an existing API, you can access the API settings by going to the navigation pane and clicking Develop. Select the API that you want to edit. The API Setup page is displayed.
2. In the API Setup tab, you can edit the following sections:
  - a. Optional: In the Info section, edit any or all of Title, Description, and Version.  
**Note:** If you change the Title or Version, when you save the API definition a new API definition is created, copying the current configuration.
  - b. Optional: In the Consumes section, select which types of media your API will accept when calls are made to it. In addition to JSON and XML, which are supported by API Connect, you can add other media types by using the Add Media Type field.  
**Note:** The configurations that you make become defaults for all of the operations in the API. However, you can override these media types for individual operations. For more information, see [Defining Paths for a REST API](#).
  - c. Optional: In the Produces section, select which types of media your API will return when calls are made to it. In addition to JSON and XML, which are supported by API Connect, you can add other media types by using the Add Media Type field.

Note: The configurations that you make become defaults for all of the operations in the API. However, you can override these media types for individual operations. For more information, see [Defining Paths for a REST API](#).

- d. Optional: If you want external developers on the portal to access your APIs by using a host address that is not the same as the API endpoints that are defined for the gateway services where your API will run, use the Address field in the Host section to define the host name that is to be used.
- Note: To enable this capability, you must additionally configure API vanity endpoints in the Catalog Settings for the Catalog to which the API is published. For information on configuring vanity endpoints, see [Creating and configuring Catalogs](#).
- e. Optional: In the Base Path section, you can change the path segment that is shared by all operations in your API. The base path is the initial URL segment of the API, and does not include the host name or any additional segments for paths or operations.
- f. In the Schemes section, select which transfer protocols you want your API to use. HTTPS is selected by default.
- Note: If your API is enforced by an API Connect gateway, only the HTTPS protocol is supported. See the following Lifecycle section for instructions about how to enable enforcement.
- g. In the Lifecycle section you can use the drop-down menu to change the phase of the lifecycle that your API is in.
- Realized (default) - The API is in the implementation phase.
  - Identified - The API is in the early conceptual phase and is neither fully designed nor implemented.
  - Specified - The API has been fully designed and passed an internal milestone but has not yet been implemented.

You can also edit the following options:

- Enforced - Select this option to enforce the API by using the API Connect Gateway. Clear this option if you are managing the API on a gateway other than an API Connect gateway. Although not published to a gateway by API Connect, an unenforced API is still available in the Developer Portal for subscription by application developers.
- Testable - Select this option to allow the API's operations to be tested using the test tool in the Developer Portal.  
Note: For the test tool to work, an API must be included in a Plan in a Product that is staged in a Catalog.
- CORS - Select this option to enable cross-origin resource sharing (CORS) access control for your API.
- Application authentication - Select this option to protect your API with a certificate, for example, by using TLS mutual authentication. When the API is called, an X509 client certificate must be supplied, either in the **X-Client-Certificate** HTTP header, or as a TLS client certificate from TLS mutual authentication. For any Developer Portal application that calls the API, the certificate must be entered in the Developer Portal user interface; for details, see [Registering an application](#).

The Gateway service to which the API is published can be configured to use TLS mutual authentication to secure API calls made to that Gateway service; for details, see [Configuring the initial Gateway service](#).

If you are using a load balancer, you must configure the load balancer to use the **X-Client-Certificate** HTTP header to relay the appropriate client certificate to the Gateway service after the load balancer terminates the TLS communication.

- **API Gateway only** Buffering - if you select this option, the API requests and responses are first buffered before being processed on the DataPower® API Gateway. If you clear this option, the gateway streams the API requests and responses byte by byte where possible. However, situations in which the gateway will buffer the requests and responses even if the Buffering option is cleared include the following:
  - The API policy assembly includes a **parse** policy; the **parse** policy buffers the message before reading it into the parse tree.
  - The API policy assembly includes policies that use GatewayScript functions that require the message to be buffered; for example, `context.message.body.readAsBuffer`, `context.message.body.readAsJSON`, `context.message.body.readAsXML`.
  - The API policy assembly includes policies that use the `apim.getvariable` GatewayScript function, which is backwardly compatible with the DataPower Gateway (v5 compatible); the gateway falls back to using buffering mode.
  - The **Content-Type** header of the incoming request is `application/x-www-form-urlencoded`; the gateway reads the message to populate the form parameters.
  - A WSDL API routes the request based on the name of the root element inside the SOAP body.
  - The API policy assembly includes a **map** policy; the gateway buffers the message before executing this policy.

h. Optional: In the Parameters section, add parameters that you want to be defined on all the Paths in the API; these parameters are inherited by all the Path definitions in the API. Click Add and complete the following fields:

  - Name - Provide a name for your parameter.
  - Located In - Select where the parameter is found in the call of your operation.
  - Type - Select the type of data that the parameter is expected to have when the call is received by API Connect. If you have set Located In to Body, then you can select a JSON schema that you have defined for your API in the Definitions section. For more information about creating JSON schemas, see [step 6](#).
  - Description - Provide a description of your parameter.
  - Required - Select this check box to specify that the parameter is required for a call to be valid.

i. Optional: In the Tags and External Documentation section, click Add to provide any tags and external documentation details that you want to refer users to. Tags added in this way appear in the OpenAPI definition of the API, but are not used by API Connect for any indexing.

j. Click Save to save your updates.

3. In the Security Definitions tab, you can specify the security definitions that might be used by the API or its operations. A security definition specifies all the settings for a particular aspect of API security; for example, the user registry that you use to authenticate access to the API. For more information, see [Configuring API security](#).

4. In the Security tab, select the security definitions that you want to apply to your API. Click Add to make your definitions appear in the list of security definitions. Click Save to save your updates.

To be available in the Security section, definitions must have been defined in the Security Definitions section.

5. In the Paths tab, you can add API Paths to your definition.

A Path is a unit of a REST API that can be called by applications. A Path comprises of one or more HTTP verb operations and a URL path that, when exposed, is combined with the base path of the API. By configuring the Path, you define how the API is exposed to your developers. For more information about Paths, operations, and adding tags to your operations, see [Defining Paths for a REST API](#) and [Configuring an operation](#).

6. In the Definitions tab, you can create schema definitions.

These definitions can then be referenced in an operation to provide developers with information about the request they should make, or the response they should expect to receive from the operation. Your schema definitions are made available to developers through the Developer Portal, but are not enforced in any calls to the API unless a [Validate - DataPower Gateway \(v5 compatible\)](#) policy is used.

a. Click Add.

The Edit Definitions page is displayed.

b. Complete the Name field for your definition.

c. From the Type drop-down list, select the type of your definition.

d. Optional: In the Description field for your definition, provide a description of what is defined by the definition.

e. If you want to allow the inclusion of properties that are not included in the definition, so that validation will not fail when a validate-rest policy is used on the request, select the Additional properties check box.

f. Click Save to complete the details of your definition's properties. Each property requires a name and type, and can also have an example and description.

g. Optional: You can add additional properties by clicking Add again.

Note:

- You can include more complex schema definitions in your API by using the Source view, and editing your API OpenAPI definition directly. For more information, see the [OpenAPI specification](#).
- If, for a schema definition property of type `long`, you specify a `minimum` or `maximum` parameter in the OpenAPI source, the highest value you can set is `9007199254740992`; if you exceed this value then, due to rounding behavior, an incorrect value might be saved.

Tip: You can click Edit schema in a definition to display the Add XML schema window. In this window, you can edit, upload, or infer an XML schema from a JSON schema.

7. In the Properties tab, you can define any API properties that you want to use. For more information, see [API properties](#).
8. In the Target Services tab, add any web services that you want to use in your API definition. Complete the following steps:
  - a. Click Add.
  - b. To upload the service information from a stand-alone .wsdl file, or a .zip file that contains a WSDL file and its dependent documents, you can either drag and drop your file, or browse and select the file that you want to use. After you upload the file, the WSDL is evaluated and a message displays whether the WSDL is valid.  
If you upload a .zip file, you can include in the .zip file an options file to specify additional directives. For details, see [Using an options file when importing a WSDL service](#).
  - c. In the Select Target Service pane, select the service that you want to add, then click Submit.  
The target services that you add are now available in the palette on the Assemble view for you to add to your API assembly flow; for more information, see [Including components in your assembly](#).
9. In the Categories tab, define any categories that you want to organize your APIs into. Click Save to save any updates.  
By organizing your APIs into categories, you can provide a hierarchical display for your APIs in the Developer Portal. For more information, see [Organizing your APIs and Products into categories](#).
10. If the gateway type associated with the API is the DataPower API Gateway, select the Activity Log tab to configure your logging preferences for the API activity that is stored in API Connect Analytics. By default, the content type that is collected and stored in API event records is activity for when API execution completes successfully, and payload for when API execution completes with an error code. For more information, see [Activity logging with the DataPower API Gateway](#). Click Save to save any updates.  
Note that if the gateway type associated with the API is the DataPower Gateway (v5 compatible), you can configure your logging preferences by applying an Activity Log policy in the Assemble view. For more information, see [Activity Log](#).  
  
The gateway type that is associated with the API by default when you first create it depends on the gateway services that are defined in the Sandbox Catalog. For more information, see [API Connect gateway types](#) and [Specifying the gateway type for an API definition](#).
11. Select the Assemble view to configure and test the policy assembly for your API. Click Save to save any updates.  
You can drag and drop policies and logic constructs onto the assembly canvas, and use the internal test tool to test your assembly. For more information, see [The Assemble view](#), and [API policies and logic constructs](#).

- **[Defining Paths for a REST API](#)**

A Path is a unit of a REST API that you can call. A Path comprises an HTTP verb and a URL path that, when exposed, is combined with the base path of the API. By configuring the Path, you define how the API is exposed to your developers.

- **[Organizing your APIs and Products into categories](#)**

You can organize your APIs and Products into categories. The APIs and Products that you categorize are displayed within the Developer Portal, in their defined categories.

- **[Activity logging with the DataPower API Gateway](#)**

If you're using the DataPower API Gateway, you can use the Activity Log tab in the API Manager UI to configure your logging preferences for the API activity that is stored in Analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.

- **[The Assemble view](#)**

The API Manager of IBM API Connect features an Assemble view that you can use to create assemblies. With assemblies, you can readily tailor your APIs to include components such as activity logging and redaction of specific fields.

- **[Including components in your assembly](#)**

An assembly is formed of components that are applied to calls to and responses from operations in your API. Components can be either policies or logic constructs.

- **[Creating a new version of an API definition](#)**

You can create multiple versions of an API definition and edit the versions independently by using IBM API Connect.

- **[Specifying the gateway type for an API definition](#)**

An API definition is specific to one or other of the gateway types, DataPower API Gateway or DataPower Gateway (v5 compatible). A default gateway type is set when you create an API definition, but you can edit the API configuration to specify a different gateway type.

- **[Converting an API definition for deployment to the DataPower API Gateway](#)**

IBM API Connect provides two gateway types, DataPower Gateway (v5 compatible) and DataPower API Gateway. If you have an API definition that was developed for the DataPower Gateway (v5 compatible) and you want to port it to the DataPower API Gateway, you must make changes to convert it before deployment.

- **[API properties](#)**

Properties are used by the gateway to control behavior of certain policies. Typically, you provide properties, but the policy can also provide properties settings.

- **[Variable references in API Connect](#)**

In API Connect you can reference different variables in your API definition.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Defining Paths for a REST API

A Path is a unit of a REST API that you can call. A Path comprises an HTTP verb and a URL path that, when exposed, is combined with the base path of the API. By configuring the Path, you define how the API is exposed to your developers.

---


### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

## Procedure

---

To define a Path, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. Click the REST API definition that you want to work with.
3. Click Paths, then click Add.
4. In the Path name field, enter the name of the Path, using the format `/path_name`.

The path is appended to the base path to construct the full URI to access the APIs. The path must start, but not end, with the / character. A parameter at the end of the path can contain a qualifier to match one or more path levels.

If you specify just the name of the parameter, then one level of that path is matched. If you want to allow for multiple levels of the path, you can prefix the parameter with one of the following qualifiers:

- \* to indicate 0 or more occurrences
- + to indicate 1 or more occurrences

The + and \* qualifiers can only be used at the end of the path.

For example, the path:

```
/petstore/{type}/{*category}
```

matches the following paths, where only one type value is matched, but all (0 or more) categories are matched:

```
/petstore/cats  
/petstore/cats/supplies  
/petstore/cats/supplies/health  
/petstore/cats/supplies/health/medicines  
/petstore/cats/supplies/health/medicines/a/b/c
```

Note: The full API does not have to be unique across all Paths in your IBM® API Connect system. However, if it is not unique then you *must* specify that an application is required to identify itself with a client ID when it calls the operation; the client ID is used to uniquely determine which operation to call, according to which Plan the application is subscribed to.

For information on specifying application identification requirements, see [Creating an API key security definition](#).

5. To add one or more HTTP operations to your Path, complete the following steps:

- a. In the Operations section, click Add.
- b. Select the operations that you want to add.  
You can select from the following operations:

GET	Retrieves data from the server.
PUT	Updates data that is stored in the server.
POST	Sends data to the server for processing.
DELETE	Removes data from the server.
PATCH	Applies partial modifications to a path. Unlike the PUT method, the PATCH method applies an incremental change rather than replacing the entire operation. Note: If you want to use the PATCH operation, your API Connect gateway must be running DataPower® firmware Version 7.0.0 or later, otherwise the request is rejected by the gateway.
HEAD	Requests the same response as for a GET method call, but without the response body. This method is useful for retrieving information that is written in response headers, without having to transport the entire content.
OPTIONS	Retrieves the HTTP methods and other options that are supported by a web server or an operation, without implying a resource action or initiating an operation retrieval.

Note: For each Path, you can have only one of each type of operation.

- c. Click Add to confirm the addition of the selected Paths.

6. Optional: Add any parameters that you want to include for the Path and all of its operations, by completing the following steps:

- a. In the Path Parameters section, click Add.
- b. In the NAME field, provide a name for your parameter.  
Note: The query parameters `appId`, `client_id`, `client_secret`, and `appSecret` are reserved and cannot be used.

- c. In the LOCATED IN field, select where the parameter is located when your path operations are called.

- d. Optional: From the TYPE drop-down list, select the type of data that the parameter is expected to have when the call is received by API Connect. If you have set LOCATED IN to body then you can select a JSON schema that you have defined for your API. For more information about creating JSON schemas, see [Step 6 of Editing an API definition](#).

- e. Optional: In the DESCRIPTION field, provide a description of your parameter.

- f. Use the REQUIRED check box to specify whether the parameter is required for a call to be valid.

Note: If you specify that a parameter is required, the `required` field for the parameter in the OpenAPI definition file is set to `true`, so that the requirement is documented for consumers of the API. The requirement is not enforced by the IBM API Connect Gateway

However, the requirement is enforced by the [Developer Portal test tool](#), because the purpose of these tools is to explore the correct operation of the API.

7. Click Save to save your changes.

8. Configure your operations. For more information, see [Configuring an operation](#). You must provide at least one property or response for each operation, depending upon the operation type.

---

## Results



The Paths and operations for your REST API are defined.

## What to do next

---

Add your API to a Plan in a Product. When your API is part of a Product you can stage your Product to a Catalog to test the operations in your API. When you stage a Product, you can publish it to the Developer Portal for application developers to use your APIs and operations.

For more information about Products, see [Working with Products](#).

- [Configuring an operation](#)  
One or more HTTP operations together form a Path. These operations are different ways of interacting with an API and you can use GET, POST, PUT, DELETE, HEAD, PATCH, and OPTIONS operations.

## Related tasks

---

- [Including components in your assembly](#).

## Related information

---

- [IBM API Connect overview](#)
- [API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring an operation

One or more HTTP operations together form a Path. These operations are different ways of interacting with an API and you can use GET, POST, PUT, DELETE, HEAD, PATCH, and OPTIONS operations.

## About this task


---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.  
An operation must have at least one success response defined.

## Procedure

---

To configure an operation, complete the following instructions:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. Click the REST API definition that you want to work with.
3. Click Paths, click the required Path, then click the required operation.  
The details page for the operation opens.
4. Optional: In the Operation Id field, provide an identifier for your operation. The operation ID must not be shared by any other operations.
5. Optional: In the Summary field, provide a summary of your operation.
6. Optional: In the Tags field, add any tags that you want to associate your operation with.  
The tags are included in the OpenAPI definition file for the API, and can be used to group operations in the Developer Portal by using the filter options for the API.
7. Optional: In the Description field, provide a description of your operation.
8. Specify which security definitions to apply to your operation. By default, all the security definitions that have been configured for the API are applied to the operation. To select which of the API security definitions you want to apply to the operation, complete the following steps:
  - a. Select Override API Security Definitions.  
An Add button is displayed.
  - b. Click Add, then select the required security definitions.
9. Specify the media types that the operation produces. By default, the operation produces the same media types as those that have been defined for the API. To specify the required media types, select Override API Produce Types; you can select JSON and XML, which are supported by API Connect, and you can add another media type in the Add media Type field.
10. Specify the media types that the operation consumes. By default, the operation consumes the same media types as those that have been defined for the API. To specify the required media types, select Override API Consume Types; you can select JSON and XML, which are supported by API Connect, and you can add another media type in the Add media Type field.
11. Optional: Add any parameters that are specific to this operation, by completing the following steps:
  - a. In the Parameters section, click Add.
  - b. In the NAME field, provide a name for your parameter.  
**Note:** The query parameters `appId`, `client_id`, `client_secret`, and `appSecret` are reserved and cannot be used.
  - c. In the LOCATED IN field, select where the parameter is located when your operation is called.
  - d. Optional: From the TYPE drop-down list, select the type of data that the parameter is expected to have when the call is received by API Connect. If you have set LOCATED IN to body then you can select a JSON schema that you have defined for your API. For more information about creating JSON schemas, see [Step 6 of Editing an API definition](#).
  - e. Optional: In the DESCRIPTION field, provide a description of your parameter.
  - f. Use the REQUIRED check box to specify whether the parameter is required for a call to be valid.  
**Note:** If you specify that a parameter is required, the `required` field for the parameter in the OpenAPI definition file is set to `true`, so that the requirement is documented for consumers of the API. The requirement is not enforced by the IBM® API Connect Gateway  
However, the requirement is enforced by the [Developer Portal test tool](#), because the purpose of these tools is to explore the correct operation of the API.



12. Add or edit any responses that you want to include.

These responses are only for the OpenAPI definition of your API that is provided to developers and are not used for any purposes other than documentation.

a. In the Response section, click Add.

b. In the STATUS field, provide the HTTP status code that might be returned.

c. Optional: If you have defined any JSON schemas for your API, you can select to reference one of them from the SCHEMA drop-down list. For more information on creating JSON schemas, see Step 6 of [Editing an API definition](#).

d. In the DESCRIPTION field, provide text to be returned for the specified status code.

13. Click Save to save your changes.

---

## Results

You have configured your operation and can add more operations to your Path.

---

## Related tasks

- [Including components in your assembly](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Organizing your APIs and Products into categories

You can organize your APIs and Products into categories. The APIs and Products that you categorize are displayed within the Developer Portal, in their defined categories.

---

## Before you begin

Your role must have the necessary permissions to stage and publish Products.

---

## About this task


You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

By organizing your APIs and Products into categories, you can provide a hierarchical display for your APIs and Products in the Developer Portal.

Note: Organizing APIs and Products into a hierarchical view in the API Manager or API Manager UI is different to tagging in the Developer Portal.

---

## Procedure

1. In the navigation pane, click  Develop.
2. Click the title of the API or Product that you want to work with.
3. Click Categories.
4. In the Categories text box, enter the hierarchical taxonomy path that you want your API or Product to follow, in the following format:

```
top_level_element / next_level_element / x_level_element
```

For example:

```
Animals / Fluffy / Cat
```

5. After you have completed entering the category specifications for the API or Product, click Save.
6. In the API Manager UI, publish the Product that you have staged by following the steps in [Publishing a new Product](#).

---

## What to do next

- Publish the Product; see [Publishing a new Product](#).
- After publishing the Product, you can, in the Developer Portal, display the APIs and Products in the categories that you have defined. For more information, see [Displaying APIs and Products in categories](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Activity logging with the DataPower API Gateway

If you're using the DataPower® API Gateway, you can use the Activity Log tab in the API Manager UI to configure your logging preferences for the API activity that is stored in Analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

## About

An API event record exists for each API execution event in the Gateway server. By default, the content type that is collected and stored in API event records is **activity** for when API execution completes successfully, and **payload** for when API execution completes with an error code. When you compose your API definition, you can change the type of content to log in these API event records. During API execution, the activity data is stored in the **log** context variable, which populates the API event record on completion of the API execution; for more information, see [API activity logging context variables](#).

Activity logging can be configured in the API Manager by using the Activity Log tab, in the Design view of the Develop API editor. Or you can edit the source directly (see the [Example source](#) section for details).

Note that if you're using the DataPower Gateway (v5 compatible), you can configure your logging preferences by applying an Activity Log policy in the Assemble view. For more information, see [Activity Log](#).

Note:

Activity logging that calls for logging of Analytics data upon success doesn't apply for the OAuth provider. The OAuth provider logs Analytics data for failure cases, but doesn't log successful cases.

## Properties

The following table lists the activity log properties in the API Manager UI, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 1. Activity log properties

Property label	Required	Description	Data type
Activity Log	Yes	Indicates whether activity logging is enabled or disabled. Use the slider control to turn activity logging On or Off. The default value is On.	string
Content	Yes	Defines the type of content to be logged when the operation is successful. Valid values: <ul style="list-style-type: none"><li><b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li><li><b>activity</b>: Logs invocation only (only the resource URI is recorded).</li><li><b>header</b>: Logs activity and header.</li><li><b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li></ul> The default value is <b>activity</b> .	string
Content on error	Yes	Indicates what content to log when an error occurs. Valid values: <ul style="list-style-type: none"><li><b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li><li><b>activity</b>: Logs invocation only (only the resource URI is recorded).</li><li><b>header</b>: Logs activity and header.</li><li><b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li></ul> The default value is <b>payload</b> .	string

## Example source

The following examples show how the OpenAPI for the activity log configuration looks in the source code.

Example one shows the default activity log preferences:

```
activity-log:  
  success-content: activity  
  error-content: payload  
  enabled: true
```

Example two shows activity logging that has been turned off:

```
activity-log:  
  enabled: false
```

## Related tasks

- [Creating a new REST OpenAPI definition](#)
- [Editing an API definition](#)

## Related information

- [API Analytics](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---


## The Assemble view

The API Manager of IBM® API Connect features an Assemble view that you can use to create assemblies. With assemblies, you can readily tailor your APIs to include components such as activity logging and redaction of specific fields.

You can access the Assemble view by clicking the Assemble tab at the top of the Develop API editor. This view includes a palette, which lists the available components, a property sheet, which is used to configure a component, and a canvas, which is used to arrange and visualize the assembly's components.

---

## The palette

The palette, shown on the left side of the Assemble view, is a list of different components that you can include in your assembly. The palette can be hidden by clicking the Show/hide policy palette icon  in the upper left corner.


---



## The canvas

You can use the canvas to create a graphical representation of the assembly flow. You can drag various components from the palette on the left to the appropriate location on the canvas on the right. When you drag a component, valid positions are shown by dashed boxes.

Note: You must click Save in the Assemble view to save your assembly updates, and to make them appear in the Source view. API calls are made at the left, unfilled, circle, and returned at the right, filled, circle. You can insert components between the two circles to modify the data received from the call, or returned by the response. To add a component, select it in the palette and drag it across to one of the dashed boxes that appear when you move the component over the canvas.

You can use the Show catches toggle, in the upper right corner of the page, to show and hide error catches in the palette. A catch is a section of the assembly that is applied when an API call results in the corresponding HTTP status code being returned. Click the catch section to open the property sheet for all your catches.


You can zoom the view of your canvas in and out by clicking the + and - icons. To fit the canvas to the screen size, click the Zoom to fit icon .

You can filter the canvas to show only the parts of it that will apply to a specific operation by clicking the Filter by operation icon  and then selecting the operation from the drop-down list. Click the Clear operation filter icon  to remove the filter.

---

## The property sheet

When you select a component that is in the assembly by clicking it, details about the component are displayed in the property sheet on the right. In this pane, you can configure the component's properties. The options available to you in the property sheet are specific to the type of component you are working with. For some components, you can add and remove properties by clicking Object Properties and selecting the property from the drop-down menu.

You can release the property sheet by clicking the Close icon .

---

## Related tasks

- [Defining Paths for a REST API](#)
- [Adding components to your assembly](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Including components in your assembly

An assembly is formed of components that are applied to calls to and responses from operations in your API. Components can be either policies or logic constructs.

---

## About this task

In the Assemble view of IBM® API Connect, you can add and configure components in your assembly. You can also directly add components to the OpenAPI definition of your API.

- [Adding components to your assembly](#)  
Create your API assembly by using the Assemble view in the API Manager.
- [Handling errors in the assembly](#)  
Use the catch section of the assembly to describe the handling of errors thrown during the assembly execution.
- [OpenAPI and assembly components](#)  
An assembly in IBM API Connect is formed of one or more components that are applied to calls to an API. These components can be part of the OpenAPI specification or extensions to the specification that are specific to API Connect.
- [The behavior of an assembly](#)  
The assembly runs policies in order and acts on different contexts of the API call.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding components to your assembly

Create your API assembly by using the Assemble view in the API Manager.

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

You can use the assembly tool in the UI to create assemblies that are used to manipulate requests made to or responses made by any of your API's operations.


Alternatively, you can use the Source view to edit the source of your API definition, in which case the syntax is as described in [execute OpenAPI extension](#).

For more information about the use of the assembly tool, see [The Assemble view](#).

### Procedure

---

To add components to your assembly using the assembly tool, complete the following steps:

1. In the navigation pane, click  Develop.  
The Develop: APIs and Products tab opens.
2. Click the API definition that you want to work with.
3. Use the Assemble view to add a component.
  - a. Click the Assemble tab at the top of the Develop API editor.
  - b. Find the component that you want to add in the palette on the left. For a list of components and the categories to which they belong, see [OpenAPI and assembly components](#).
  - c. Drag the component onto the canvas; dashed boxes are displayed. Drop the component in a dashed box to insert it into that position in the assembly.  
Note:
    - Components are applied in order from the left, unfilled, circle to the right, filled, circle.
    - Unless an Operation Switch component is used, the whole assembly applies to every operation in the API.
  - d. Add and edit properties of the component by clicking the component and using the property sheet that is shown on the right.  
For some components, you can add and remove properties by clicking Object Properties and selecting the property from the drop-down list. For information about the properties of policy components, see [API policies and logic constructs](#). For information about the properties of logic constructs, see [Logic Constructs](#).
4. Optional: Repeat Steps [3.b](#) to [3.d](#) for any additional components that you want to add.
5. Click Save to save your changes.

### Results

---

You have added one or more components to your assembly.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Handling errors in the assembly

Use the catch section of the assembly to describe the handling of errors thrown during the assembly execution.

### About this task

---


You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.





The catch section of the assembly is used to implement an assembly in the instance that an error is thrown during the assembly execution. For example, the assembly could contain a throw component, the API caller could fail to authenticate, or a policy could fail to execute correctly. Each error can be handled with a different catch and each catch can handle multiple status errors.

### Procedure

---

To create a catch and include components in it, complete the following steps:

1. In the navigation pane, click  Develop.  
The Develop: APIs and Products tab opens.
2. Click the API definition that you want to apply a catch to, and then click the Assemble tab at the top of the Develop API editor.
3. Set the Show catches toggle in the menu bar above the canvas to the Show position.

4. Click Catch at the bottom of the canvas, or a Catch icon  if one is displayed.  
The property sheet for the API's catches opens.
5. To add a default catch that is executed when an otherwise uncaught error is thrown, click + Default.  
Note: If you have a default catch above another catch in precedence, the default catch will activate even when the other catch's error is thrown.
6. To add a new catch, click + Catch.
7. To specify which errors the catch applies to, type the name of a custom error and press Enter, or use the search errors field to search for the appropriate error.
8. Optional: To remove an error case from a catch, click the corresponding cross.
9. Optional: To change the precedence of your catches, use the Move up  or Move down  icons.  
If an error case is handled by multiple catches, the catch at the top of the list is applied.
10. To add a component to a catch, drag the component over the dashed, gray box that appears in the flow from the Catch icon  for the catch that you want to apply the component to.
11. Click Save to save your changes.

## What to do next

If you added a catch for `ConnectionError`, `SOAPError`, or `OperationError`, you must add the same error to the Stop on error setting for the Invoke policy in your assembly, otherwise if the error occurs during the execution of the Invoke policy, it is not caught, the policy execution is allowed to complete, and the assembly flow continues. For details on configuring an Invoke policy, see [Invoke](#).

- [Error cases supported by assembly catches](#)  
Several error cases that can be returned by the assembly are available to the catch function.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Error cases supported by assembly catches

Several error cases that can be returned by the assembly are available to the catch function.

### ConnectionError

An error occurred while establishing a connection to another URL.

### JavaScriptError

An error occurred while executing JavaScript or GatewayScript in a policy.

### PropertyError

An error occurred due to an incorrect property during an invoke call or during the execution of a set-variable policy when an action was not set, add, or clear.

### RedactionError

An error occurred during the redaction of a field as part of a redact policy.

### TransformError

An error occurred during a transformation policy.

### RuntimeError

An otherwise unspecified error occurred.

### BadRequestError

An error occurred while trying to access the request.

### UnauthorizedError

The API cannot be invoked based on the client ID and/or client secret provided, or id/secret were specified in the wrong location.

### ForbiddenError

The application making the API request has been disabled or is not active.

### ValidateError

An error occurred while trying to schema validate a message payload.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OpenAPI and assembly components

An assembly in IBM® API Connect is formed of one or more components that are applied to calls to an API. These components can be part of the OpenAPI specification or extensions to the specification that are specific to API Connect.

The full OpenAPI specification can be found at the [OpenAPI website](#).

For draft APIs with an OpenAPI definition file, the API Manager UI performs validation on it. Any warnings or errors that are highlighted will not prevent further editing of the API, or prevent it from being saved. Further validation occurs when the API is staged, as part of a Product, to a Catalog. For more information, see [Staging a Product](#).

## API Connect policies and logic constructs

To provide additional functions, API Connect uses various extensions to the OpenAPI specification. For more information, see [API policies and logic constructs](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## The behavior of an assembly

The assembly runs policies in order and acts on different contexts of the API call.

When an API call is made, security and rate limits are enforced before the assembly is executed. During the assembly, the flow can branch or be thrown and caught, according to the policies contained in it. The message context can be thought to flow through the assembly, being used and altered by various policies. In addition to the message, other contexts can be accessed and created.

## Security and rate limiting

Before the assembly is executed, security and then rate limits are enforced.

First, security definitions and CORS access control are used to authenticate an API call. Any API Key security definitions are used to identify applications that have subscriptions to a Product containing the API. If a security definition does not allow access, the API call is rejected.

If an application is identified by its client ID or client secret, a rate limit can be enforced based on the Plan or operation called.

## The assembly

The assembly is executed in order from the initial, filled, circle to the final, unfilled, circle. However, there is room for branching, when if and operation-switch logic constructs are used, or for the remaining assembly to be ignored when a throw policy is executed.

The message is the context that is acted upon by any policy that isn't otherwise configured. At the beginning of the API call, the message is empty and, at the end of the API call, the message is used as the response.

The request context contains the information that is sent by the API caller and varies with the type of operation called and the configuration of that operation. For example, a GET operation can never have a populated `request.body`, and you can configure an operation to have `request.parameters` (query parameters). The first policy in an assembly acts on the request and produces the first instance of the message. If there are no policies, the request is returned to the caller.

## Managing contexts

Because the message can be overwritten, it can be useful to create and reference new contexts where possible so that they are saved and reusable during the API call.

- Use the map policy to overwrite the message context when you need to execute a policy that only acts on the message.
- Use the request context when you want to use the original request made to the API.
- Use the map, invoke, and proxy policies to create new contexts when you want to save your message.

Note: When you create a new context, unless you are also mapping to the message, the message is overwritten with an empty object.

For example, an invoke policy is the first policy in the assembly and its response overwrites the request as the message. The message is then acted upon by a validate policy, and a map policy then saves the message as a new context, ready for a second invoke policy to overwrite the message without losing the first invoke policy's output.

You can also access contexts outside of the message or your custom contexts, but these cannot be written to. For a list of contexts, see [API Connect context variables](#).

## Branches and catches

Using logic constructs, such as operation-switch or if, you can execute different sections of the assembly when certain conditions are fulfilled. When the assembly branches, the subsection of the assembly contained by the construct is executed in the same manner as a complete assembly. However, contexts are shared with the complete assembly.

When a catch is triggered, either by an error occurring during the execution of a policy, or because a throw policy is encountered, the rest of the assembly flow is ignored. All contexts are shared by the catch being executed and when the end of the catch is reached, the API call is completed. There is no way to return from a catch to the rest of the assembly.

## Related concepts

- [Variable references in API Connect](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a new version of an API definition

You can create multiple versions of an API definition and edit the versions independently by using IBM® API Connect.

### About this task

---




You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

Note: If you want to publish a modified API to a production Catalog, you **must** create a new version of the API. You cannot publish an API to a production Catalog if there is already a published API with the same name and version.

### Procedure

---

To create a new version of an API definition, complete the following instructions:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. You can create a new version of an API definition either from the APIs and Products page, or from within the API definition itself.
  - a. To create a new version of an API definition from the APIs and Products page, complete the following steps:
    - i. Alongside the API version that you want to work with, click the options icon , then click Save as a new version. The Save API as a new version window opens.
    - ii. Enter your new version number, then click Save.
  - b. To create a new version of an API definition from within the API definition itself, complete the following steps:
    - i. Click the API definition that you want to create a new version of. The details of your API definition are displayed, and the API name and version are shown; for example:  

    - ii. Click the down arrow alongside the version, then select create new version. The Create New API Version window opens.
    - iii. Enter your new version number, then click Create.

Note: The version corresponds to the `info.version` property value of the API's OpenAPI definition. The `version.release.modification` version numbering scheme is recommended, for example 1.0.0.

### Results

---

You have created a new version of your API definition, which you can now edit independently of other versions. Each version of the API definition is listed separately on the APIs and Products page.

### Related tasks

---

- [Creating an API definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Specifying the gateway type for an API definition

An API definition is specific to one or other of the gateway types, DataPower® API Gateway or DataPower Gateway (v5 compatible). A default gateway type is set when you create an API definition, but you can edit the API configuration to specify a different gateway type.

### About this task

---

DataPower Gateway (v5 compatible) has been available with IBM® API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

For more information on how to choose which gateway type to use, see [API Connect gateway types](#).

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

You must specify which type of gateway each API uses. APIs can use only one type of gateway.

When you create an API, the gateway type defaults to that which matches the Gateway Service configuration for your Sandbox Catalog. If your Sandbox Catalog does not have a Gateway Service configured, or if the Sandbox Catalog has both types of Gateway Services configured, the gateway type defaults to DataPower Gateway (v5 compatible). You can use the user interface to update this setting.


Note that when you modify your API definitions to use a specific gateway type, you must ensure that each policy and policy version in the API are supported by the gateway type. DataPower Gateway (v5 compatible) and DataPower API Gateway each support policies that the other gateway type does not. In some cases, the same policy is supported by both gateway types, but with a different version number.


For example, DataPower Gateway (v5 compatible) supports version **1.0.0** of the `invoke` policy, but DataPower API Gateway requires version **2.0.0**.

For information on policies, see [execute](#).

For information on policy versions, see the documentation for each individual policy. For example, to review the **invoke** policy, see [invoke](#).

## Procedure

1. In the navigation pane, click  Develop, then select the APIs tab.
2. Click the API you want to update.
3. On the API Setup page, scroll to the Gateway Type section.
4. Select the type:
  - DataPower Gateway (v5 compatible)
  - DataPower API Gateway
5. Click Save to retain the changes.
6. Click the Assemble tab to ensure that your API uses policies that are supported by the gateway type you selected.

Policies in an existing API that are not supported by the new gateway type are grayed out and flagged with a red exclamation point: .

Some policies are supported by only one gateway type. Some policies are supported by both gateway types, but require a different version of the policy for each gateway type.

## Related tasks

- [Converting an API definition for deployment to the DataPower API Gateway](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Converting an API definition for deployment to the DataPower API Gateway

IBM® API Connect provides two gateway types, DataPower® Gateway (v5 compatible) and DataPower API Gateway. If you have an API definition that was developed for the DataPower Gateway (v5 compatible) and you want to port it to the DataPower API Gateway, you must make changes to convert it before deployment.

### About this task

DataPower Gateway (v5 compatible) has been available with IBM API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

To convert an API for deployment to the DataPower API Gateway, you must, as a minimum, change the gateway type setting on the API definition, and ensure that each policy and policy version in the API are supported by the DataPower API Gateway; DataPower Gateway (v5 compatible) and DataPower API Gateway each support policies that the other gateway type does not. In some cases, the same policy is supported by both gateway types, but with a different version number.

In addition, depending on the specific policies in your API assembly, some policies might require modification to adapt them for use with the DataPower API Gateway.

The following procedure provides instructions for converting your API definition:

## Procedure

1. Set the gateway type for the API to DataPower API Gateway, and ensure policy type and version compatibility. For details, see [Specifying the gateway type for an API definition](#).
2. Refer to the [DataPower API Gateway porting notes](#) for detailed instructions on how to convert your API definition.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API properties


Properties are used by the gateway to control behavior of certain policies. Typically, you provide properties, but the policy can also provide properties settings.

In IBM® API Connect, you can create API properties that consist of Catalog-specific values to eliminate the need for source code modifications. You can then reference the properties elsewhere in your API definition.

Pre-supplied API properties for various policies are shown in the tables.

A list of invoke related API properties that control the behavior of the invoke policy.

Table 1. Properties controlling the invoke policy

Property	Required	Description	Data type
 x-ibm-gateway-decode-request-params	No	If set to a value of <b>true</b> , any request parameters that are referenced by a variable definition on an invoke target-url are URL-decoded. The default behavior is to not decode any parameters, thereby sending them to the target URL without alteration.	Boolean



Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- invoke- suppress- clientid	No	When set with a value of <b>true</b> , or not specified, the X-IBM-Client-Id HTTP header (if specified on the API request) is suppressed from being sent to the invoke target URL. When set with a value of <b>false</b> , the X-IBM-Client-Id HTTP header is no longer suppressed from being sent to the invoke target URL. This property is supported only by the DataPower® Gateway (v5 compatible). If you are using the DataPower API Gateway then to achieve the same functionality add a <b>header-control</b> property to the <b>invoke</b> policy configuration in the OpenAPI definition for your API, as in the following examples.  Suppress the <b>X-Client-ID</b> header as follows: <pre>- invoke:   target-url: http://myhostname/mypath   header-control:     type: blacklist     values:       - ^X-Client-ID\$</pre> Suppress the <b>userId</b> query parameter as follows: <pre>- invoke:   target-url: http://myhostname/mypath   parameter-control:     type: blacklist     values:       - ^userId\$</pre>	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- optimize-invoke	No	If set to <b>false</b> , prevents the replacement of the last invoke in a policy with proxy. Any value other than <b>false</b> (case insensitive) will result in the last invoke in a policy possibly being replaced by proxy when the API is executed in the gateway.	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- queryparam- encode-plus- char	No	If set to a value of <b>true</b> , all "+" characters in the query parameter values of the target-url of the Invoke and Proxy policies are encoded to "%2F". The default value is <b>false</b> .	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- api-enforce- response-limits	No	If set to a value of <b>true</b> , allows the JSON parser to be enforced on the response rule. If the response body size is higher than the JSON parser limit set in the DataPower domain, a status code of 500 is returned. Note: The x-ibm-gateway-api-enforce-response-limits property is supported by the DataPower Gateway (v5 compatible) but not by the DataPower API Gateway. However, if you are using the DataPower API Gateway, consider using a <a href="#">Parse</a> policy in your API assembly to enforce these limits. For information on the different types of gateway, see <a href="#">API Connect gateway types</a> .	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- invoke-emulate- v4-soap-error	No	IBM API Management Version 4.0 initiates a DataPower error when a SOAP fault is returned from a web service. IBM API Connect provides a mechanism to catch SOAP errors and does not initiate a DataPower error. For compatibility with APIs developed in IBM API Management Version 4.0, set this property to <b>true</b> only in the case where a gateway extension is expecting to handle a SOAP error in a post error rule. The default value is <b>false</b> .  Note: This property is deprecated in favor of x-ibm-gateway-invoke-emulate-v4-invoke-error.	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- invoke-keep- payload	No	If set to a value of <b>true</b> , the invoke policy sends a payload on an HTTP DELETE method. This property is available for use with IBM DataPower Gateway version 7.7.1.1 and later. The default value is <b>false</b> .	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- invoke-emulate- v4-invoke-error	No	IBM API Management Version 4.0 initiates a DataPower error when a backend server error is returned, either a SOAP fault returned from a web service or a JSON or XML (non-SOAP) error from a restful service. IBM API Connect provides a mechanism to catch SOAP errors and operation errors and does not initiate a DataPower error when they occur. If no catch policy is configured, a generic error message is generated. For compatibility with APIs developed in IBM API Management Version 4.0, set this property to <b>true</b> only in the case where a gateway extension is expecting to handle a backend server error in a gateway extension post error rule or if the client of the API is expecting the backend server error to be returned. The default value is <b>false</b> .	Boolean

A list of map-related API properties that control the behavior of the map policy.

Table 2. Properties controlling the map policy

Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- map-array-first- element-value	No	In IBM API Management Version 4.0, if a mapping source value is from an array then only the first value is output. In API Connect, the default behavior is to return an array of all array element values. To maintain compatibility with IBM API Management Version 4.0, set this API property to <b>true</b> to only return the first array element value.	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- map-resolve- apic-variables	No	By default, any API Connect variable that is found in the map configuration is resolved. For example, <b>\$(request.headers.content-type)</b> resolves to the request's content type header. Because searching for variables in every map property can be CPU intensive, you can choose not to resolve variables by setting this API property to <b>false</b> . If this property is not configured or is set to any other value, the existing behavior to search for these variables continues. Note that variable usage within a map value JavaScript snippet is not changed provided that the variables that are referenced come from a configured map input.	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway- map-create- empty-array	No	This property controls how the map policy handles the output of an empty array; it can have the following values: <ul style="list-style-type: none"> <li><b>all</b>: Output all empty arrays, including empty children arrays. This is the default value if the property is not configured or has an invalid value.</li> <li><b>parent</b>: Output only the current property's empty array value. Children map actions of this property are not attempted.</li> <li><b>none</b>: Prevent any empty output array values from being produced.</li> </ul>	String

Property	Required	Description	Data type
<small>DataPower Gateway (5.5 compatible only)</small> x-ibm-gateway-optimize-schema-definition	No	Set the value of this property to <b>true</b> to provide a performance improvement to the map policy when a very complex schema definition is referenced by a policy output definition; for example, some very complex schemas that are generated by importing a very complex WSDL schema. The map policy builds a schema from an API definition when a referenced definition is provided as the value of the schema. If the schema does not have references that generate a circular reference, setting this property to <b>true</b> might provide a performance benefit while generating the same schema as would otherwise have been generated. However, in cases where the schema is very complex, with many potentially circular references, the generated schema could be different because the enhanced schema handling processes circular references differently. In such cases therefore, you should examine the resulting output to determine if the performance benefit gained is not at the expense of a change in the map policy output.  The default value of this property, is <b>false</b> , maintaining the existing behavior and performance.	Boolean
<small>DataPower Gateway (5.5 compatible only)</small> x-ibm-gateway-map-null-value	No	Set the value of this API property to <b>true</b> to allow a property from a map policy's input data with a value of <b>null</b> to be mapped to the output document. By default, a property from a map policy's input data with a value of <b>null</b> is not mapped to the output document.	Boolean
<small>DataPower Gateway (5.5 compatible only)</small> x-ibm-gateway-map-resolve-xmlinput-datatypes	No	XML input elements with numeric or boolean data have no metadata to indicate whether this data should be mapped as a string value or as the specific data type. If you set the value of this property to <b>false</b> , XML input elements are always mapped as a string. If you set the value to <b>true</b> , numeric or boolean XML input elements are mapped as the corresponding data type from the input schema. The default value is <b>false</b> .	Boolean
<small>DataPower Gateway (5.5 compatible only)</small> x-ibm-gateway-map-xml-empty-element	No	This property controls how the map policy handles XML input empty elements and impacts JSON output when the input document is XML; it can have the following values: <ul style="list-style-type: none"> <li>• <b>string</b>: The value for an empty XML element is considered to be an empty string. This is the default value if the property is not configured or has an invalid value.</li> <li>• <b>null</b>: The value for an empty XML element is considered to be null. A mapping of this element to a JSON output property does not occur unless the API property x-ibm-gateway-map-null-value is also specified with a value of <b>true</b>.</li> <li>• <b>none</b>: The empty XML element is ignored.</li> <li>• <b>string-badgerfish</b>: The value for an empty XML element is considered to be an empty string. The empty string value will be placed into a JSON badgerfish value property.</li> <li>• <b>null-badgerfish</b>: The value for an empty XML element is considered to be null. The null value will be placed into a JSON badgerfish value property. A mapping of this element to a JSON output property does not occur unless the API property x-ibm-gateway-map-null-value is also specified with a value of <b>true</b>.</li> </ul>	Boolean
<small>DataPower Gateway (5.5 compatible only)</small> x-ibm-gateway-schema-definition-reference-limit	No	Set the value of this property to an integer value that specifies the maximum allowed number of iterations of a circular schema definition. The default value is 1, which means that circular schema definitions are not followed. The maximum possible value is 5. If you specify a value greater than 5, a value of 5 is assumed. If you specify a non-numeric value, a value of 1 is assumed.	String

Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-map-emulate-v4-default-required-properties	No	<p>Set this property to true to have default values generated in the output for required properties that are either not mapped, or for which there is no input data present, in the following specific cases:</p> <ul style="list-style-type: none"> <li>An array consists of objects that contain one or more required properties.</li> <li>An object which is optional has one or more child properties that are required.</li> </ul> <p>By default, these required properties are not present in the output. If you set the <b>x-ibm-gateway-map-emulate-v4-default-required-properties</b> API property to <b>true</b>, these required properties will be present in the output. If the output schema defines a <b>default</b> property for the output property then the specified default value is used, otherwise a default value is assigned dependent on the data type, as follows:</p> <ul style="list-style-type: none"> <li>String: empty string ("")</li> <li>Number: 0</li> <li>Boolean: false</li> <li>Object: empty object</li> <li>Array: empty array</li> </ul> <p>Example 1 The input data has the following array of objects:</p> <pre>[{"a": "value1"}, {"a": "value2"}, {"b": "value3"}]</pre> <p>The output schema defines the output object as having two properties, <b>a</b> and <b>b</b>, of which <b>b</b> is required. The map policy defines the following mappings:</p> <ul style="list-style-type: none"> <li><b>input.array.a</b> to <b>output.array.a</b></li> <li><b>input.array.b</b> to <b>output.array.b</b></li> </ul> <p>If the <b>x-ibm-gateway-map-emulate-v4-default-required-properties</b> API property is set to <b>true</b>, and <b>b</b> is either not mapped or has no input data present, then <b>b</b> is assigned a default value of an empty string, and the output is as follows:</p> <pre>[{"a": "value1", "b": ""}, {"a": "value2", "b": "value3"}]</pre> <p>Example 2 The output schema defines the following structure:</p> <pre>{"a" : {"b" : {"c" : "value1", "d" : "value2"} } }</pre> <p>Property <b>b</b> is optional but property <b>d</b> within <b>b</b> is required.</p> <p>The map policy defines a mapping to <b>output.a.b.c</b>.</p> <p>If the <b>x-ibm-gateway-map-emulate-v4-default-required-properties</b> API property is set to <b>true</b>, and <b>d</b> is not mapped, then <b>d</b> is assigned a default value of an empty string, and the output is as follows:</p> <pre>{"a" : {"b" : {"c" : "value1", "d" : ""} } }</pre> <p>If the <b>x-ibm-gateway-map-emulate-v4-default-required-properties</b> API property is not specified or does not have a value of <b>true</b>, these required properties are <b>not</b> created in the output with their default values.</p>	Boolean
<small>DataPower Gateway (v5 compatible) only</small> ibm-gateway-map-post-process-json-output	No	<p>Set the value of this property to <b>true</b> to enable post processing of mapped JSON output. The post processing of JSON output will use the output schema to ensure that property values are of the same data type as that defined in the schema. It will also normalize output property values that have a Badgerfish JSON syntax due to object mapping of an XML input. Set the value to <b>false</b> for no post processing of mapped JSON output. The default value is <b>false</b>.</p>	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-map-emulate-v4-empty-json-object	No	<p>If a mapping fails because its input is not present and there is no default mapping configured, the default behavior is to not to make any change to the output mapping. Set the value of this property to <b>true</b> to create an empty object for the parent of the target mapping, emulating the behavior of IBM API Management Version 4.0.</p> <p>Example The map policy defines a mapping to <b>output.a.b.c</b>. If input data is present, the output is as follows:</p> <pre>{   "a": {     "b": {       "c": "inputvalue"     }   } }</pre> <p>If there is no input data, and the <b>x-ibm-gateway-map-emulate-v4-empty-json-object</b> API property is set to <b>true</b>, the output is as follows:</p> <pre>{   "a": {     "b": {     }   } }</pre> <p>Properties <b>a</b> and <b>b</b> are created but the value of <b>b</b> is an empty object. The default value is <b>false</b>.</p>	Boolean

DataPower Gateway (v5 compatible) only

A list of proxy-related API properties that control the behavior of the proxy policy.

Table 3. Properties controlling the proxy policy

Property	Required	Description	Data type
x-ibm-gateway-proxy-suppress-clientid	No	<p>A setting of <code>false</code> activates the injection of the X-IBM-Client-Id HTTP header (if it is specified on the API request), or the <code>client_id</code> query parameter in the request URL, to the proxy target-url. If not specified, or set with a value of <code>true</code>, suppresses the sending of this parameter to the proxy target-url.</p> <p>This property is supported only by the DataPower Gateway (v5 compatible). If you are using the DataPower API Gateway then to achieve the same functionality add a <code>header-control</code> property to the <code>invoke</code> policy configuration in the OpenAPI definition for your API, as in the following examples.</p> <p>Suppress the <code>X-Client-ID</code> header as follows:</p> <pre>- proxy:   target-url: http://myhostname/mypath   header-control:     type: blacklist     values:       - ^X-Client-ID\$</pre> <p>Suppress the <code>userId</code> query parameter as follows:</p> <pre>- proxy:   target-url: http://myhostname/mypath   parameter-control:     type: blacklist     values:       - ^userId\$</pre>	Boolean
x-ibm-gateway-optimize-invoke	No	If set to <code>false</code> , prevents the replacement of the last invoke in a policy with proxy. Any value other than <code>false</code> (case insensitive) will result in the last invoke in a policy possibly being replaced by proxy when the API is executed in the gateway.	Boolean
x-ibm-gateway-queryparam-encode-plus-char	No	If set to a value of <code>true</code> , all "+" characters in the query parameter values of the target-url of Invoke and Proxy policies are encoded to "%2F". The default value is <code>false</code> .	Boolean
x-ibm-gateway-api-enforce-response-limits	No	If set to a value of <code>true</code> , allows the JSON parser to be enforced on the response rule. If the response body size is higher than the JSON parser limit set in the DataPower domain, a status code of 500 is returned.	Boolean

A list of API properties that control the behavior of the rate limit policy.

Table 4. Properties controlling the rate limit policy

Property	Required	Description	Data type
x-ibm-gateway-emulate-v4-plan-rate-limit	No	By default in IBM API Connect Version 2018, if you configure a rate limit only for a Plan and not for the API operations within the Plan then a single rate limit threshold is set for the API as a whole, regardless of which operation in the API is requested. This behavior differs from IBM API Management Version 4.0 where the rate limit is set <b>individually</b> for <b>each</b> operation in the API. To change the Version 2018 behavior to emulate the Version 4.0 behavior, set this API property to a value of <code>true</code> .	Boolean

A list of API properties that control the behavior of multiple policies, or API behavior in general.

Table 5. Properties controlling multiple policies, or API behavior in general

Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-sourcecode-resolve-apic-variables	No	<p>If set to <code>true</code>, API Connect variable references are resolved. Set to <code>false</code> if you want the policy to ignore API Connect variable references.</p> <p>The default value is <code>true</code>.</p> <p>This property applies to the following policies:</p> <ul style="list-style-type: none"> <li>Map</li> <li>GatewayScript</li> <li>XSLT</li> <li>if</li> <li>switch</li> </ul> <p>Note: This property setting is overridden by the x-ibm-gateway-map-resolve-apic-variables API property setting for the Map policy.</p>	Boolean
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-api-json-parse-error-handling	No	<p>If an API request or response payload includes valid JSON content that contains characters that cannot be represented in the JSON XML internal syntax that is used by the DataPower Gateway, set this property to <code>escape-unicode</code> to allow the payload to be accepted without parsing errors. If this property is not configured or is set to any other value, the payload is rejected as invalid JSON.</p> <p>This property applies to the API request payload, and to the API response payload when <code>x-ibm-gateway-api-enforce-response-limits</code> is enabled.</p>	String
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-framework-preserve-escaped-reverse-solidus	No	By default, the string <code>\\</code> in a policy property is converted to a single <code>\</code> character. Set this property to <code>true</code> to preserve the string <code>\\</code> .	Boolean

Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-inspect-request-headers	No	Causes an inspection of the HTTP headers in the API request to check for characters in the header values that are illegal XML characters; it can have the following values: <ul style="list-style-type: none"> <li><b>default:</b> There is no inspection for these characters in the header values. If one is present, the API request fails with an HTTP 500 Internal Server Error.</li> <li><b>sanitize:</b> Any illegal XML characters in header values are replaced with a ? character. The API processing will continue. Any API that attempts to read <code>request.headers.&lt;headername&gt;</code> will see the ? character in the value. However, the original protocol headers representing <code>message.headers</code> will still have the original character, which will be sent to an invoke or proxy backend server.</li> <li><b>bad-request:</b> There will be an inspection for these characters in the header values. If one is present, the API request fails with an HTTP 400 Bad Request.</li> </ul> <p>The default value is <b>default</b>.</p>	Boolean
x-ibm-gateway-cors-allow-credentials-when-no-cors-policy	No	<b>From API Connect version 2018.4.1.17 onward:</b> By default, when CORS is enabled for an API, every response to a CORS request contains the following header:  <b>Access-Control-Allow-Credentials: true</b>  However, the inclusion of this header does not provide optimal security. To prevent this header from being returned, set the x-ibm-gateway-cors-allow-credentials-when-no-cors-policy property to the value <b>off</b> . If the property is not present, or is present with any other value, the header is returned.	String

A list of API properties that control the behavior of custom policies.

Table 6. Properties controlling custom policies

Property	Required	Description	Data type
<small>DataPower Gateway (v5 compatible) only</small> x-ibm-gateway-custom-policy-with-gws-action	No	If set to <b>true</b> , the <code>request.body</code> and <code>message.body</code> context variables will be populated for access by an <code>apim.getvariable('request.body')</code> or <code>apim.getvariable('message.body')</code> function call in a GatewayScript action of a custom policy. If the custom policy does not use a GatewayScript action that requires these variables to be populated, set this property to <b>false</b> or do not specify it.  The default value is <b>false</b> .	Boolean

- [Setting API properties](#)

In addition to the pre-supplied API properties that you can use to control the behavior of API Connect policies, you can define your own API properties. The properties that you define can be referenced in your API definitions.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Setting API properties

In addition to the pre-supplied API properties that you can use to control the behavior of API Connect policies, you can define your own API properties. The properties that you define can be referenced in your API definitions.

## About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

API properties include property name, value, and, optionally a specific Catalog to which a property value applies. For a list of pre-supplied API properties relating to various policies, see [API properties](#).

Note: Once defined, an API property is read only.


For information on how to reference a property in an API definition, see [Variable references in API Connect](#).

It is also possible to define properties that are specific to a Catalog and can be referenced by any of the APIs in that Catalog; for more information, see [Creating and configuring Catalogs](#). Note that if you define a Catalog property of the same name as an API property, the API property takes precedence over the Catalog property.

Tip: If you add or change an API property on an API that is already staged or published, you must re-stage or re-publish the Product that contains the updated API for the change to take effect.

## Procedure

To set API properties, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. Click the title of the API definition that you want to work with.
3. Click Properties, then click Add.
4. Enter a property Name and, optionally, a Default value and a Description. The following character set is supported for the Name of an API property: `[A-Za-z0-9_-]+`. No spaces are allowed.
  - Note: If you publish the API to the DataPower® Gateway (v5 compatible) the property name must begin with either a letter or a \_ (underscore) character. This restriction does not apply if you publish the API to the DataPower API Gateway. For information on the different types of gateway, see [API Connect gateway types](#).
5. Select Encoded if you want to hide the property values, or protect user passwords from casual observance.
  - Note: If you encode a property value, it is saved in Base64 encoded form; it is **not** encrypted. If you subsequently clear the Encoded check box, the original property value is restored in its unencoded form.
6. To define a property value that is specific to a particular Catalog, complete the following steps:

- a. In the Define catalog specific values for this property section, select the required Catalog and enter the corresponding value.
- Note: The Catalogs that you can select from are those that are defined for the management server and provider organization that you are connected to. If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

For details of how to create a Catalog in a provider organization, see [Creating and configuring Catalogs](#).

- b. To define further Catalog/value pairs, click Add.
7. Click Save when done.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Variable references in API Connect

In API Connect you can reference different variables in your API definition.

When defining an API, creating a custom policy, or configuring another policy or logic construct, you can include references to context variables and properties.

Variable references are resolved either when the API is staged in a Product, for static variables that are fixed upon staging, or when the API is called, for variables that can change with each API call.

## Types of variables

### Context variables

A context variable is a variable relevant during an API call, for example, the input of the call, the path of the call, or the message during the call. A context variable is one of the variables that makes up that particular context.

Context variables can consist of more than one part, for example, `request.headers`.

For a list of available context variables, see [API Connect context variables](#).

### API properties

An API property is a variable in an API where its value depends upon the Catalog in which the API is staged or published. By referencing an API property, you can use the same API definition in different Catalogs where there are small differences between the instances of the API between the Catalogs. For example, an assembly could contain an `if` construct that executes its case when a particular Catalog is used, determined from the value of the API property. API properties can also be used to hide a value such as a password by encoding the value.

API properties are referenced by name.

For a list of API properties, see [API properties](#)

For more information, see [Setting API properties](#).

Note: Once defined, an API property is read only.

### Catalog properties


A Catalog property is specific to a Catalog, and can be referenced in any of the API definitions in that Catalog. For more information, see [Creating and configuring Catalogs](#).


Note:

- If you change the value of a Catalog property, any API that references that property must be republished for it use the new value.
- Catalog properties and API properties are not supported with global policies. Therefore, if you use such properties in a global policy, they are **not** replaced with the values specified in the Catalog or API property definitions.

## Methods of referencing variables

You can get or set the value for a referenced variable.

- Get the value for a variable in either of the following ways:
  - Through the [GatewayScript](#) policy, run the `apim.getvariable()` API.
  -  Through the [GatewayScript](#) policy, use the `context.get()` function; for more information, see [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).
  - Through the [XSLT](#) policy, run a stylesheet that uses the `apim:getVariable` extension function.
  - In an assembly policy field that supports variable references, use the following syntax:  

```
$(variable)
```
- Set the value for a variable in either of the following ways:
  - Through the [Set Variable](#) policy.
  - Through the [GatewayScript](#) policy, use the `apim.setvariable()` API.
  -  Through the [GatewayScript](#) policy, use the `context.set()` function; for more information, see [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).
  - Through the [XSLT](#) policy, run a stylesheet that uses the `apim:setVariable` extension element.

## GatewayScript references

When you want to reference a variable in a GatewayScript context, use one of the following methods:

```
apim.getvariable(variable)
```

where *variable* is the name of the context variable or API property that you want to reference.

```
apim.setvariable(variable, value, action)
```

where

- *variable* is the name of the context variable or API property that you want to reference.
- *value* is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, use the following code:

```
var contentType = apim.getvariable('request.headers.content-type');
apim.setvariable(variable, contentType, 'set');
```

This property is required only when **set** or **add** is specified as the action.

- *action* is the action that you want to apply to the variable. Valid options are:
  - **set**
  - **add**
  - **clear**

If no option is set, the default option of **set** is applied.

Use the **getvariable** method to retrieve the value of a context variable or API property, and the **setvariable** method to change one.

Some of the situations where you would use this type of reference are:

- GatewayScript policies. -For more information, see [GatewayScript](#).
- **if** logic constructs. For more information, see [if](#).
- User-defined policies. For more information, see [Authoring policies](#).

## Stylesheet references

You can reference a variable by using functions and elements in an XSLT policy with the following syntax:

```
<xsl:variable name="variable_name" select="apim:getVariable(variable)" />
```

where *variable* is a literal value, another variable, or a valid XSLT XPath statement.

```
<xsl:call-template name="apim:setVariable">
  <xsl:with-param name="varName" select="variable"/>
  <xsl:with-param name="value" select="value"/>
  <xsl:with-param name="action" select="action"/>
</xsl:call-template>
```

where

- *variable* is the name of the context variable or API property that you want to reference. This can be a literal value, another variable, or a valid XSLT XPath statement.
- *value* is the string value that you want to set the variable to. This can be a literal value, another variable, or a valid XSLT XPath statement. This property is required only when **set** or **add** is specified as the action.
- *action* is the action that you want to apply to the variable. This can be a literal value, another variable, or a valid XSLT XPath statement. Valid options are:
  - **set**
  - **add**
  - **clear**

If no option is set, the default option of **set** is applied.

The following example sets a named variable to the value of the Content-Type header in a request:

```
<xsl:variable name="contentType" select="apim:getVariable('request.headers.content-type')" />
<xsl:call-template name="apim:setVariable">
  <xsl:with-param name="varName" select="'variable'"/>
  <xsl:with-param name="value" select="$contentType"/>
  <xsl:with-param name="action" select="'set'"/>
</xsl:call-template>
```

## Inline references

In many situations you can make a simpler reference, by using the following syntax:

```
$(variable)
```

where *variable* is the name of the context variable or API property that you want to reference.

Some of the situations where you would use this type of reference are:

- The URL called by an Invoke or Proxy policy. For more information about the policies, see [Invoke](#) or [Proxy](#).
- A Map policy. For more information, see [Map](#).

Note: The map policy can reference the following variables inline:

- Variables that are defined as inputs to the map policy and specified in the from field of a mapping.
- Context variable or API properties, provided that the **x-ibm-gateway-map-resolve-apic-variables** API property is not set to **false**. If an inline reference in a map policy resolves to a context variable or API property, it is immediately replaced by the corresponding value. For more information on the **x-ibm-gateway-map-resolve-apic-variables** API property, see [Properties controlling the map policy](#).

For more information on the use of inline references, and examples, see the following [About inline references](#) section.

## About inline references



When you use an inline reference for a property value in an assembly policy, the variable reference is replaced with the corresponding string value at run time, before the property is evaluated. For example, consider the following incoming request, which has two headers and two query parameters (a `curl` command is used to illustrate the request):

```
curl --data-binary @request.json \  
-H 'Content-Type: application/json' \  
-H 'X-Environment: test' \  
http://apiserver/org/catalog/petstore/getPetDetails?petid=285&storeid=z03
```

If an `invoke` policy is configured as follows:

```
invoke:  
target-url: https://backend/GetPetInfo/$(request.parameters.storeid)/$(request.parameters.petid)
```

then the value of the `target-url` property is evaluated as `https://backend/GetPetInfo/z03/285`, and this is therefore the URL to which the `invoke` policy sends its request.

[DataPower Gateway \(v5 compatible\) only](#) You can also use an inline reference in the `condition` property of a `switch` policy. Consider the following example:

```
- switch:  
  case:  
    - condition: "$(request.headers.X-Environment)" == "production"  
      execute:  
        - invoke:  
            http://www.finance.com/base/stockquote  
    - condition: "$(request.headers.X-Environment)" == "test"  
      execute:  
        - invoke:  
            http://testserver1.finance.com/stockquote
```

Using the incoming request referred to previously, the value of the `X-Environment` header is `"test"`. The inline references are first evaluated, then the conditions are evaluated as JavaScript. Therefore, the first condition is evaluated as `"test" == "production"`, returning `false`, and the second condition is evaluated as `"test" == "test"`, returning `true`, so the `invoke` policy sends its request to `http://testserver1.finance.com/stockquote`.

Important: This example, which encloses the inline references inside double quotes, assumes that the API is deployed to the DataPower API Gateway. If you deploy the API to the DataPower Gateway (v5 compatible), then the double quotes are added implicitly when the inline references are evaluated, and you must therefore omit them from the `condition` property; the two conditions in this example would therefore be

```
condition: $(request.headers.X-Environment) == "production"
```

and

```
condition: $(request.headers.X-Environment) == "test"
```

However, as an alternative to using an inline reference in a `condition`, you can instead use pure JavaScript, as follows:

```
- switch:  
  case:  
    - condition: request.headers["X-Environment"] == "production"  
      execute:  
        - invoke:  
            http://www.finance.com/base/stockquote  
    - condition: request.headers["X-Environment"] == "test"  
      execute:  
        - invoke:  
            http://testserver1.finance.com/stockquote
```

in which case the policy configuration will work correctly as-is in both types of gateway.

For more information on the types of gateway, see [API Connect gateway types](#).

In addition to single strings, you can use an inline reference for a complete object. Consider the following example, which uses an inline reference in a `set-variable` policy:

```
- set-variable:  
  actions:  
    - set: saved.reqHdrs  
      value: $(request.headers)  
      type: string
```

Here, the `request.headers` structured JSON object is substituted with its corresponding string representation, and a variable named `reqHdrs` is therefore created, under the `saved` object, with a string value of `{"Content-Type": "application/json", "X-Environment": "test", ...}`. However, if you are using the DataPower API Gateway, you can specify `type: any`, in which case the `set-variable` policy copies the value as-is, and `saved.reqHdrs` is created as a JSON object rather than a string:

```
- set-variable:  
  actions:  
    - set: saved.reqHdrs  
      value: $(request.headers)  
      type: any
```

You can reference a property in a JSON or XML request directly by name, by using the following syntax:

```
$(request.body.property_name)
```

For example, if the request contains `{ "account-number": 123 }` or `<account-number>123</account-number>` you can retrieve the value `123` by using the following inline reference:

```
$(request.body.account-number)
```

If the property value is within a nested property structure, you can reference it by concatenating the property names. For example, if the request contains `{ "account": { "balance": 123 } }` or `<account><balance>123</balance></account>` you can retrieve the value `123` by using the following inline reference:

```
$(request.body.account.balance)
```



Similarly, you can reference a property in a JSON or XML response by using the following syntax:

`$(message.body.property_name)`

Restriction: If you are using the DataPower API Gateway, you can reference properties in this way only if the format of the request or response is JSON; the mechanism is **not** supported with XML request or response formats. If the you are using the DataPower Gateway (v5 compatible) the mechanism is supported with both JSON and XML formats.

## Related concepts

- [The behavior of an assembly](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Activating an API

After you have created an API definition you can activate it to make it available for testing.

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

When you activate an API, API Connect automatically completes the following actions:

- Creates a draft Product, adds the API to the Product, and publishes the Product to the Sandbox Catalog so that the API is available to be called. The Product has the title `api_title` auto product. Note that if you later want to delete the draft Product, you cannot delete it directly; instead, delete the API and the draft Product is deleted together with the API; see [Deleting an API definition](#). If you want to remove the Product from any Catalogs to which it is published, you must do this separately; see [Removing a Product from a Catalog](#)
- Subscribes the Sandbox test application to the Product so that you can immediately test the API in the test environment. For information on testing an API, see [Testing an API with the assembly test tool](#).

To activate an API, you must be assigned a role that has the `Product:Manage` and `Subscription:Manage` permissions. The pre-supplied Developer role has these permissions by default; if you are assigned a custom role it must have these permissions. For more information, see [Creating custom roles](#).

Note:

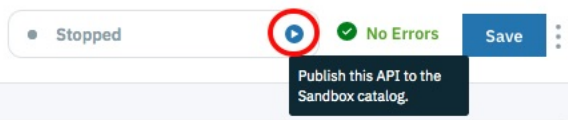
- The API activation will not complete successfully if lifecycle approval is enabled in the Sandbox Catalog for the Stage, Publish, or Replace actions. If any such lifecycle approvals are enabled, then to be able to activate and API they must be disabled; for information on lifecycle approval settings, see [Creating and configuring Catalogs](#).
- To activate an API from the API Designer user interface, you must be connected to a Management server; API activation is not available with API Designer in offline mode.
- Products that contain an API with a Swagger property using `regex` that include lookahead assertions, such as "(?)" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!  
The multipart 'openapi' field contains an OpenAPI definition with validation errors.  
definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0,  
col: 0)  
400
```

## Procedure

To activate an API, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. Click the API definition that you want to work with.
3. Click the Publish this API to the Sandbox catalog icon:

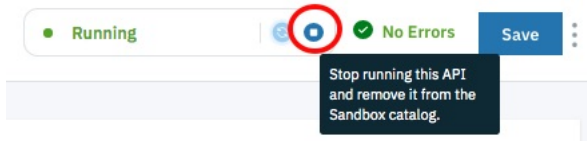


## Results

On successful completion, the API is shown as Running:



You can stop the API by clicking the Stop running this API icon:

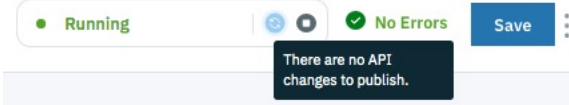


If you stop an API, the application subscription is deleted, and the auto Product is removed from the Sandbox Catalog. You can republish a running API by clicking the republish icon:

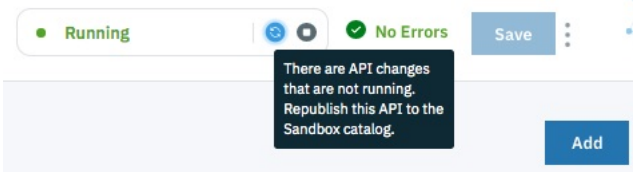


The republish icon changes color to indicate whether you have made changes that require the API to be republished:

- Republish not required:



- Republish required:

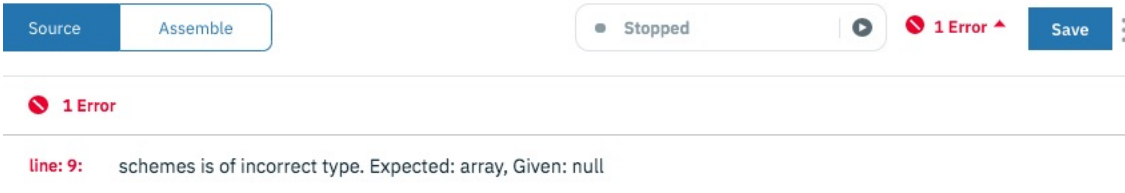


The error indicator shows whether there are validation errors in the OpenAPI source for the API definition. If there are errors, click the icon for more details:

- No validation errors:



- Validation error:



You can also activate an API during the creation process, and on the API test page; see [Creating an API definition](#) and [Testing an API with the assembly test tool](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Testing an API with the assembly test tool

IBM® API Connect provides a test environment for you to ensure that your APIs are defined and implemented correctly.

### Before you begin

If you want to test an API offline, API Connect provides the following options:

- Call the API in the Local Test Environment with a cURL command, as described in [Testing an API with the Local Test Environment](#).
- Invoke the API from the locally installed API Designer UI application running in Online mode as described in the following sections.

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.




Note:

- If you are testing an API that contains references to API properties, only those references that are defined inside the API assembly are resolved and replaced with their corresponding values when you invoke the API in the assembly test tool; property references that are defined outside of the API assembly are not resolved. For more information on API properties, see [Setting API properties](#).
- Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

## Procedure

---

To test an API, complete the following steps.

1. If you are using API Designer, set the mode to Online using the Options menu  on the main page.
2. In the navigation pane, click  Develop, then select the APIs tab.
3. Click the API definition that you want to work with.
4. Click Assemble to open the Assemble view, then click the Test icon .
5. If you are testing the API for the first time and, when you created the API definition, you selected the Activate API option, your test setup will already be configured and you can proceed immediately to the next step to test your API. Otherwise, click Activate API to have your test setup configured.  
Note: If you are retesting your API after making changes, click Republish product to make your changes available.
6. In the Operation section, select the API operation that you want to test, then click Invoke.  
The API response is displayed in the Response section.  
Note: If you receive a message relating to an untrusted certificate, click the link provided, accept the certificate, then return to the test environment and click Invoke again. The message also mentions a lack of CORS support on the server, but this is just one possible cause for the connection failing.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Testing an API with the Local Test Environment

Use the Local Test Environment to test APIs on your local machine, without the need to connect to an API Connect management server. The Local Test Environment is a lightweight API Manager running on your local machine. It allows you to rapidly test APIs locally.

Note: A later version of the Local Test Environment is available. For details, see the [IBM® API Connect Version 10 Local Test Environment documentation](#).  
API Connect provides the following methods for testing an API on your local machine:

- Invoke the API from the API Designer UI application running in Online mode as described in [Testing an API with the assembly test tool](#).
- Call the API in the Local Test Environment with a cURL command, as described in the following sections.

Important: The current version of the Local Test Environment is a beta release.

## Prerequisites

---

- The API Connect developer toolkit, including the API Designer user interface, installed. For installation and running instructions, see [Installing the toolkit](#).
- The Local Test Environment and the API Designer must be downloaded from the same API Connect release page on [IBM Fix Central](#), in order for them to work together. For example, for API Connect Version 2018.4.1.7, use the distributions of Local Test Environment and API Designer from <https://www.ibm.com/support/pages/ibm-api-connect-v2018417-ifix15-available>.
- Docker installed.  
Note: The Local Test Environment is not supported with Docker Version 18.09.x.
- A minimum of 4 GB of RAM available to Docker if a single gateway type is used, or 6 GB if both the DataPower® API Gateway and DataPower Gateway (v5 compatible) are used.  
Note: As you increase the number of APIs that are published to your gateways, you will need to allocate further memory to Docker. You will also need to start the Local Test Environment with a larger database; see [apic-lte start](#).
- If you are using Windows, ensure that your C: drive (or the drive on which your HOME directory is located, if different), is enabled as a shared drive so that the Local Test Environment files are accessible to the Docker containers.

## Installing the Local Test Environment

---

There are two options for installing the Local Test Environment:

- Each user downloads the Local Test Environment images to their local machine and installs the Local Test Environment from there.
- One user downloads the Local Test Environment images and uploads them to a private Docker registry, from where any user can install the Local Test Environment.

To install the Local Test Environment from your local machine, complete the following steps:

1. Download the `apic-lte-version.zip` file from IBM Fix Central, and extract the contents, which are as follows:
  - `apic-lte-images.tar.gz`, which contains all required Docker images.
  - Separate `apic-lte` binary files for the Mac OS X, Linux®, and Windows platforms, in respective subfolders.
2. Load the Docker images into your local Docker image repository by entering the following command:

```
docker load < apic-lte-images.tar.gz
```

To upload the Local Test Environment images to a private Docker registry, complete the following steps:

1. Download the `apic-lte-version.zip` file as described in step [1](#).
2. Distribute the appropriate `apic-lte` binary file to all users.
3. Upload the Local Test Environment to your private Docker registry; enter the following command:

```
apic-lte registry-upload apic-lte-images.tar.gz registry_host
```

where `registry_host` is the host name or IP address of your private Docker registry. Now, any user can install and run the Local Test Environment as follows:  
a. If the private Docker registry requires authentication, log in by entering the following command:

```
docker login registry_host
```

b. Load the Docker images into your local Docker image repository by entering the following command:

```
apic-lte init registry_host
```

## Starting the Local Test Environment

1. Start the Docker images by entering the following command:

```
apic-lte start
```

Note:

- By default, the `apic-lte start` command starts only a DataPower API Gateway. To also start a DataPower Gateway (v5 compatible), enter the following command:

```
apic-lte start --datapower-gateway-enabled --datapower-api-gateway-enabled
```

- The Local Test Environment might fail to start with an error message that includes the strings `Error: certificate is not yet valid` and `CERT_NOT_YET_VALID`. The most likely cause is that the date and time setting is incorrect on the machine that is running the Local Test Environment. Ensure that the date and time setting is correct, before attempting the start command again. If you are using Docker for Windows, the clock in the Docker containers can become out of sync with the system clock, especially after a machine has been put in sleep mode. In this case, restarting Docker should fix the clock discrepancy; for more information, see <https://github.com/docker/for-win/issues/4526>.

2. Verify that the Local Test Environment is installed and running correctly by entering the following commands:

a. `apic-lte status`

This output from this command shows the status of all components, and provides endpoint and authentication details, and should be similar to the following:

```
Container          Status
-----
apic-lte-apim      Up 3 minutes
apic-lte-datapower-gateway Not Running
apic-lte-datapower-api-gateway Up 2 minutes
apic-lte-db        Up 3 minutes
apic-lte-juhu      Up 3 minutes
apic-lte-lur       Up 3 minutes

- Platform API url: https://localhost:2000
- Admin user: username=admin, password=7iron-hide
- 'localtest' org owner: username=shavon, password=7iron-hide
- 'localtest' org sandbox test app credentials client id: 80963e74076afe50d346d76401c3c08a
- Datapower API Gateway API base url: https://localhost:9444/localtest/sandbox/
```

b. `apic login --server localhost:2000 --username shavon --password 7iron-hide --realm provider/default-idp-2`

This command confirms that you can log in to the management server, and the response should be as follows:

```
Logged into localhost:2000 successfully
```

## Preparing an API for testing in the Local Test Environment

To prepare an API for testing in the Local Test Environment, you must publish it to the Sandbox Catalog in the Local Test Environment. If you want to test an API that you already published, proceed to [Testing an API in the Local Test Environment](#), otherwise, complete the following steps:

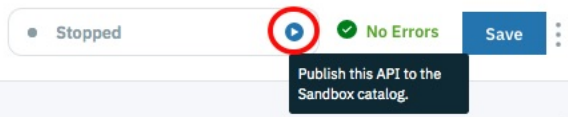
- Launch the API Designer user interface.
- Open the required local directory; this is the directory in which your API and Product definition files will be stored.
- Connect to the Local Test Environment. If you haven't previously connected to the Local Test Environment, click Add Another Cloud, then complete the following steps:

- In the HOST URL field, enter `https://localhost:2000`, then click Next..
- In the Username field, enter `shavon`, in the Password field enter `7iron-hide`, then click Sign in.

If you have previously connected to the Local Test Environment, click the existing tile to log in immediately.

The API Designer welcome page opens.

- Click Develop APIs and Products, then click the API that you want to test. For details on how to configure an API definition, see [Developing your APIs and applications](#).
- Click the Publish this API to the Sandbox catalog icon:



When the publish operation completes, your API is ready for testing.

Note: Whenever you make any changes to an API, you must republish it before retesting.

## Testing an API in the Local Test Environment.

To test an API in the Local Test Environment, issue a REST API call to the following URL:

```
https://localhost:9444/localtest/sandbox/basepath/operation_path?client_id=lte_client_id
```

where:

- `basepath` is the base path that is configured in the API definition.

- `operation_path` is the path for the operation that you want to invoke, as configured in the API definition.
- `lte_client_id` is the client ID for the test application in the local test environment, as returned by the `apic-lte status` command in step 2.

The following example show how to test the API that is created in the tutorial [Creating a proxy REST API definition](#), by using the `curl` utility; the API returns the details of bank branches:

```
curl -k https://localhost:9444/localtest/sandbox/branches/details?client_id=80963e74076afe50d346d76401c3c08a
[{"id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5", "type": "atm", "address": {"street1": "600 Anton Blvd.", "street2": "Floor
5", "city": "Costa Mesa", "state": "CA", "zip_code": "92626"}},
{"id": "9d72ece0-7e7b-11e5-9038-55f9f9c08c06", "type": "atm", "address": {"street1": "4660 La Jolla Village Drive", "street2": "Suite
300", "city": "San Diego", "state": "CA", "zip_code": "92122"}},
{"id": "ae648760-7e77-11e5-8059-a1020f32cce5", "type": "atm", "address": {"street1": "New Orchard
Road", "city": "Armonk", "state": "NY", "zip_code": "10504"}},
{"id": "c23397f0-7e76-11e5-8059-a1020f32cce5", "type": "branch", "phone": "512-286-5000", "address": {"street1": "11400 Burnet
Rd.", "city": "Austin", "state": "TX", "zip_code": "78758-3415"}},
{"id": "ca841550-7e77-11e5-8059-a1020f32cce5", "type": "atm", "address": {"street1": "334 Route
9W", "city": "Palisades", "state": "NY", "zip_code": "10964"}},
{"id": "dc132eb0-7e7b-11e5-9038-55f9f9c08c06", "type": "branch", "phone": "978-899-3444", "address": {"street1": "550 King
St.", "city": "Littleton", "state": "MA", "zip_code": "01460-1250"}},
{"id": "e1161670-7e76-11e5-8059-a1020f32cce5", "type": "branch", "phone": "561-893-7700", "address": {"street1": "5901 Broken Sound
Pkwy. NW", "city": "Boca Raton", "state": "FL", "zip_code": "33487-2773"}},
{"id": "f9ca9ab0-7e7b-11e5-9038-55f9f9c08c06", "type": "atm", "address": {"street1": "1 Rogers
Street", "city": "Cambridge", "state": "MA", "zip_code": "02142"}]}
```

## Local Test Environment commands

The following table summarizes the Local Test Environment commands; use the `help` command to get full usage details for any command.

Table 1. Local Test Environment command summary

Command	Description
<code>apic-lte help command</code>	Display help information for any command.
<code>apic-lte init</code>	Download the Local Test Environment Docker images.
<code>apic-lte start</code>	Start the Local Test Environment Docker images. Use the <code>--database-max-heap-size</code> parameter to set the size of the Local Test Environment database, in bytes; for example:  <pre>apic-lte start --database-max-heap-size 4096M apic-lte start --database-max-heap-size 1G apic-lte start --database-max-heap-size 1048576K apic-lte start --database-max-heap-size 1073741824</pre> <p>The default value is <b>1024M</b>. Tip: By default, the <code>apic-lte start</code> command deletes all previous data and re-initializes the Local Test Environment configuration, so all your previous configuration, including published Products, is deleted. To retain the previous configuration, and to apply the same command parameters that were used in the previous <code>apic-lte start</code> command, supply the <code>--keep-config</code> parameter.</p>
<code>apic-lte status</code>	Display status information for the Local Test Environment components, and endpoint and authentication details.
<code>apic-lte stop</code>	Stop the Local Test Environment Docker images.
<code>apic-lte version</code>	Display Local Test Environment version information.

## Troubleshooting the Local Test Environment

You can consult the log file for each Local Test Environment microservice or database by using the following command:

```
docker logs container-name
```

where `container-name` is one of the following:

- `apic-lte-juhu`: the authentication gateway
- `apic-lte-apim`: the API Management service
- `apic-lte-lur`: the Local User Registry
- `apic-lte-db`: the Cassandra database of the API Management service
- `apic-lte-datapower-api-gateway`: the DataPower API Gateway
- `apic-lte-datapower-gateway`: the DataPower Gateway (v5 compatible)

You can access the gateway logs in either of the following ways:

- Use the gateway administration web UI:
  1. Open the page `https://localhost:web_ui_port` in a browser; for details of the required port value, see [Local Test Environment port values](#).
  2. Select the apiconnect domain and the WebGUI interface, and log in with user name `admin` and password `admin`.
  3. Click View Logs.
- Use the gateway administration CLI:
  1. Open an SSH connection by using the following command:

```
ssh -p gateway-ssh-port localhost
```

For details of the required port value, see [Local Test Environment port values](#). The user name is `admin` and the password is `admin`.

2. Enter the command `switch domain apiconnect`.
3. To view the gateway log, enter the command `show log`.
4. To view the log for the communication between the gateway and the API management system, enter the command `show logging gwd-log`.

## Local Test Environment port values

If any of the default port values for the Local Test Environment components conflict with ports already in use on your system, you can change them when you start the Local Test Environment by passing one or more `--component port_value` parameters to the `apic-lte start` command, where:

- `component` is the Local Test Environment component whose port value you want to change.
- `port_value` is the required value.

For example:

```
apic-lte start --datapower-api-gateway-api-port 9445
```

The following table lists the components, together with the corresponding `component` parameters, and the default port values:

Component	component parameter	Default port value
DataPower API Gateway API port	<code>datapower-api-gateway-api-port</code>	9444
DataPower API Gateway API Connect service port	<code>datapower-api-gateway-apic-service-port</code>	3001
DataPower API Gateway SSH port	<code>datapower-api-gateway-ssh-port</code>	9023
DataPower API Gateway administration web UI	<code>datapower-api-gateway-web-gui-port</code>	9091
DataPower Gateway (v5 compatible) API port	<code>datapower-gateway-api-port</code>	9443
DataPower Gateway (v5 compatible) service port	<code>datapower-gateway-apic-service-port</code>	3000
DataPower Gateway (v5 compatible) SSH port	<code>datapower-gateway-ssh-port</code>	9022
DataPower Gateway (v5 compatible) administration web UI	<code>datapower-gateway-web-gui-port</code>	9090

## Related information

- [Creating TLS Client Profile in the Local Test Environment](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Staging an API

The graphical wizard provides an option that adds the API to a Product and stages the Product in a Catalog. When a Product is in the staged state, it is not yet visible to, or subscribable by, any developers. The syndication feature in IBM® API Connect means that if Spaces are enabled for a Catalog, Products can be staged only to a Space within that Catalog.

## Before you begin

Ensure that you have a Catalog to stage to in the API Manager or API Designer user interfaces (UI). For more information, see [Creating and configuring Catalogs](#).

Ensure that the Catalog has at least one gateway service configured.

Note: All references in this topic to a Catalog can also be applied to Spaces in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in IBM API Connect®](#).

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Product > Stage permission for the target Catalog or Space. For information on configuring Product management permissions for a Catalog or Space, see [Creating and configuring Catalogs](#) or [Managing user access in a Space](#).

## About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI. Staging is not available when working offline in API Designer.


A Catalog is a staging target, and behaves as a logical partition of the DataPower® Gateway, and the Developer Portal.

Validation of the API OpenAPI definition file occurs during the staging or publishing process. The following validation occurs:


- Validation against the OpenAPI specification schema
- Validation against IBM extension properties
- Semantic validation, which includes the following types of validation:
  - Ensuring that if an API is enforced by an API Connect Gateway, then the scheme must be HTTPS, or the parameter name for an API key security scheme in the header must be either `X-IBM-Client-Id` or `X-IBM-Client-Secret`.
  - Ensuring that if the API is not enforced by an API Connect Gateway, then a "host" must be provided

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published (the `$ref` field is supported only if you are using the API Connect for IBM Cloud developer toolkit). For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

## Procedure

1. In the navigation pane, click  Develop.

The Develop: APIs and Products tab opens.

- Optional:** If you have accounts on multiple provider organizations, you can select a new provider organization for staging and publishing from the Organization menu.
- You can stage an API either from the APIs and Products listing page, or from within the API definition itself.
  - To stage an API from the APIs and Products listing page, click the options menu icon  alongside the required API, and then select Stage.
  - To stage an API from within the API definition, complete the following steps:
    - Click the API definition that you want to work with.
    - Click the options menu icon:



iii. Click Stage.

4. Choose one of the following actions:

- To stage the API by adding it to a new Product:
  - Select New Product - Publish using a new product.
  - When prompted, enter a Title and Version.
  - The Product Name is automatically entered. The Name is the used to refer to the product in CLI commands. See [apic products](#).
  - Click Next.
- To stage the API by adding it to an existing Product:
  - Select Existing Product - Publish using an existing product
  - Select the Product you want to use.
  - Click Next.

5. On the Stage To page, select the Catalog to which you want to stage the Product.

Note: The Catalogs that you can select from are those that are defined for the management server and provider organization that you are connected to.


If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

For details of how to create a Catalog in a provider organization, see [Creating and configuring Catalogs](#).

6. If, when the staged Product is subsequently published, you want it to be published only to selected gateway services, select Publish to specific gateway services, then select the required gateway services. Only the gateway services whose type matches the gateway type setting for the Product are listed. For information on gateway types, see [API Connect gateway types](#).

7. Click Stage.

## Results

Your Product is staged to a Catalog. You can view the state of the Product in the Catalog in API Manager. If you staged the product from API Designer, ensure you are logged into API Manager with the same user name and password that you used for API Designer. Click  Manage in the API Manager UI, then select the required Catalog. The Product is shown with a state of Staged.

For information about the lifecycle of a product, see [The Product lifecycle](#).

If approval is required to stage Products in the Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is staged when the request is approved. If approval is not required, the Product is staged immediately.

For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Publishing an API

The graphical wizard provides an option that adds the API to a Product and publishes the Product in a Catalog. Publishing a draft Product makes the APIs in that Product visible on the Developer Portal for use by application developers. The syndication feature in IBM® API Connect means that if Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog.

## Before you begin

Ensure that you have a Catalog to stage to in the API Manager or API Designer user interfaces (UI). For more information, see [Creating and configuring Catalogs](#).

Ensure that the Catalog has at least one gateway service configured.

Note: All references in this topic to a Catalog can also be applied to Spaces in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in IBM API Connect®](#).

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Product<sub>z</sub> Manage permission for the target Catalog or Space. For information on configuring Product management permissions for a Catalog or Space, see [Creating and configuring Catalogs](#) or [Managing user access in a Space](#).

## About this task



You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI. Publishing is not available when working offline in API Designer.

Validation of the API OpenAPI definition file occurs during the staging or publishing process. The following validation occurs:



- Validation against the OpenAPI specification schema
- Validation against IBM extension properties
- Semantic validation, which includes the following types of validation:
  - Ensuring that if an API is enforced by an API Connect Gateway, then the scheme must be HTTPS, or the parameter name for an API key security scheme in the header must be either `X-IBM-Client-Id` or `X-IBM-Client-Secret`.
  - Ensuring that if the API is not enforced by an API Connect Gateway, then a "host" must be provided

Note: Products that contain an API with a Swagger property using `regex` that include lookahead assertions, such as "(?" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!
The multipart 'openapi' field contains an OpenAPI definition with validation errors.
definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0, col:
0)
400
```

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published (the `$ref` field is supported only if you are using the API Connect for IBM Cloud developer toolkit). For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).


## Procedure

1. In the navigation pane, click  Develop.  
The Develop: APIs and Products tab opens.
2. **Optional:** If you have accounts on multiple provider organizations, you can select a new provider organization for staging and publishing from the Organization menu.
3. You can publish an API either from the APIs and Products listing page, or from within the API definition itself.
  - a. To publish an API from the APIs and Products listing page, click the options menu icon  alongside the required API, and then select Publish.
  - b. To publish an API from within the API definition, complete the following steps:
    - i. Click the API definition that you want to work with.
    - ii. Click the options menu icon:



- iii. Click Publish.
4. Choose one of the following actions:
    - To publish the API by adding it to a new Product:
      - Select New Product - Publish using a new product.
      - When prompted, enter a Title and Version.
      - The Product Name is automatically entered. The Name is the used to refer to the product in CLI commands. See [apic products](#).
      - Click Next.
    - To publish the API by adding it to an existing Product:
      - Select Existing Product - Publish using an existing product
      - Select the Product you want to use.
      - Click Next.
  5. Select the Catalog to which you want to publish the Product.
  6. If Spaces have been enabled in the selected Catalog, select the Space that you require.  
Note: The Catalogs that you can select from are those that are defined for the management server and provider organization that you are connected to.  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).  
For details of how to create a Catalog in a provider organization, see [Creating and configuring Catalogs](#).
  7. If you want to publish the Product only to selected gateway services, select Publish to specific gateway services, then select the required gateway services. Only the gateway services whose type matches the gateway type setting for the Product are listed. For information on gateway types, see [API Connect gateway types](#).
  8. Click Publish.

## Results

Your Product is published to a Catalog. You can view the state of the Product in the Catalog in API Manager. If you published the product from API Designer, ensure you are logged into API Manager with the same user name and password that you used for API Designer. Click  Manage in the API Manager UI, then select the required Catalog. The Product is shown with a state of Published.

For information about the lifecycle of a product, see [The Product lifecycle](#).

If approval is required to publish Products in the Catalog, a publish approval request is sent, and the Product moves to the Pending state; the Product is published in the Catalog when the request is approved. If approval is not required, the Product is published immediately.

Note: If approval is required to stage Products to the Catalog, a stage approval request is sent. When the request is approved, the Product moves to the staged state and must be separately published in the Catalog. For information on publishing a staged Product in a Catalog, see [Publishing a Product](#).  
For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Downloading an API definition

You can download the details of your API definitions so that you can store them for later recovery.

### Before you begin

To complete this task, you must have created an API definition. For more information, see [Creating an API definition](#).

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

**API Manager UI only:** To complete this task, you must be assigned a role that has the **Drafts:View** permission. The pre-supplied Developer role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).



You can download REST APIs that were created by using [Creating a REST proxy API from a target service](#) for example. This download creates a .yaml file that defines the API.

You can download SOAP APIs that were created by using [Creating a REST proxy API from an existing WSDL service](#), or [Creating a SOAP proxy API from an existing WSDL service](#) for example. This download creates a .zip file that contains the WSDL definition and the YAML file that defines the API.

You can use the downloaded .yaml and .zip files to re-create your API. For information about creating an API from a .yaml file, see [Adding a REST API by importing an OpenAPI definition file](#). For information about creating an API from a .zip file, see [Adding a SOAP API by importing a zip file](#).

### Procedure

To download a .yaml, or .zip file that contains the details of your API, complete the following steps:

1. In the navigation pane, click  Develop.
2. Alongside the API version that you want to work with, click the options icon  and then click Download.
3. Save the file to the required location.

### Results

You have downloaded a .yaml representation of your API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting an API definition



You can delete an API definition that is no longer required.

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

**API Manager UI only:** To complete this task, you must be assigned a role that has the **Api-Drafts:Edit** permission. The pre-supplied Developer role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).

### Procedure

1. In the navigation pane, click  Develop.
2. Alongside the API version that you want to delete, click the options icon  and then click Delete.  
Note: If a Product was previously generated automatically from the API, by using the Activate API option, that Product will be deleted together with the API; such Products have the title *api\_title* auto product.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using an options file when importing a WSDL service

When you create an API definition, or add a target WSDL service to an API definition, by importing a .zip file, you can specify additional directives by including an options file in the .zip file.

You can use an options file when importing a .zip file while performing any of the following tasks:

- Creating a REST proxy API; see [Creating a REST proxy API from an existing WSDL service](#).
- Creating a SOAP proxy API; see [Creating a SOAP proxy API from an existing WSDL service](#).
- Adding a target WSDL service to an API definition; see [Editing an API definition](#).

The options file is a YAML file, and must have the file name apiconnect.yaml.

You can include the following fields in the file:

Table 1. Fields that can be included in the apiconnect.yaml file

Field name	Value	Default	Description
<code>suppressExamples</code>	true or false	false	Suppress auto-generation of examples.
<code>wssecurity</code>	true or false	true	Enable generation of definitions for WS-Security headers.
<code>implicitHeaderFiles</code>	Array of XSD file locations		Additional schema files, not referenced in the WSDL, that are used to define additional SOAP headers.
<code>port</code>	The <code>name</code> attribute for a <code>wsdl:port</code> in a <code>wsdl:service</code> definition.		Create the API by using the information for the specified port. If the specified port does not exist, or refers to a REST/XML port rather than a SOAP port, the API creation fails.  If no <code>port</code> field is present in the options file, API Connect creates the API by using the first SOAP port that it finds in the <code>wsdl:service</code> definition.

## Example

```
suppressExamples: true
wssecurity: false
implicitHeaderFiles:
  - xsdDir/schema1.xsd
  - xsdDir/schema2.xsd
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API policies and logic constructs

Policies and logic constructs are pieces of configuration that control a specific aspect of processing in the Gateway server during the handling of an API invocation at run time.

Policies are the building blocks of assembly flows, and they provide the means to configure capability, such as security, logging, routing of requests to target services, and transformation of data from one format to another. Policies can be configured in the context of an API or in the context of a Plan.

Logic constructs behave in a similar way to policies, but they affect how and which parts of the assembly are implemented without modifying the data flow of the assembly.

IBM® API Connect provides the following ways that you can create, configure, and apply policies and logic constructs:

### Policies associated with a Plan

A Plan provides a mechanism for grouping API operations or subsets of operations from one or more APIs. You can set rate limiting policies on a Plan to specify how many requests an application is allowed to make during a specified time interval. You can also configure a policy for each operation that is included in a Plan. For more information, see [Working with Products](#).

### Built-in policies

A built-in policy enables you to apply a pre-configured policy statement to an assembly to control processing capabilities in the Gateway server. Built-in policies are applied by using the API Manager assembly editor to add a built-in policy to your assembly and to configure the properties for that policy. For more information, see [Built-in policies](#).

Note: You can also apply built-in policies to your APIs by adding an `assembly` extension to your OpenAPI definition file. For more information, see [IBM extensions to the OpenAPI specification](#).

### Logic constructs

A logic construct enables you to control the flow of data through your assembly during an API call. Like policies, logic constructs are applied to an API by using the API Manager assembly editor to add a logic construct to your assembly and to configure the behavior of the construct. For more information, see [Logic Constructs](#).

Note: You can also apply logic constructs to your APIs by adding an `assembly` extension to your OpenAPI definition file. For more information, see [IBM extensions to the OpenAPI specification](#).

### User-defined policies

A user-defined policy enables you to create your own policies to control extra processing features in the Gateway server, such as security, or routing of requests. User-defined policies are created outside of API Connect and then imported into one or more Catalogs, so they can be applied to an operation in the same way as built-in policies. For more information, see [Importing a user-defined policy into a Catalog](#).

#### • [Logic Constructs](#)

IBM API Connect includes a number of logic constructs that you can use to apply preconfigured logic to an assembly to control the flow of data through your assembly when the API is called.

#### • [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#)

In the code for GatewayScript and XSLT policies in your API assemblies, you can use API context variables to work with the messages that are generated when an API is called.

## Related reference

- [execute](#)

## Related information



- [Working with API definitions](#)
- [Including components in your assembly](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Built-in policies

IBM® API Connect includes a number of built-in policies that you can use to apply preconfigured policy statements to an operation to control an aspect of processing in the Gateway server when an API is invoked.

Note: Although some built-in policies can be used with both the DataPower® Gateway (v5 compatible) and the DataPower API Gateway, some policies are restricted to a particular Gateway. The following icons indicate which Gateway each policy can be used with:

-  Indicates that the policy can be run on the DataPower Gateway (v5 compatible).
-  Indicates that the policy can be run on the DataPower API Gateway.



For details of the two types of gateway, see [API Connect gateway types](#).

Built-in policies are configured in the context of an API. You can use the API Manager assembly editor to add a built-in policy to an API and to configure the properties for that policy.

You can also add built-in policies to an API by creating an OpenAPI definition file. For more information, see [Creating an OpenAPI definition file](#).

The following table shows the list of built-in policies that are available. The table contains links to configuration information for both the built-in policy definitions, and the OpenAPI policy definitions. The policies are the same, but they are created in different ways.

Table 1. Built-in policies

Built-in policy	OpenAPI policy	Description		
<a href="#">Activity Log</a>	<a href="#">activity-log</a>	Use the Activity Log policy to configure your logging preferences for the API activity that is stored in IBM API Connect analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity. Note: The Activity Log policy is not supported in the assembly for an API whose gateway type is DataPower API Gateway. Instead, you configure activity logging in the API design settings. For details, see <a href="#">Activity logging with the DataPower API Gateway</a> .	✓	✓ Functionality provided in the API design; see <a href="#">Activity logging with the DataPower API Gateway</a> .
<a href="#">Client Security</a>	<a href="#">client-security</a>	Provides a range of options for authenticating client access to your APIs, extending the capabilities of the OpenAPI specification.	✗	✓ Available from V2018.4.1.5
<a href="#">GatewayScript</a>	<a href="#">gatewayscript</a>	Use the gatewayscript policy to execute a specified DataPower GatewayScript program.	✓	✓ Available from V2018.4.1.0
<a href="#">Generate JWT</a>	<a href="#">jwt-generate</a>	Use the Generate JWT security policy in IBM API Connect to generate a JSON Web Token (JWT).	✓	✓
<a href="#">Validate JWT</a>	<a href="#">jwt-validate</a>	Use the Validate JWT security policy to enable the validation of a JSON Web Token (JWT) in a request before allowing access to the APIs.	✓	✓
<a href="#">if</a>	<a href="#">if</a>	Use the if policy to apply a section of the assembly when a condition is fulfilled.	✓	✓ Available from V2018.4.1.0; functionality provided by <a href="#">switch</a>
<a href="#">Invoke</a>	<a href="#">invoke</a>	Apply the Invoke policy to call another service from within your assembly. The response from the backend is stored either in the variable <code>message.body</code> or in the response object variable if it is defined. The policy can be used with JSON or XML data, and can be applied multiple times within your assembly.	✓	✓
<a href="#">JSON to XML</a>	<a href="#">json-to-xml</a>	Use the JSON to XML policy to convert the context payload of your API from the JavaScript Object Notation (JSON) format to the extensible markup language (XML) format.	✓	✓
<a href="#">Log</a>	<a href="#">log</a>	Use the Log policy to customize or override the default activity logging configuration for an API.	✗	✓ Available from V2018.4.1.7
<a href="#">Map</a>	<a href="#">map</a>	Use the Map policy to apply transformations to your assembly flow and specify relationships between variables.	✓	✓

Built-in policy	OpenAPI policy	Description	DataPower Gateway	API Gateway
<a href="#">operation-switch</a>	<a href="#">operation-switch</a>	Use the operation-switch policy to apply a section of the assembly to a specific operation.	✓	✓ Available from V2018.4.1.0; functionality provided by <a href="#">switch</a>
<a href="#">OAuth</a>	<a href="#">oauth</a>	Use the OAuth policy to policy to perform OAuth processing based on defined OAuth provider settings.	✗	✓
<a href="#">Parse</a>	<a href="#">parse</a>	Use the Parse policy to control the parsing of an input document. When the input document is a JSON string, the string is parsed instead of copied over.	✗	✓
<a href="#">Proxy</a>	<a href="#">proxy</a>	Apply the Proxy policy to invoke another API within your assembly, particularly if the separate API contains a large payload. The response from the backend is stored in the <code>message.body</code> and in the response object variable if it is defined. Only one policy is permitted to be run per unique assembly flow.	✓	✓ Functionality provided by <a href="#">Invoke</a>
<a href="#">Rate Limit</a>	<a href="#">ratelimit</a>	Use the Rate Limit policy to apply one or more rate or burst limits at any point in your API assembly flow. Rate and burst limits restrict the number of calls that an application can make to an API in a specified time period.	✗	✓ Available from V2018.4.1.7
Redaction <a href="#">Redaction - DataPower API Gateway</a> <a href="#">Redaction - DataPower Gateway (v5 compatible)</a>	<a href="#">redact - DataPower API Gateway</a> <a href="#">redact - DataPower Gateway (v5 compatible)</a>	Use the Redaction policy to completely remove or to redact specified fields from the Request body, the Response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.	✓	✓ Available from V2018.4.1.7
<a href="#">Set Variable</a>	<a href="#">set-variable</a>	Use the Set Variable policy to set the value of a runtime variable, or to clear a runtime variable, or to add a header variable.	✓	✓ Available from V2018.4.1.0
<a href="#">switch</a>	<a href="#">switch</a>	Use the switch policy to execute one of a number of sections of the assembly based on which specified condition is fulfilled.	✓	✓ Available from V2018.4.1.0
<a href="#">throw</a>	<a href="#">throw</a>	Use the throw policy to throw an error when it is reached during the execution of an assembly flow.	✓	✓ Available from V2018.4.1.0
<a href="#">User Security</a>	<a href="#">user-security</a>	Use the user-security policy to extract a user's credentials, authenticate those credentials, and obtain authorization from the user.	✗	✓
Validate <a href="#">Validate - DataPower API Gateway</a> <a href="#">Validate - DataPower Gateway (v5 compatible)</a>	<a href="#">validate - DataPower API Gateway</a> <a href="#">validate - DataPower Gateway (v5 compatible)</a>	Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.	✓	✓ Available from V2018.4.1.0
<a href="#">Validate Username Token</a>	<a href="#">validate-username-token</a>	Use the Validate Username Token policy to validate a Web Services Security (WS-Security) UsernameToken in a SOAP payload before allowing access to the protected resource.	✓	✗
<a href="#">XML to JSON</a>	<a href="#">xml-to-json</a>	Use the XML to JSON policy to convert the context payload of your API from the extensible markup language (XML) format to JavaScript Object Notation (JSON).	✓	✓
<a href="#">XSLT</a>	<a href="#">xslt</a>	Use the XSLT policy to apply an XSLT transform to the payload of the API definition.	✓	✓ Available from V2018.4.1.0

## Related tasks

- [Including components in your assembly.](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Activity Log

Use the Activity Log policy to configure your logging preferences for the API activity that is stored in IBM® API Connect analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [activity-log](#).

Note that if you are using the DataPower API Gateway, you can configure your logging preferences by using the Activity Log tab, in the Design view of the Develop API editor of the API Manager UI. For more information, see [Activity logging with the DataPower API Gateway](#).

## About

An API event record exists for each API execution event in the Gateway server. By default, the content type that is collected and stored in API event records is **activity** for when API execution completes successfully, and **payload** for when API execution completes with an error code. Apply the Activity Log policy to your assembly to change the type of content to log in these API event records. For more information about API event records, see [API event record fields](#).

You can attach this policy to the following API flows:

- REST
- SOAP

Note:

Activity log policies that call for logging of Analytics data upon success do not apply for the OAuth provider. The OAuth provider logs Analytics data for failure cases, but does not log successful cases.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Activity Log policy properties

Property label	Required	Description	Data type
Title	Yes	A title for the policy is required, but a default value, <b>activity-log</b> is provided.	string
Description	No	A description of the policy.	string
Content	Yes	Defines the type of content to be logged when the operation is successful. Valid values: <ul style="list-style-type: none"> <li>• <b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li>• <b>activity</b>: Logs invocation only (only the resource URI is recorded).</li> <li>• <b>header</b>: Logs activity and header.</li> <li>• <b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <b>activity</b> .	string
Error content	No	Indicates what content to log if an error occurs. Valid values: <ul style="list-style-type: none"> <li>• <b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li>• <b>activity</b>: Logs invocation only (only the resource URI is recorded).</li> <li>• <b>header</b>: Logs activity and header.</li> <li>• <b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <b>payload</b> .	string

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related information

- [API Analytics](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Client Security

The Client Security policy provides a range of options for authenticating client access to your APIs, extending the capabilities of the OpenAPI specification.

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.5	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [client-security](#).

## About

You use the Client Security policy in an API assembly to define how clients that call the API must supply authentication credentials.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Client Security policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>client-security</b> .	string
Description	No	A description of the policy.	string
Stop on Error	Yes	If this setting is enabled, assembly processing stops if client security fails, and an error is returned. The check box is selected by default.	boolean
Secret Required	Yes	If this setting is enabled, the client secret must be sent in the request. The secret is compared to the registered secret on the application that is identified by the client ID. The check box is selected by default.	boolean
Credential Extraction Method	Yes	Select one of the following options to define how the calling application authenticates: <ul style="list-style-type: none"> <li>Header: client ID and client secret credentials must be supplied in the request header.</li> <li>Query: client ID and client secret credentials must be supplied as query parameters in the request URL.</li> <li>Form: client ID and client secret credentials must be supplied as form data sent in a POST request.</li> <li>Cookie: client ID and client secret credentials must be supplied in a header called <b>Cookie</b>.</li> <li>HTTP: the calling application must authenticate by using basic authentication.</li> <li>Context Variable: the credentials that are used for authenticating the client are obtained from context variables that you set in the assembly flow prior to the Client Security policy, by using a <a href="#">GatewayScript</a> policy for example. The names of these context variables are determined by the values that you supply in the ID Name and Secret Name properties of the Client Security policy.</li> </ul> The default value is Header.	string
ID Name	Yes, unless the selected value of Credential Extraction Method is HTTP	The name of the parameter whose value specifies the client ID. For all Credential Extraction Method options other than Context Variable and HTTP, the calling application must supply a parameter with this name, in the location defined by the Credential Extraction Method option. For the Context Variable option, this property specifies the name of a context variable. This property does not apply if the Credential Extraction Method is HTTP.	string
Secret Name	Yes, if Secret Required is enabled and the selected value of Credential Extraction Method is other than HTTP	The name of the parameter whose value specifies the client secret. For all Credential Extraction Method options other than Context Variable and HTTP, the calling application must supply a parameter with this name, in the location defined by the Credential Extraction Method option. For the Context Variable option, this property specifies the name of a context variable. This property does not apply if the Credential Extraction Method is HTTP.	string
HTTP Type	Yes, if the selected value of Credential Extraction Method is HTTP	The authentication type. Currently, the only available option is Basic.	
Authenticate Client Method	Yes	Select one of the following options: <ul style="list-style-type: none"> <li>Native: only client ID and client secret are used to authenticate the client. If the selected value of Credential Extraction Method is HTTP then the calling application must supply the client ID for the user name, and the client secret for the password.</li> <li>Third party: a user registry is used to authenticate the client. If the selected value of Credential Extraction Method is other than HTTP then the calling application must supply the user name for the client ID, and the password for the client secret.</li> </ul> The default value is Native.	string
User Registry Name	Yes, if the selected value of Authenticate Client Method is Third party	Select the user registry that will be used to authenticate the client. The supported registry types are LDAP and authentication URL.	string

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## GatewayScript

Use the `gatewayscript` policy to execute a specified DataPower® GatewayScript program.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [gatewayscript](#).

## About

You can attach this policy to the following API flows:

- REST
- SOAP

The gatewayscript policy gives you built-in access to the DataPower Gateway module via variable `apim`.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. gatewayscript policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <code>gatewayscript</code> .	string
Description	No	A description of the policy.	string
Source	Yes	The GatewayScript source code to execute. For example: <pre>var message = [ 'Hello', 'World!' ]; console.debug(message.join(' '));</pre>	string

## Examples

The following examples show how the full OpenAPI for the policy looks in the source code.

Example one:

```
gatewayscript:  
  title: writes message to DataPower log  
  source: console.debug('Hello World!');
```

Example two:

```
gatewayscript:  
  title: script written in multiple lines  
  source: |  
    var message = [ 'Hello', 'World!' ];  
    console.debug(message.join(' '));
```

For more code examples, see [GatewayScript code examples](#) and, if you are using the DataPower API Gateway, [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

For general information on using GatewayScript, see the following topics in the DataPower product documentation:

- [GatewayScript APIs for API management](#)
- [Accessing and manipulating a variable in the API context](#)
- [OAuth context variables](#)

## Errors

The following error can be thrown while the policy is being executed:

- `JavaScriptError` - a generic error that captures all errors that occur during the execution of the policy.

## Related tasks

- [Creating a new REST OpenAPI definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## GatewayScript code examples

Example GatewayScript code snippets to help the creation of a gatewayscript policy.

Note:

- The GatewayScript functions that are listed here use the common name `apim` to call the methods. However, you can change the common name to one of your choice by using one or other of the following function calls depending on your gateway type:

```
DataPower Gateway (v5 compatible) var name = require('./apim.custom.js');
API Gateway var name = require('apim');
```

where `name` is your chosen name; for example:

```
var apim = require('apim');
```

- The mechanisms described here for interacting with the API Connect context are compatible with Version 5. However, if you are writing new GatewayScript policy, XSLT policy, or user-defined policy code, the preferred mechanisms, which provide better performance, are as described on [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

## Access the assembly context

The following code snippets show examples of how to access the assembly context.

Example one returns the `request` context:

```
apim.getvariable('request');
```

Example two returns the header named `foo`:

```
apim.getvariable('request.headers.foo');
```

Example three shows how to set, add, or clear a context variable, in this case a message header:

```
apim.setvariable('message.headers.name', value, action)
```

where

- `name` is the name of the message header that you want to set, add, or clear.
- `value` is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, you would specify the `value` as `request.headers.content-type`. This property is only required when `set` or `add` is specified as the action.
- `action` is the action that you want to apply to the variable. Valid options are:
  - `set`
  - `add`
  - `clear`

If no option is set, the default option of `set` is applied.

For a complete list of context variables, see [API Gateway context variables](#), and for more information about how to reference context variables in IBM API Connect see [Variable references in API Connect](#). If you are using the DataPower® API Gateway, see also [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

## Read input synchronously

The following code snippets show examples of how to read input synchronously by using the `apim.getvariable()` function to read data direct from a context. JSON input is returned as a JavaScript object. XML data is returned as a NodeList object (DOM).

Example one returns a synchronous read of the original request body into the variable that is called `json`:

```
var json = apim.getvariable('request.body');
```

Example two returns a synchronous read of the current message into the variable that is called `xml`:

```
var xml = apim.getvariable('message.body');
```

Note: If the content type of the data is neither XML nor JSON, the `apim.getvariable()` function returns a NodeList object consisting of one binary node, which must be converted to a string. If the content really is XML or JSON, you must then parse this string, as illustrated in the following example:

```
var data = apim.getvariable('request.body');

// if a nodelist was returned and the type is 13 (BLOB/Binary Node), convert it
// to a Buffer, which can then be converted to a string
if ((data.item && data.length > 0 && data.item(0).nodeType === 13)) {
  data = data.item(0).toBuffer().toString();
}

// if the string really is XML or JSON, then now add code to parse it with
// the appropriate XML or JSON parse function
.
.
.
```

## Read input asynchronously

The following code snippets show the `apim.readInput()` calls that you can use to perform a JavaScript callback function to read input data asynchronously into a variable. JSON input is returned as a JavaScript object. XML data is returned as a NodeList object (DOM). Other types of input are returned as a Buffer object.

```
apim.readInput(callback(err, input) {});
apim.readInputAsJSON(callback(err, json) {});
```



```
apim.readInputAsXML(callback(err,nodelist) {});
apim.readInputAsBuffer(callback(err,buffer) {});
```

Note: The `apim.readInput()` function attempts to determine the input data type and call the appropriate `apim.readInputAsType()` function in order to return the data in the correct format. However, to ensure that the correct data type is returned, use the `apim.readInputAsType()` function.

The `apim.readInputAsType()` function reads the data from either the INPUT context (for example `session.INPUT`), or from a policy output context (for example `policy.output`), depending on whether a previous policy had been called that had written output to a policy output context. This function then calls the underlying IBM DataPower GatewayScript `readAsType` method on the relevant context to read the data, and then use JavaScript callbacks to return the data to a variable. For more information about DataPower methods, see [APIs for methods](#).

The following is an example of how to use the `apim.readInputAsJSON()` callback function to get data into a variable that is called `json`:

```
apim.readInputAsJSON(function (error, json) {
  if (error)
  {
    // handle error
  }
  else
  {
    // the json parameter will contain the json data that has been read
    // process the json object in some way and write to the output context
    if (json.test=='true')
    {
      // if json data contains a variable called test that is set to true, then also add some test data
      json.data = 'This is my test data'
    }
    session.output.write(json);
    // write to the output context
  }
});
```

## Configure error information

The following code block shows an example of how to configure the policy implementation to produce error information by using the `apim.error()` function. In this example, `MyError` is thrown to the assembly flow. If there is a catch handler it catches this error, otherwise the assembly skips the execution of the flow and sends a `500 Internal Error` response.

```
apim.error('MyError', 500, 'Internal Error', 'Some error message');
```

where:

- `MyError` is the name of the error.
- `500` is the HTTP code of the required error message.
- `Internal Error` is the HTTP reason phrase for the error.
- `Some error message` is the suggested action for the user.

## Accessing the caught exception in a catch block

The following example shows how, in the `catch` block of an API assembly, you can obtain the details of the current caught exception. A possible use would be to create a custom error response using the details of the caught exception.

```
let exception = apim.getError();
```

The function returns a JSON object; for example:

```
{
  "name": "OperationError",
  "message": "This is a thrown Operation Error",
  "policyTitle": "Throw Operation Error",
  "status": {
    "code": "500",
    "reason": "Internal Server Error"
  }
}
```

## Write output

The following example shows how to write data into the output, in this case the JSON data `{ "status": "created" }`, by using the `session.output.write` variable:

```
session.output.write('{ "status": "created" }');
```

The `session.output.write` variable is a standard GatewayScript method that writes data to the output context. For more information, see [Contexts and sessions](#). Use the following function to set the format of the content that is written to the `session.output` variable:

```
apim.output('application/type');
```

For example, if you were creating a policy that called:

```
session.output.write('<test>this is some xml data</test>');
```

the policy would write XML data to the output context. You would need to configure the policy to also call

```
apim.output('application/xml');
```

to tell the system that the output is XML. It can cause issues in the processing of the policy if the actual output context format, and the format that is specified in `apim.output`, are different.

Note: If you use `apim.setvariable` to manipulate `message.body`, you must immediately follow the `apim.setvariable` function call with an `apim.output` function call that specifies the content type of the payload that was written to `message.body` by `apim.setvariable`.

## Related information

- 🔗 [GatewayScript APIs for API management](#)
- 🔗 [Accessing and manipulating a variable in the API context](#)
- 🔗 [OAuth context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Generate JWT

Use the Generate JWT security policy in IBM® API Connect to generate a JSON Web Token (JWT).

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [jwt-generate](#).

## About

JSON Web Token (JWT) is a compact, URL-safe way of representing claims that are to be transferred between two parties. The Generate JWT policy enables you to generate claims and configure whether they are to be used as the payload of a JSON Web Signature (JWS) structure, or as the plain text of a JSON Web Encryption (JWE) structure. Specifying the cryptographic material for both the JWS and the JWE produces a nested JWT that is both digitally signed and encrypted. The JWT is then assigned to the Authorization header as a Bearer token (the default option), or to the runtime variable in the JSON Web Token (JWT) property, if specified.

You can attach this policy to the following API flows:

- REST
- SOAP

Note:

- For algorithm types HS256, HS384, and HS512 the cryptographic objects referenced must be a Shared Secret Key.
- For algorithm types RS256, RS384, RS512, ES256, ES384, ES512, PS256, PS384, PS512 the cryptographic objects referenced must be a Crypto Key (private key).
- The cryptographic material can be provided through a JSON Web Key (JWK).
- If both a cryptographic object and a JWK are specified, the cryptographic object is used to sign the JWT.

## Prerequisites

The following prerequisites apply:

- If you are using one or more cryptographic objects, they must be located in the API Connect domain on the DataPower appliance. The cryptographic objects must reference the Shared Secret Key or certificate that is needed to encrypt or sign the JWT contents.
- If a JSON Web Key (JWK) is being used, it must be referenced by a runtime variable.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Generate JWT policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <code>jwt-generate</code> .	string
Description	No	A description of the policy.	string
JSON Web Token (JWT)	No	Runtime variable in which to place the JWT that is generated. The default value is: <code>generated.jwt</code> . However, if not set, the JWT that is generated is written to the Authorization Header as a Bearer token.	string
JWT ID Claim	No	Indicates whether a JWT ID (jti) claim should be added to the JWT. If selected, the property is set to <code>true</code> , and a UUID is generated and set as the JTI claim value.	boolean
Issuer Claim	Yes	Runtime variable from which the Issuer (iss) claim string can be retrieved. This claim represents the Principal that issued the JWT. The default value is: <code>iss.claim</code>	string

Property label	Required	Description	Data type
Subject Claim	No	Runtime variable from which the Subject (sub) claim string can be retrieved.	string
Audience Claim	No	Runtime variable from which the Audience (aud) claim string can be retrieved. Multiple variables are set by using a comma-separated string.	string
Validity Period	Yes	The length of time (in seconds), that is added to the current date and time, in which the JWT is considered valid. The default value is <b>3600</b> .	integer
Private Claims	No	Runtime variable from which a valid set of JSON claims can be retrieved. These claims are added to any set of claims specified previously.	string
Sign JWK variable name	No	Runtime variable that contains the JWK that is used to sign the JWT. <sup>1</sup>	string
Cryptographic Algorithm	No	The cryptographic algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>• HS256</li> <li>• HS384</li> <li>• HS512</li> <li>• RS256</li> <li>• RS384</li> <li>• RS512</li> <li>• ES256</li> <li>• ES384</li> <li>• ES512</li> <li>• PS256</li> <li>• PS384</li> <li>• PS512</li> </ul>	string
Sign Crypto Object	No	The cryptographic object to use to sign the JWT. <sup>1</sup>	string
Encryption Algorithm	No	The encryption algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>• A128CBC-HS256</li> <li>• A192CBC-HS384</li> <li>• A256CBC-HS512</li> </ul>	string
Encrypt JWK variable name	No	Runtime variable that contains the JWK to use to encrypt the JWT.	string
Key Encryption Algorithm	No	The key encryption algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>• RSA1_5</li> <li>• RSA-OAEP</li> <li>• RSA-OAEP-256</li> <li>• dir</li> <li>• A128KW</li> <li>• A192KW</li> <li>• A256KW</li> </ul>	string
Encrypt Crypto Object	No	The cryptographic object to use to encrypt the claim.	string

## Example

```
- jwt-generate:
  version: 1.0.0
  title: jwt-generate
  iss-claim: iss.claim
  exp-claim: 3600
  jwt: generated.jwt
  jti-claim: true
  sub-claim: sub.claim
  aud-claim: aud.claim
  private-claims: private.claims
  jws-jwk: jws.jwk
  jws-alg: HS256
  jws-crypto: jwsCryptoObjectName
  jwe-enc: A128CBC-HS256
  jwe-jwk: jwe.jwk
  jwe-alg: A128KW
  jwe-crypto: jweCryptoObjectName
```

## Errors

The following error can be thrown while the policy is being executed:

- **RuntimeError** - a generic error that captures all errors that occur during the execution of the policy. Upon failure, the detailed error message that is received from the underlying JOSE module is written to the default system log as an error message. This detailed error message is also assigned to the runtime variable `jwt-generate.error-message`, so it can be retrieved via `catch`.

If a `catch` is not configured, in the case of a failure the Generate JWT policy returns an HTTP code 500 **Invalid-JWT-Generate** failure. The detailed error message from the underlying JOSE module can be found in the system log.

Attention: If you are an API developer who is troubleshooting a failure that one of your customers had with your API, consider the security risks before sending a customer the exact content of the log message. You can avoid the possibility of someone launching an attack based on the information that they receive from the log message by sending the customer only general information about the message.

## Related information

- [Introduction to JSON Web Tokens](#)

<sup>1</sup> A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to sign the JWT. However, if both data types are specified, only the Crypto Object is used.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Validate JWT

Use the Validate JWT security policy to enable the validation of a JSON Web Token (JWT) in a request before allowing access to the APIs.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [jwt-validate](#).

## About

JSON Web Token (JWT) is a compact, URL-safe way of representing claims that are to be transferred between two parties. The Validate JWT policy enables you to secure access to your APIs by using JWT validation. For example, when an input request that contains a JWT in the header is received, the Validate JWT policy extracts the token, verifies, and decrypts (if appropriate) the signature, and validates the claim. If valid, the claim is put in a runtime variable (for subsequent use if required), and access is allowed to the API. If the claim is not valid, access is denied.

All claims that are specified in the Validate JWT policy are validated, but this is not necessarily all the claims that are contained in the JWT. Not all claims that are in the JWT must be validated, but if any one of the claims that are specified in the Validate JWT policy fail, the whole validation fails. If the validation succeeds, the full set of claims that are contained in the JWT are written to the runtime variable specified in the Output Claims property. Thereby allowing any subsequent action to use this runtime variable to further validate the full set of claims that were in the JWT, as necessary.

You can attach this policy to the following API flows:

- REST
- SOAP

Note:

- If the original message was signed with a Shared Secret Key, the cryptographic object that is specified must also be a Shared Secret Key.
- If the original message was signed with a Private Key, the cryptographic object that is specified must be a Crypto Certificate (public certificate).
- The cryptographic material can be provided through a JSON Web Key (JWK).
- If a JWK header parameter is included in the header of the JWT, the parameter must match the JWK or cryptographic object that is specified in the policy, or the JWT validation will fail.
- If both a cryptographic object and a JWK are specified, the cryptographic object is used to decrypt or verify the JWT.
- The JWT validate action on the DataPower API Gateway can verify a JWT by using either a single JWK, or a JWK set.

## Prerequisites

The following prerequisites apply:

- IBM® DataPower V7.5 with the Application Optimization (AO) option.
- If you are using one or more cryptographic objects, they must be located in the IBM API Connect domain on the DataPower appliance. The cryptographic objects must reference the Shared Secret Key or public certificate that is needed to decrypt the JWT contents or verify the signature.
- If a JSON Web Key (JWK), or a JWK set, is being used, it must be referenced by a runtime variable.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Validate JWT policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>jwt-validate</code> .	string
Description	No	A description of the policy.	string

Property label	Required	Description	Data type
JSON Web Token (JWT)	Yes	Context or runtime variable that contains the JWT to be validated. The default value is: <code>request.headers.authorization</code> . However, if this property is not set, the policy looks for the JWT in the <code>request.headers.authorization</code> location by default.  Note: The format of the authorization header must be:  <code>"Authorization: Bearer jwt-token"</code>  where <code>jwt-token</code> is the encoded JWT.	string
Output Claims	Yes	Runtime variable to which the full set of claims that are contained in the JWT is assigned. The default value is: <code>decoded.claims</code> .	string
Issuer Claim	No	The Perl Compatible Regular Expressions (PCRE) to use to validate the Issuer (iss) claim.	string
Audience Claim	No	The PCRE to use to validate the Audience (aud) claim.	string
Decrypt Crypto Object	No	The cryptographic object (a shared key or certificate) to use to decode the claim. <sup>1</sup>	string
Decrypt Crypto JWK variable name	No	Runtime variable that contains the JWK to use to decrypt the JWT. <sup>1</sup>	string
Verify Crypto Object	No	The cryptographic object (a shared key or certificate) to use to verify the signature. <sup>2</sup>	string
Verify Crypto JWK variable name	No	Runtime variable that contains the JWK, or JWK set, to use to verify the signature. <sup>2</sup>	string

## Example

```
- jwt-validate:
  version: 1.0.0
  title: jwt-validate
  jwt: request.headers.authorization
  output-claims: decoded.claims
  iss-claim: "'^data.*'"
  aud-claim: "'^id.*'"
  jwe-crypto: jweCryptoObjectName
  jwe-jwk: jwe.jwk
  jws-crypto: jwsCryptoObjectName
  jws-jwk: jws.jwk
```

## Errors

The following error can be thrown while the policy is being executed:

- **RuntimeError** - a generic error that captures all errors that occur during the execution of the policy. Upon a validation failure, the detailed error message that is received from the underlying JOSE module is written to the default system log as an error message. This detailed error message is also assigned to the runtime variable `jwt-validate.error-message`, so it can be retrieved via `catch`.

If a `catch` is not configured, in the case of a validation failure the Validate JWT policy returns an HTTP code 500 `Invalid-JWT-Validate` failure. The detailed error message from the underlying JOSE module can be found in the system log.

Attention: If you are an API developer who is troubleshooting a failure that one of your customers had with your API, consider the security risks before sending a customer the exact content of the log message. You can avoid the possibility of someone launching an attack based on the information that they receive from the log message by sending the customer only general information about the message.

For examples, see [jwt-validate](#).

## Related information

- [Introduction to JSON Web Tokens](#)

<sup>1</sup> A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to decrypt the JWT. However, if both data types are specified, only the Crypto Object is used.

<sup>2</sup> A JWK, or JWK set, and a Crypto Object are both valid ways of providing the cryptographic data necessary to verify the JWT. However, if both data types are specified, only the Crypto Object is used.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Invoke

Apply the Invoke policy to call another service from within your assembly. The response from the backend is stored either in the variable `message.body` or in the response object variable if it is defined. The policy can be used with JSON or XML data, and can be applied multiple times within your assembly.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [invoke](#).

## About

You can attach this policy to the following API flows:

- REST
- SOAP

Users might notice that the last invoke in their policy is replaced by a proxy. The replacement is sometimes done automatically by the IBM® API Connect DataPower Gateway to improve performance. The proxy is functionally equivalent to the invoke, but the API caller might notice the following differences when proxy is used.

- If the HTTP request made by the invoke or proxy gets a redirect (3xx) response:
  - invoke returns the response from following the redirect response.
  - proxy does not follow 3xx responses, and the redirect response is returned.
- The IBM API Connect test tool shows that proxy was used, but invoke appears in the Analytics latency records.
- The response from the proxy can contain different whitespace or escaping than a response from invoke. Despite the differences in the response, it is still valid.

If you want to prevent replacement of the last invoke in the assembly with proxy, you can set the API property `api.properties.x-ibm-gateway-optimize-invoke` to `false`. For more information, see [API properties](#)

Note:

- The Invoke policy does not support responses with multipart form data, that is, when the response is set to `Content-Type: multipart/related`.
- The Invoke policy always uses chunked encoding, which is not supported by the HTTP 1.0 protocol.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Invoke policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>invoke</code> .	string
Description	No	A description of the policy.	string
URL	Yes	Specifies a URL for the target service. For a SOAP API, a URL is added by default. Where possible, the Invoke URL value is pre-supplied from information that is defined in the imported WSDL.	string
TLS profile	No	Specifies a TLS profile to use for the secure transmission of data.	string
Timeout	Yes	The time to wait before a reply back from the endpoint (in seconds). The default value is <code>60</code> .	integer
Username	No	The username to use for HTTP Basic authentication.	string
Password	No	The password to use for HTTP Basic authentication.	string
HTTP Method	Yes	The HTTP method to use for the Invoke. Valid values are: <ul style="list-style-type: none"> <li>• Keep</li> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• PATCH</li> <li>• HEAD</li> <li>• OPTIONS</li> </ul> The default value is <code>GET</code> . However, if set to <code>Keep</code> , or the property is removed from the source, the HTTP method from the incoming request is used.	string
Compression	No	Select this check box to enable Content-Encoding compression on upload. The check box is cleared by default.	boolean

Property label	Required	Description	Data type
Cache Type	No	<p>The cache type determines whether to cache documents, honoring or overriding the HTTP Cache Control directives received in the response from the target URL. This property takes effect only when a response is received, otherwise the policy always returns the non-expired response that was previously saved in cache. Valid values are:</p> <p><b>Protocol</b> The cache behavior is determined by the Cache-Control headers on the response, in accordance with RFC 7234. To optimize performance, if the gateway receives more than one request for a resource that is not in the cache but could be cached when the response from the target URL is received, the gateway sends only one request to the target URL; the remaining requests are not processed until the response from the first request has been received and the cache behavior has been determined from this response. If the response indicates that caching is possible, the gateway responds to all waiting requests with the cached resource. If the response indicates that caching is not possible, the gateway sends all waiting requests to the target URL.</p> <p>Use this option only if you expect that responses from the target URL can be cached, in which case it should improve performance and limit the demand on the target URL. If, however, the target URL never indicates that the gateway should cache its response, performance might be impaired when compared to the No Cache option.</p> <p><b>No Cache</b> Responses from the target URL are not cached on the gateway regardless of any caching headers returned. In this case, every request from the client is sent to the target URL. Use this option if you do not want to cache any of the backend responses on the gateway, or if it is unlikely that a response from the target URL will allow caching through the Cache-Control header settings.</p> <p><b>Time to Live</b> This option is similar to the Protocol option except it allows you to specify the amount of time that you want the successful response from the invoke or proxy to remain in the cache. Use this option only if you expect that responses from the target URL can be cached.</p> <p>The default value is Protocol.</p>	string
Time to Live	No	Specifies the amount of time in seconds that the response stays in the cache. Applies only if the property Cache type is set to <b>Time to Live</b> . Enter a value in the range 5 - 31708800. The default value is 900.	integer
Cache key	No	Specifies the unique identifier of the document cache entry. If omitted, the entire URL string is used as the key.	string
<input type="checkbox"/> API Gateway only Inject proxy headers	No	If you select this check box, the <b>invoke</b> policy injects the <b>X-Forwarded-For</b> , <b>X-Forwarded-To</b> , <b>X-Forwarded-Host</b> , and <b>X-Forwarded-Proto</b> headers to the request that is sent to the target URL. The check box is cleared by default.	boolean
<input type="checkbox"/> API Gateway only Decode Request Params	No	If you select this check box, any request parameters that are referenced by a variable definition on the target URL of the invoke policy are URL-decoded. The check box is cleared by default.	boolean
<input type="checkbox"/> API Gateway only Queryparam Encode	No	If you select this check box, all "+" characters in the query parameter values of the target URL are encoded to %2F. The check box is cleared by default.	boolean
<input type="checkbox"/> API Gateway only Keep Payload	No	If you select this check box, the invoke policy sends a payload on an <b>HTTP DELETE</b> method. The check box is cleared by default.	boolean
<input type="checkbox"/> API Gateway only Restrict to HTTP 1.0 (policy version 2.0.0 only)	No	If you select this check box, HTTP transactions are restricted to version 1.0. The check box is cleared by default.	boolean
<input type="checkbox"/> API Gateway only Allow chunked uploads	No	<p>If you select this check box, chunked-encoded documents are sent to the server. When the HTTP 1.1 protocol is used, the body of the document can be delimited by either <b>Content-Length</b> or chunked encoding. While all servers can interpret <b>Content-Length</b>, many applications fail to understand chunked-encoded documents. For this reason, <b>Content-Length</b> is the standard method.</p> <p>The use of <b>Content-Length</b> interferes with the ability of the DataPower Gateway to fully stream. If you must stream full documents to the target server, enable this property.</p> <p>When enabled, the server must be RFC 2616 compatible. Unlike all other HTTP 1.1 features that can be negotiated down at run time, you must know beforehand that the target server is RFC 2616 compatible.</p> <p>The check box is selected by default.</p> <p>Note: Chunked encoding is not supported by the HTTP 1.0 protocol.</p>	boolean

Property label	Required	Description	Data type
<div style="border: 1px solid green; padding: 2px;">API Gateway only</div> Header control	No	<p>Specifies the headers in <code>message.headers</code> that you want to copy to the target URL.</p> <p>To prevent headers from being copied, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Select Blacklist.</li> <li>2. Click Add blacklist.</li> <li>3. In the empty field that is displayed, enter the header name.</li> <li>4. To add further headers, repeat the previous steps.</li> </ol> <p>To specify headers that you want to be copied, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Select Whitelist.</li> <li>2. Click Add whitelist.</li> <li>3. In the empty field that is displayed, enter the header name.</li> <li>4. To add further headers, repeat the previous steps.</li> </ol> <p>The values that you specify are in regular expression format. For example, to specify the Content-Type header, enter <code>^Content-Type\$</code></p> <p>By default, Blacklist is selected, with no blacklist entries, meaning that <b>all</b> headers are copied.</p>	string
<div style="border: 1px solid green; padding: 2px;">API Gateway only</div> Parameter control	No	<p>Specifies the parameters in the incoming request that you want to be copied to the target URL.</p> <p>To prevent parameters from being copied, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Select Blacklist.</li> <li>2. Click Add blacklist.</li> <li>3. In the empty field that is displayed, enter the parameter name.</li> <li>4. To add further parameters, repeat the previous steps.</li> </ol> <p>To specify parameters that you want to be copied, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Select Whitelist.</li> <li>2. Click Add whitelist.</li> <li>3. In the empty field that is displayed, enter the parameter name.</li> <li>4. To add further parameters, repeat the previous steps.</li> </ol> <p>The values that you specify are in regular expression format.</p> <p>For example, if the incoming request is</p> <pre>http://apigw/org/sandbox/petstore/base?petid=100&amp;display=detailed</pre> <p>and you specify a white list entry of <code>^petid\$</code>, the target URL at run time will be</p> <pre>http://myhost/mypath?storeid=3&amp;petid=100</pre> <p>By default, Whitelist is selected, with no whitelist entries, meaning that <b>no</b> parameters are copied.</p>	string
Stop on error	No	<p>Select the errors that, if thrown during the policy execution, cause the assembly flow to stop. If there is a <b>catch</b> flow configured for the error, it is triggered to handle the error thrown. If an error is thrown and there are no errors selected for the Stop on error setting, or if the error thrown is not one of the selected errors, the policy execution is allowed to complete, and the assembly flow continues.</p>	string
Response object variable	No	<p>The name of a variable that will be used to store the response data from the request. By default, the invoke response, that is the body, headers, statusCode and statusMessage, is saved in the variable <code>message</code>. Use this property to specify an alternate location to store the invoke response. This variable can then be referenced in other actions, such as <a href="#">Map</a>.</p> <p>Note: If you want the response to be saved in <code>message</code>, leave the Response object variable property blank, do <b>not</b> supply the value <code>message</code>.</p>	string

## Example

```
- invoke:
  version: 2.0.0
  title: get the account status
  target-url: https://example.com/accounts/{id}?status={status}
  cache-response: time-to-live
  cache-putpost-response: true
  tls-profile: MyTLSProfile
  verb: POST
  timeout: 60
  compression: false
  username: MyUser
  password: MyPassword
  stop-on-error:
    - ConnectionError
    - OperationError
```

## Related tasks

- [Creating a new REST OpenAPI definition](#)
- [Handling errors in the assembly](#)

## Related information

- [TLS profiles](#)



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## JSON to XML

Use the JSON to XML policy to convert the context payload of your API from the JavaScript Object Notation (JSON) format to the extensible markup language (XML) format.

### Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [json-to-xml](#).

### About

The JSON to XML policy uses a simple convention, based on BadgerFish, to convert your API context payload from JSON to XML. The policy expects the JSON input to be in the same format as the BadgerFish convention, so the structure can be rebuilt in XML. No additional configuration is required. For more information about the BadgerFish convention, see [BadgerFish](#).

Note: The JSON to XML policy does convert the JSON structure { "a" : "hello" } (which is not BadgerFish convention) into `<a>hello</a>`.

You can attach this policy to the following API flows:

- REST
- SOAP

Use the IBM® API Connect API Manager assembly view when you are creating your API definition to add a built-in policy to the flow.

The policy must be attached to the flow at the point at which you require the conversion to be performed. For example, if you need to convert a JSON-formatted request into an XML-formatted request, the policy must be attached to the request flow.




The policy reads input from the `message.body`, if that context exists, otherwise from the `request.body`, and then writes the output to the `message.body`.

Note: If you are using the DataPower API Gateway, the input to the JSON to XML policy must be parsed data. One way to produce parsed data is to use a [Parse](#) policy before a JSON to XML policy in your assembly flow, which provides explicit control of the parse action.

### Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>json-to-xml</code> .	string
Description	No	A description of the policy.	string
 Input	No	The input message to convert. Specify the name of a variable in the API context. <code>variableName.body</code> , the message payload, represents the JSON input to convert. The default value of the variable is <code>message</code> and <code>message.body</code> is the default input.	string
 Output	No	The output message to store the conversion result. Specify the name of a variable in the API context. <code>variableName.body</code> represents the result of conversion from JSON format to XML format. When the specified input message is the default message, the default output is <code>message.body</code> . Otherwise, when the input message is the variable <code>my-message-variable</code> , for example, the default output is <code>my-message-variable.body</code> . The variable cannot be any read-only in the API context.	string
 Conversion type	No	The conversion type that determines the target format of the output. The following options are available: <ul style="list-style-type: none"> <li>• None: No conversion of the output takes place.</li> <li>• badgerFish: BadgerFish convention is used to determine the target conversion format of the output.</li> </ul>	string
Root XML Element Name	Yes	The root element name of the resultant XML document. This property is used only if the input JSON document is not hierarchical and has more than one top-level property, or if the Always output the root element check box is selected. The default value is <code>json</code> .	string
Always output the root element	Yes	Select this check box if you always want the policy to output the root element, even if it is not required to make the XML document well formed. The default value is <code>false</code> .	boolean
Element name for JSON array elements	No	The XML element name to be used for JSON array elements.	string

## Examples

For example, the following simple JSON object

```
{ "a": { "$" : "hello" } }
```

becomes

```
<a>hello</a>
```

The following JSON object with an attribute

```
{ "a": { "$" : "hello", "@type" : "world" } }
```

becomes

```
<a type="world">hello</a>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Log

Use the Log policy to customize or override the default activity logging configuration for an API.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [log](#).

## About

By default, the logging of activity when an API is called, and the sending of the log data to the analytics server, is determined by the settings in the API definition, as described in [Activity logging with the DataPower API Gateway](#). However, you can customize activity logging by adding a Log policy to the API assembly flow.

The Log policy enables you to gather all API analytics data for processing in your assembly, and to control what activity log data is sent to the analytics server. Here are some examples of the ways in which you can use the Log policy to customize API activity logging:

- Gather activity log data, use a Redaction policy to redact sensitive fields in the data, then send to the analytics server.
- Send different levels of analytics data depending on certain conditions; for example, the API operation that was called.
- Configure custom activity logging in a global policy that is applied to every API in a Catalog. For more information on global policies, see [Working with global policies](#).

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Log policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>log</b> .	string
Description	No	A description of the policy.	string
Mode	Yes	Select one of the following options: <ul style="list-style-type: none"><li>• Gather-only: gather all analytics data and write it to the <b>log</b> context variable, which populates the API event record on completion of the API execution. For more information on the fields in the <b>log</b> context variable, and the consequent API event record, see <a href="#">API event record fields</a>.</li><li>• Send-only: perform the following actions:<ul style="list-style-type: none"><li>◦ Read the data from the <b>log</b> context variable.</li><li>◦ Truncate all message payloads and convert to a textual representation.</li><li>◦ Send the data to the analytics server.</li></ul></li><li>• Gather-and-send: perform a <b>gather-only</b> operation, immediately followed by a <b>send-only</b> operation.</li></ul> <p>If you use the Send-only or Gather-and-send option, data is buffered and sent to the analytics server in batches according to the time interval configured for the Analytics Endpoint on the DataPower API Gateway. For more information, see <a href="#">Configuring an analytics endpoint</a> in the DataPower knowledge center.</p> <p>Note: If you are offloading to a third party analytics server, you can redact any aspect of the event data. If you are using API Connect analytics, you can redact only request and response payloads.</p>	string

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Map

Use the Map policy to apply transformations to your assembly flow and specify relationships between variables.

### Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [map](#).

### About

Be aware that MAP decodes all parameter values to an ASCII character equivalent. If a MAP parameter contains entries such as %nn, the MAP output contains decoded values.

For information about the structure of a Map policy and its behavior, see [The Map policy structure](#).

For information about configuring a Map policy by using the user interface, see [Configuring the Map policy in the user interface](#).

For examples of YAML representations of different Map policy configurations, see [Map policy examples](#).

You can attach this policy to the following API flows:

- REST
- SOAP

### Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Map policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The field is visible when editing inputs. The default value is <b>map</b> .	string
Description	No	A description of the policy. The field is visible when editing inputs.	string
Inputs	Yes	A list of variables that are inputs of the policy.	array (string)
Outputs	Yes	A list of variables that are outputs of the policy.	array (string)
Value	Yes	A GatewayScript program to be performed by the policy in order to map its inputs to its outputs, or to set the value of outputs.	string

Note: The map policy has other properties that are not displayed in the user interface. For a complete list of properties, see [map](#).

### Related tasks

- [Creating a new REST OpenAPI definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## The Map policy structure

The Map policy uses a structure within its OpenAPI definition to specify the behavior of the policy.

This topic contains the following sections:

- [Structure](#)
- [Input and output definitions](#)
- [Actions](#)
- [Script](#)
- [Fields](#)

- [References to inputs and outputs](#)
- [Accessing other contexts](#)

For map policy examples, see [Map policy examples](#).

Note: If you deploy your API to the DataPower® Gateway (v5 compatible) then, with the exception of client ID and client secret, the passing of form input as a parameter into an API is not supported. This restriction does not apply if you deploy your API to the DataPower API Gateway.

## Structure

In addition to its title and description, a Map policy has the following four main sections:

### inputs

A list of variables that form the input of the Map policy. Each input has a context variable in which the input variable is found, the variable's name within the Map policy, the content type of the variable, and the definition of the variable or a schema defining its structure.

### outputs

A list of variables that form the output of the Map policy. These include a context variable where the output variable is found or should be created, the variable's name at output, and the definition of the variable or a schema defining its structure.

### action

An array containing details of the actions to be performed in order. Each entry includes either a **set** or **create** field, which specifies the output variable or variables that are part of the action. An action can also contain a **from** field, which specifies the input variable or variables that are part of the action.

Each action also contains either a **value** field, when the output is set or created, or a **foreach** field and a new **actions** section, when further actions are nested within the first to set or create elements in a nested array.

### options

The following property options are available for you to select:

- Include empty. If the check box is selected (the default option), empty XML elements are included in the output of the map policy. Clear the check box if you do not want empty XML elements to be included in the output of the map policy.
- Namespace inheritance. If the check box is selected (the default option), XML namespaces are inherited from the parent element. Clear the check box if you want the map policy to use explicit namespaces.
- Namespace inlining. If the check box is selected (the default option), XML namespaces will be inserted into the document where they are first used. Clear the check box if you want namespaces to all be defined on the root element.
- API Gateway only Resolve XML input data type. If the check box is selected, XML elements whose schema is configured as type boolean or numeric will be converted to that data type. Clear the check box (the default option) if you want all XML element values to be returned as a string.
- API Gateway only Empty XML element handling. This property controls how the map policy handles the output of an empty XML element. The following choices are available:
  - String (the default option): An empty XML element is handled as an empty string.
  - Null: An empty XML element is handled as a null value.
  - None: The data is ignored.
  - API Gateway only String Badgerfish: The value for an empty XML element is considered to be an empty string. The empty string value will be placed into a JSON badgerfish value property.
  - API Gateway only Null Badgerfish: The value for an empty XML element is considered to be null. The null value will be placed into a JSON badgerfish value property. A mapping of this element to a JSON output property does not occur unless the Allow null value option is selected.
- API Gateway only Use only first array element. If the check box is selected, if an array is encountered in the traversal of the input, only the first element is used. Clear the check box (the default option) if you want the map policy to use all array elements.
- API Gateway only Resolve API Connect variable references. If the check box is selected (the default option), API Connect variable references found in the map properties are resolved. Clear the check box if you want the map policy to ignore API Connect variable references in the map properties.
- API Gateway only Allow null value. If the check box is selected, an input property value with a null value is mapped to the output document. Clear this check box (the default option) if you want the map policy to ignore null input values.
- API Gateway only Optimize schema definition. If the check box is selected, complex schema types evaluation handles circular type references in an optimized manner. Clear this check box (the default option) to evaluate these schema types in a standard manner.
- API Gateway only Create required child properties for array objects and mapped optional objects. If the check box is selected, default values are generated in the output for required properties that are either not mapped, or for which there is no input data present, in the following specific cases:
  - An array consists of objects that contain one or more required properties.
  - An object which is optional has one or more child properties that are required.
 By default, these required properties are not present in the output. If you select this option, these required properties will be present in the output. If the output schema defines a **default** property for the output property then the specified default value is used, otherwise a default value is assigned dependent on the data type, as follows:
  - String: empty string ("")
  - Number: 0
  - Boolean: false
  - Object: empty object
  - Array: empty array

#### Example 1

The input data has the following array of objects:

```
[{"a": "value1"}, {"a": "value2", "b": "value3"}]
```

The output schema defines the output object as having two properties, **a** and **b**, of which **b** is required. The map policy defines the following mappings:

- `input.array.a` to `output.array.a`
- `input.array.b` to `output.array.b`

If the check box is selected, and **b** is either not mapped or has no input data present, then **b** is assigned a default value of an empty string, and the output is as follows:

```
[{"a": "value1", "b": ""}, {"a": "value2", "b": "value3"}]
```

#### Example 2

The output schema defines the following structure:

```
{"a" : {"b" : {"c" : "value1", "d" : "value2"} } }
```

Property **b** is optional but property **d** within **b** is required.

The map policy defines a mapping to `output.a.b.c`.

If the check box is selected, and **d** is not mapped, then **d** is assigned a default value of an empty string, and the output is as follows:

```
{"a" : {"b" : {"c" : "value1", "d" : ""} } }
```

If the check box is not selected, these required properties are **not** created in the output with their default values.

- API Gateway only** Empty JSON array handling. This property controls how the map policy handles the output of an empty array. The following choices are available:
  - All (The default value): Output all empty arrays, including empty children arrays.
  - Parent: Output only the current property's empty array value. Children map actions of this property are not attempted.
  - None: Prevent any empty output array values from being produced.
- API Gateway only** Enable JSON post processing. Select this check box to enable post processing of mapped JSON output. The post processing of JSON output will use the output schema to ensure that property values are of the same data type as that defined in the schema. It will also normalize output property values that have a Badgerfish JSON syntax due to object mapping of an XML input. The check box is cleared by default, meaning that there is no post processing of mapped JSON output.
- API Gateway only** Schema definition circular reference limit. Set the value of this property to an integer value that specifies the maximum allowed number of iterations of a circular schema definition. The default value is 1, which means that circular schema definitions are not followed. The maximum possible value is 5.
- Severity level for input data log messages; This property specifies the severity level for log messages that relate to input data. The following choices are available:
  - error
  - warn
  - info
- API Gateway only** Create empty parent object for failed mapping. If a mapping fails because its input is not present and there is no default mapping configured, the default behavior is to not to make any change to the output mapping. Select this check box to create an empty object for the parent of the target mapping, emulating the behavior of IBM® API Management Version 4.0.

#### Example

The map policy defines a mapping to `output.a.b.c`.

If input data is present, the output is as follows:

```
{
  "a": {
    "b": {
      "c": "inputvalue"
    }
  }
}
```

If there is no input data, and the Create empty parent object for failed mapping option is selected, the output is as follows:

```
{
  "a": {
    "b": {
    }
  }
}
```

Properties **a** and **b** are created but the value of **b** is an empty object.

The check box is cleared by default.

## Input and output definitions

You define the inputs and outputs of your Map policy in their own sections. Each input or output is an element in the array inputs or outputs and is defined by a name, a schema definition or reference, and a variable within the context from which it should be read or to which it should be written. After they have been defined, inputs and outputs are referenced by the name provided in the definition, not by the name of the variable.

The following example shows the inputs and outputs sections of a Map policy.

```
inputs:
  input_string:
    schema:
      type: string
      variable: request.parameters.name_in
  input_integer:
    schema:
      type: integer
      variable: request.parameters.age_in
outputs:
  output:
    schema:
```

```
$ref: '#/definitions/output'
variable: message.body
```

The schema field specifies the schema that describes the variable and can be a simple type, a reference to a definition, or an inline schema definition.

The variable field describes the variable and the context that should be assigned to the input or output variable during the execution of the map policy.

## Actions

The fields included in the **actions** section are used in the following ways:

### set

Use the **set** field when you want to assign the result of the **value** field to the output variable specified in the **set** field, replacing the existing value of the output variable. You can specify only one output variable, although this variable can be an array or object.

### create

Use the **create** field when you want to use the result of the **value** field to create a new entry for the output array specified in the **create** field, appending it to the array. You can specify only one output variable, although this variable can be an array or object.

### from

Specify which variables are used in the action as either a single variable or an array of variables, where a variable can be an array or an object. The **from** field is not included if no inputs are used.

### value

Use GatewayScript to provide a script that produces output variables. When a single input is mapped to a single output, the **value** field can be omitted and the variable in **from** is set or created as the variable in **set** or **create** respectively.

### default

Provide a static value, or an inline variable reference, to be applied to the output when no input value is provided. For information on inline variable references, see [Inline references](#).

### foreach

Specify a variable if you want to execute the associated **actions** field for each entry of the array. The variable can be from the input or output of the Map policy.

### actions

Use the **actions** field to nest actions within an action. Because another action could achieve the same result if applied only once, it is primarily for use with the **foreach** field.

## Script

In a **value** or **default** field, use GatewayScript to write the behavior of the action to which the **value** field belongs.

Include the script in single quotation marks. For example: `'4 + 5'` or `'variable_1.toUpperCase()'`.

For information about GatewayScript, see [Gateway programming model and GatewayScript](#)

## Fields

### from, set, and create

Each action must have a single **set** or **create** field that specifies the output variable to which the action is applied. Each action can also have a **from** field containing one or more entries that are used to specify the input variable or variables that are used in the action.

The **set** and **create** fields are both used to assign values to the output variable.

- **set** replaces the current value of the output variable or creates the variable if it does not already exist.
- **create** appends a new array entry to the output variable.

For **set** and **create**, use **output\_variable\_name.variable\_name** to specify which of your defined output variables to use, where **output\_variable\_name** is as defined in the outputs section of the Map policy and **variable\_name** refers to an optional field that belongs to the output variable.

For **from**, use **input\_variable\_name.variable\_name** to specify which of your defined output variables to use, where **input\_variable\_name** is as defined in the outputs section of the Map policy and **variable\_name** refers to an optional field that belongs to the output variable.

### foreach

Use the **foreach** field to specify an input array for which the following **actions** or **value** field will be executed for each entry of the array.

For example:

```
foreach: input.in
actions:
  actions
```

where **input.in** is an input variable that is an array and **actions** is one or more actions in the same format as the parent section. In this example the instructions specified in **actions** are executed once for each array entry of **input.in**.

Referencing the input variable specified in the **foreach** field references the array entry that the current iteration corresponds to.

If a single variable is used in the **foreach** field instead of an array, the following **actions** or **value** field will be applied to or based upon the single variable once and then the loop will terminate.

## References to inputs and outputs

Reference variables from the **from** field either by name or by using a number preceded by a '\$' and enclosed in parentheses. The variables are numbered from 1, where 1 is the first variable in the array, or the only variable when the **from** field lists only a single variable. For example:

```
value: '$(1) + $(2)'
```

or

```
value: '$(variable_1) + $(variable_2)'
```

where each *variable* is a variable that is included in the **from** field.

During a **foreach** loop, you can reference **\$(0)**. The **\$(0)** variable begins a **foreach** loop empty but, after an iteration, becomes equivalent to the output of the iteration and can then be referenced again. In this manner, you can apply an array to a single value. For example:

```
- set: out.total
from: in.input
foreach: in.input
value: '$(0) + $(in.input)'
```

where *out.total* is referenced by **\$(0)**. In each iteration, the current value of *out.total* and the current array entry of *in.input* are summed, and the value of *out.total* is set as this summation.

When using a **foreach** to operate on an array, if the elements of the array do not have named fields, you can use **\$(this)** to reference the current level of nesting.

## Accessing other contexts

At any point within your Map policy's **value** or **default** fields, you can access the context of the API call using the syntax **\$(context.variable)**.

Alternatively, you can include the variables from other contexts when you define an input to your map policy and then reference it as you would any other input variable.

For a list of available context variables, see [API Connect context variables](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring the Map policy in the user interface

The assemble view in API Manager provides a visual representation of the relations between the inputs and outputs of your Map policy.

### Procedure

To configure your Map policy, complete the following steps:

1. Click the Map policy in the canvas of the Assemble view.  
The property sheet opens.

2. Click the Edit inputs icon  in the Input column.

3. Add an input variable.

- a. Click + input.

- b. In the Context variable field for an input, provide the location of your input variable in the context of the assembly. For a list of context variables, see [API Connect context variables](#).

- c. In the Name field for an input, provide a name for your input for use within only the Map policy.

Note:

You must ensure that the name that you provide does not exactly match the value of the Context variable field, or the result might be unpredictable.


- d. Optional: In the Content type field, specify the type of your input. If None is selected, the content type is treated as JSON.

- e. In the Definition field for an input, provide the type of the variable.


The type can be one from a standard set of types, a definition that you have created for your API, or you can select Inline schema to provide a schema as one of the following options:

- YAML
- JSON
- Generate from sample JSON
- Generate from sample XML

If you select Object or Array, you can create a schema through the user interface after you have clicked Done and returned to the main view of the property sheet.

4. Optional: To remove a variable, click the corresponding Remove input icon .

5. After you have added all of your input variables, click Done.

6. Click the Edit outputs icon  in the Output column.

7. Add an output variable.

- a. Click + output.

- b. In the Context variable field for an output, provide the location of your output. This location can be a new context or one already established during the assembly. For a list of context variables, see [API Connect context variables](#).

- c. In the Name field for an output, provide a name for your variable to use within the Map policy and when it is included in its context at output.

Note:

You must ensure that the name that you provide does not exactly match the value of the Context variable field, or the result might be unpredictable.




- d. Optional: In the Content type field, specify the type of your input. If None is selected, the content type is treated as JSON.

- e. In the Definition field for an output, provide the type of the variable.


The type can be one from a standard set of types, a definition that you have created for your API, or you can select Inline schema to provide a schema as one of the following options:

- YAML
- JSON
- Generate from sample JSON
- Generate from sample XML

If you select Object or Array, you can create a schema through the user interface after you have clicked Done and returned to the main view of the property sheet.

8. Optional: To remove a variable, click the corresponding Remove output icon .
  9. After you have added all of your output variables, click Done.
  10. Optional: If you selected Object or Array for the type of an input or output, create an inline schema definition through the user interface by completing the following steps:
    - a. For an Array, click add item. Provide a type for the item and then click the Add icon .
    - b. For an Object, click add property. Provide a name and type for the property and then click the Add icon .
- For objects and arrays created in this manner, you can continue to add items and properties, which can themselves be objects and arrays.
11. To connect an input variable to an output variable, click the circle that is directly on the right of the input variable and then click the circle that is directly on the left of the output variable.

A green line is drawn, linking the two variables together. You can connect multiple inputs to a single output, and a single input can be connected to multiple outputs.
  12. To configure an output, whether it has inputs connected to it or not, click the circle directly to the left of the output variable without first clicking on a circle for an input variable.

The Configure mapping window opens.
  13. Optional: In the Mapped from section of the window, you can view which inputs are mapped to the output you are editing. To remove an input, click the Remove input icon  beside the input.

If the output is part of an array, further configuration options are available. The array, or levels of array in the case of a multidimensional array, can be created by iterating over arrays on the input side of the mapping. For each level of your array, select which array on the input side is to be iterated over. In the Value field, you can use `$(this)` to reference elements of an array that are not named within the array.
  14. Optional: In the Value field, use GatewayScript to configure how any inputs are transformed to produce the output.

For more information about valid code, see the [Script](#) section of the [The Map policy structure](#) topic.
  15. Optional: In the Default field, provide a static value, or an inline variable reference, to be applied to the output when no input value is provided. For information on inline variable references, see [Inline references](#).
  16. Optional: To delete all mappings to the output, click Delete.
  17. When you have configured your outputs, click OK.
  18. Optional: Click the Settings icon in the Map column.
    - a. Optional: Provide a Title and Description for your Map policy.
    - b. To control the XML output of the map policy, select the following options as required:

#### Include empty

If the check box is selected (the default option), empty XML elements are included in the output of the map policy. Clear the check box if you do not want empty XML elements to be included in the output of the map policy.

#### Namespace inheritance

If the check box is selected (the default option), XML namespaces are inherited from the parent element. Clear the check box if you want the map policy to use explicit namespaces.

#### Namespace inlining

If the check box is selected (the default option), XML namespaces will be inserted into the document where they are first used. Clear the check box if you want namespaces to all be defined on the root element.

#### API Gateway only

#### Resolve XML input data type

If the check box is selected, XML elements whose schema is configured as type boolean or numeric will be converted to that data type. Clear the check box (the default option) if you want all XML element values to be returned as a string.

Note: These options effect the XML output only and have no effect on the JSON data.

#### API Gateway only

- c. From the Empty XML element handling list, select one of the following options to control how the map policy handles the output of an empty XML element:
  - String (the default option): An empty XML element is handled as an empty string.
  - Null: An empty XML element is handled as a null value.
  - None: The data is ignored.
  - **API Gateway only** String Badgerfish: The value for an empty XML element is considered to be an empty string. The empty string value will be placed into a JSON badgerfish value property.
  - **API Gateway only** Null Badgerfish: The value for an empty XML element is considered to be null. The null value will be placed into a JSON badgerfish value property. A mapping of this element to a JSON output property does not occur unless the Allow null value option is selected.
- d. **API Gateway only** Select the following general configuration options as required:

#### Use only first array element

If the check box is selected, then if an array is encountered in the traversal of the input, only the first element is used. Clear the check box (the default option) if you want the map policy to use all array elements.

#### Resolve API Connect variable references

If the check box is selected (the default option), API Connect variable references found in the map properties are resolved. Clear the check box if you want the map policy to ignore API Connect variable references in the map policies.

#### Allow null value

If the check box is selected (the default option), an input property value with a null value is mapped to the output document. Clear this check box (the default option) if you want the map policy to ignore null input values.

#### Optimize schema definition

If the check box is selected, complex schema types evaluation handles circular type references in an optimized manner. Clear this check box (the default option) to evaluate these schema types in a standard manner.

#### API Gateway only

#### Enable JSON post processing

If the check box is selected, post processing of mapped JSON output is enabled. The post processing of JSON output will use the output schema to ensure that property values are of the same data type as that defined in the schema. It will also normalize output property values that have a Badgerfish JSON syntax due to object mapping of an XML input. The check box is cleared by default, meaning that there is no post processing of mapped JSON output.



API Gateway only

Create empty parent object for failed mapping

API Gateway only

Use this setting to control the behavior if a mapping fails because its input is not present and there is no default mapping configured.

The default behavior is to make no change to the output, but if you select this check box an empty object is created for the parent of the target mapping, emulating the behavior of IBM API Management Version 4.0.

#### Example

The map policy defines a mapping to `output.a.b.c`.

If input data is present, the output is as follows:

```
{
  "a": {
    "b": {
      "c": "inputvalue"
    }
  }
}
```

If there is no input data, and the Create empty parent object for failed mapping option is selected, the output is as follows:

```
{
  "a": {
    "b": {
    }
  }
}
```

Properties `a` and `b` are created but the value of `b` is an empty object.

The check box is cleared by default.

#### Create required child properties for array objects and mapped optional objects

If the check box is selected, default values are generated in the output for required properties that are either not mapped, or for which there is no input data present, in the following specific cases:

- An array consists of objects that contain one or more required properties.
- An object which is optional has one or more child properties that are required.

By default, these required properties are not present in the output. If you select this option, these required properties will be present in the output. If the output schema defines a `default` property for the output property then the specified default value is used, otherwise a default value is assigned dependent on the data type, as follows:

- String: empty string ("")
- Number: 0
- Boolean: false
- Object: empty object
- Array: empty array

#### Example 1

The input data has the following array of objects:

```
[{"a": "value1"}, {"a": "value2", "b": "value3"}]
```

The output schema defines the output object as having two properties, `a` and `b`, of which `b` is required. The map policy defines the following mappings:

- `input.array.a` to `output.array.a`
- `input.array.b` to `output.array.b`

If the check box is selected, and `b` is either not mapped or has no input data present, then `b` is assigned a default value of an empty string, and the output is as follows:

```
[{"a": "value1", "b": ""}, {"a": "value2", "b": "value3"}]
```

#### Example 2

The output schema defines the following structure:

```
{"a" : {"b" : {"c" : "value1", "d" : "value2"} } }
```

Property `b` is optional but property `d` within `b` is required.

The map policy defines a mapping to `output.a.b.c`.

If the check box is selected, and `d` is not mapped, then `d` is assigned a default value of an empty string, and the output is as follows:

```
{"a" : {"b" : {"c" : "value1", "d" : ""} } }
```

If the check box is not selected, these required properties are **not** created in the output with their default values.

API Gateway only

e. From the Empty JSON array handling list, select one of the following options to control how the map policy handles the output of an empty array:

- All (the default option): Output all empty arrays, including empty children arrays.
- Parent: Output only the current property's empty array value. Children map actions of this property are not attempted.
- None: Prevent any empty output array values from being produced.

f. From the Severity level for input data log messages list, select one of the following options to specify the severity level for log messages that relate to input data:

- error
- warn
- info

- g. API Gateway only Set the Schema definition circular reference limit field to an integer value that specifies the maximum allowed number of iterations of a circular schema definition. The default value is 1, which means that circular schema definitions are not followed. The maximum possible value is 5.
- h. Click Save when done.

## Results

---

You have configured a Map policy to transform and map variables in your assembly flow.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Map policy examples

Examples of the OpenAPI definitions of Map policies.

- [One to one mapping](#)
- [Many to one mapping](#)
- [A simple transformation using the value field](#)
- [Mapping from multiple contexts into a new context](#)
- [Mapping to an inline schema definition](#)
- [Mapping with a default value](#)
- [Mapping an array into a single value](#)
- [Mapping array elements to an array](#)
- [Using the advanced XML options](#)

## One to one mapping

---

The following example:

- maps parameters from the request's query to an object in the message body.
- maps one strings directly to a single string.
- maps one integer directly to a single integer.

The referenced definition, `output`, defines an object containing a string, name, and an integer, age.

```
- map:
  title: 1-1 map
  inputs:
    input_string:
      schema:
        type: string
        variable: request.parameters.name_in # the location of the variable, named "name_in" and found in the query
parameters of the request
    input_integer:
      schema:
        type: integer
        variable: request.parameters.age_in # another variable in the query parameters of the request
  outputs:
    output:
      schema:
        $ref: '#/definitions/output' # a schema definition reference to use for the output. The schema is for the
whole of the message body.
        variable: message.body
  actions:
    - set: output.name_out # in the actions section, variables are referenced by their name within the map policy
      from: input_string # if a value field is not used, the mapping is direct with the input value used for the
output variable
    - set: output.age_out # because the 'output' variable is itself an object, further references are made to
variables that it contains
      from: input_integer
```

## Many to one mapping

---

The following example:

- maps parameters from the request's query to an object in the message body.
- maps two strings to a single string by concatenating them.
- maps two integers to a single integer by summing them.

The referenced definition, `output`, defines an object containing a string and an integer.

```
- map:
  title: many-1 map
  inputs:
    input_string_1:
      schema:
        type: string
        variable: request.parameters.first_name
    input_string_2:
      schema:
```

```

    type: string
    variable: request.parameters.last_name
input_integer_1:
  schema:
    type: integer
    variable: request.parameters.balance_1
input_integer_2:
  schema:
    type: integer
    variable: request.parameters.balance_2
outputs:
  output:
    schema:
      $ref: '#/definitions/output'
      variable: message.body
actions:
- set: output.full_name
  from:
    - input_string_1
    - input_string_2
  value: |
    var retValue = undefined;
    if ($(input_string_1) !== undefined && $(input_string_2) !== undefined) {
      retValue = $(input_string_1).toUpperCase() + ' ' + $(input_string_2).toUpperCase()
    }
    retValue;
- set: output.total_balance
  from:
    - input_integer_1
    - input_integer_2
  value: |
    var i1 = 0;
    var i2 = 0;
    if ($(input_integer_1) !== undefined) i1 = $(input_integer_1);
    if ($(input_integer_2) !== undefined) i2 = $(input_integer_2);
    i1 + i2;

```

## A simple transformation using the value field

The following example:

- maps a parameter from the request's query to an object in the message body.
- maps one string featuring lowercase characters to a single string containing only uppercase characters.

The referenced definition, `output`, defines an object containing a single string.

```

- map:
  title: Uppercase map
  inputs:
    input_lowercase:
      schema:
        type: string
        variable: request.parameters.name_in
  outputs:
    output_uppercase:
      schema:
        $ref: '#/definitions/output'
        variable: message.body
  actions:
    - set: output_uppercase.name_out
      from: input_lowercase
      value: $(input_lowercase).toUpperCase() # the input variable is referenced and calls the toUpperCase method to
produce a value for the output variable

```

## Mapping from multiple contexts into a new context

The following example:

- maps an integer from the request's header to an integer in a new message body.
- maps a string from the message's body to the body of a new custom context, named `new_context`.

```

- map:
  title: Context map
  inputs:
    input_integer:
      schema:
        type: integer
        variable: request.headers.age_in # the 'age_in' header of the request is used as an input
    input_string:
      schema:
        type: string
        variable: message.body.name_in # as is the 'name_in' field of the request body
  outputs:
    output_integer:
      schema:
        type: integer
        variable: message.body.age_out
    output_string:
      schema:
        type: string
        variable: new_context.body.name_internal # the context named 'new_context' is created by the map policy and
exists only while the assembly is processed

```

```

actions:
  - set: output_string
    from: input_string
  - set: output_integer
    from: input_integer

```

## Mapping to an inline schema definition

The following example:

- maps parameters from the request's headers to an object in the message body, which is defined within the map policy.
- maps one string directly to a single string.
- maps one integer directly to a single integer.

```

- map:
  title: Inline schema map
  inputs:
    input_integer:
      schema:
        type: integer
        variable: request.headers.age_in
    input_string:
      schema:
        type: string
        variable: request.headers.name_in
  outputs:
    output:
      schema: # instead of a simple type or a reference to a definition, an inline YAML definition
        type: object
        properties:
          name_out:
            type: string
            name: name_out
          age_out:
            type: integer
            format: int32
            name: age_out
        title: output
        variable: message.body
  actions:
    - set: output.age_out
      from: input_integer
    - set: output.name_out
      from: input_string

```

## Mapping with a default value

The following example:

- maps one string directly to a single string.
- provides a default value for the output string if a valid input string is not provided.

```

- map:
  title: Default Map
  inputs:
    input_string:
      schema:
        type: string
        variable: request.headers.name_in
  outputs:
    output_string:
      schema:
        type: string
        variable: message.body.name_out
  actions:
    - set: output_string
      from: input_string
      default: John Smith # the default field is specified in the same way as a value, in this case

```

providing a fixed value

## Mapping an array into a single value

The following map policy:

- maps a single array of integers into a single integer.
- sums the integers in the array.
- \$ (0) represents the accumulated output because map evaluates all array element values.

```

- map:
  title: Summation map
  inputs:
    input:
      schema:
        $ref: '#/definitions/balance_array_in'
        variable: request.body
  outputs:
    output:
      schema:
        type: integer

```

```

        variable: message.body.total_balance_out # instead of using a full schema definition of the message body, a
single variable in the message body is specified
    actions:
    - set: output
      from: input
      foreach: input # the foreach field specifies that each value of the array 'input' is to be iterated over
      value: $(0)+$(input) # the $(0) reference is the accumulated value of the 'output' variable

```

## Mapping array elements to an array

The following map policy:

- maps an array, whose elements are objects containing two integers, to an array, whose entries contain a single integer field.
- maps an array to an array of the same length.
- takes the difference of the values of the integers in each array element to create a single integer in each array element.

```

- map:
  title: Array summation
  inputs:
  input:
    schema:
      $ref: '#/definitions/balance_and_credit_array'
    variable: request.body
  outputs:
  output_array:
    schema:
      type: array
      variable: message.body
  actions:
  - create: output_array
    from: input
    foreach: input
    actions:
    - set: total_balance_out
      from: # inside the actions section, variables inside the array elements being
iterated over are used
        - integer_in_1
        - integer_in_2
      value: $(integer_in_1)-$(integer_in_2) # the difference of the variables is taken for each array entry
of 'input'

```

## Using the advanced XML options

The following example:

- does not include empty elements in the output of the map policy.
- uses explicit name spaces rather than inheriting name spaces from the parent element
- indicates that all namespace declarations will be placed on the root XML element.
- uses all array elements rather than only the first
- ignores API Connect variable references
- maps input property values with a null value to the output document
- handles circular type references in an optimized manne

```

actions:
- set: output, two
  from: input, one
options:
includeEmptyXMLElements: false
namespaceInheritance: false
inlineNamespaces: false
mapArrayFirstElementValue: false
mapResolveApicVariables: false
mapNullValue: true
mapOptimizeSchemaDefinition: true
mapCreateEmptyArray: parent

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OAuth

An OAuth policy performs the requested OAuth processing based on the defined OAuth provider settings.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
---------	----------------

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [oauth](#).

## About

By adding an assembly OAuth policy, you specify the OAuth provider settings to use to perform the requested OAuth processing and the supported OAuth processing components.

Note: You add an OAuth policy to the assembly in a native OAuth provider. For more information, see the following topics:

- [Editing the native OAuth provider configuration using the API Editor](#) (Cloud Manager UI)
- [Editing the native OAuth provider configuration using the API Editor](#) (API Manager UI)

You can specify the settings through one or more of the following methods.

- URL reference
- Literal configuration
- Object reference

When you use more than one method to specify the same aspect of the OAuth provider settings, the following precedence rules apply to enforce the settings dynamically for the incoming transactions.

- URL reference takes precedence over any existing literal configuration or object reference.
- Literal configuration takes precedence over any existing object reference.

You can configure the policy to support one or more of the following processing components:

Validate request

Validates the authorization request from the client.

Generate authorization code

Generates the authorization code for the client, which represents the resource owner's authorization that grants access to the requested resource.

Verify authorization code

Verifies the authorization code from the client.

Verify refresh token

Verifies the refresh token that is presented by the client.

Generate access token

Generates the access token to the client when the authorization code or refresh token is verified.

Introspect token

Introspects the token to determine its state and, when active, its metadata.

When the policy does not support a processing component but that processing is requested, the unsupported component is not run.

You can include multiple OAuth policies in your OAuth provider assembly. For example, your assembly might include the following flow:

1. OAuth policy that validates the request.
2. GatewayScript policy.
3. OAuth policy that generates authorization code.

For more information, see [Example - using multiple OAuth policies in an OAuth provider assembly](#) and [OAuth context variables](#).

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. OAuth policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <code>oauth</code> .	string
Description	No	A description of the policy	string
Default OAuth Provider Settings Object	Yes	The name of an existing OAuth provider that defines the required settings.	string
Dynamic OAuth configuration from a URL	No	A URL to a document that contains serialized XML or JSON properties that defines OAuth token generate settings.	string
Dynamic OAuth configuration from a literal string	No	A literal string that contains serialized XML or JSON properties that defines OAuth token generate settings.	string
Supported OAuth components	No	Select the OAuth components that are supported by this policy.	string

## Overriding the default OAuth provider settings

You can use either the Dynamic OAuth configuration from a literal string property or the Dynamic OAuth configuration from a URL property to dynamically override any OAuth provider configuration settings defined by the Default OAuth Provider Settings Object property.

For example, to override the access token expiration time with a value of 200 seconds, include the following configuration in either the literal string or the document at the specified URL:

```
<OAuthProviderSettings><APICAccessTokenTTL>200</APICAccessTokenTTL></OAuthProviderSettings>
```

For a list of all OAuth provider settings, refer to the `OAuthProviderSettings` management schema, defined in the `xml-mgmt.xsd` file located in the store: directory on the DataPower API Gateway.

If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related reference

- [API Connect context variables](#)

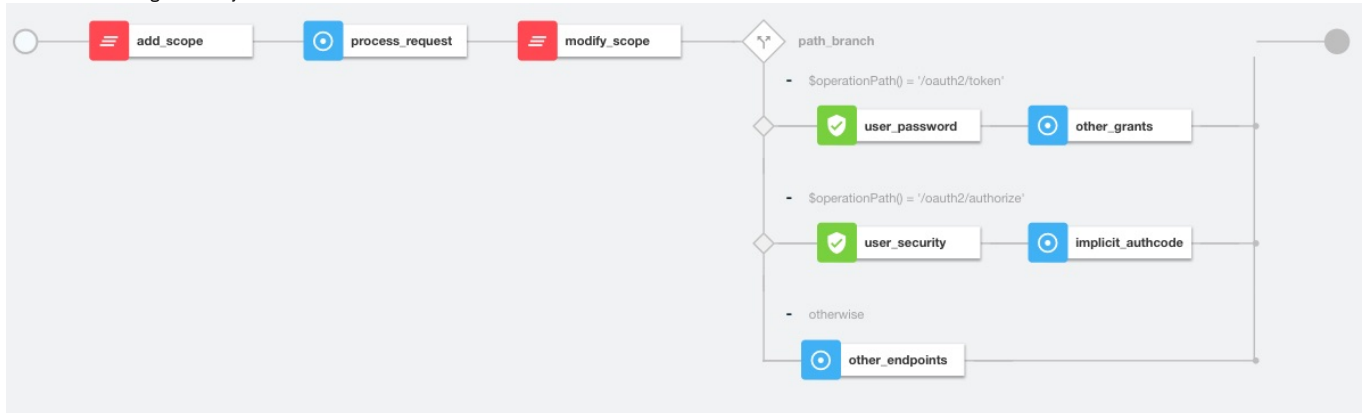
**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Example - using multiple OAuth policies in an OAuth provider assembly

This example demonstrates the use of multiple OAuth policies in the assembly flow for a native OAuth provider.

The example is based on the default assembly that is generated when you create a native OAuth provider, and is customized with the addition of `gatewayscript` policies that use [OAuth context variables](#) to manipulate the OAuth flow. For details on creating a native OAuth provider, see [Configuring a native OAuth provider](#).

It has the following assembly flow:



The following sections describe the OpenAPI source code that underlies each of the policies in the assembly; for the complete assembly code, download [multiple\\_oauth\\_policies.txt](#).

## Sample policy to add a custom scope

The `add_scope` policy is a `gatewayscript` policy that adds a custom scope to the request.

The underlying OpenAPI source YAML is as follows:

```
- gatewayscript:
  version: 2.0.0
  title: add_scope
  source: |-
    // Add another custom scope to the request
    let scope = context.get("request.parameters.scope.values[0]");
    if (scope)
      context.set("oauth.processing.scope", scope + " custom");
```

## Validate the initial OAuth request

The first `process_request` policy is an `oauth` policy that processes the initial request and verifies that the request is valid. The result of the processing is stored automatically in the `oauth.processing` context variables for use, as required, by the next OAuth policy in the assembly flow.

The underlying OpenAPI source YAML is as follows:

```
- oauth:
  title: process_request
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect protocol steps
    that are needed for OAuth Validation by default. The inputs and
    outputs of each of the steps are driven by documented context
    variables. Add or remove the Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
```

```
supported-oauth-components:
- OAuthValidateRequest
```

## Sample policy to modify the scope

---

The `modify_scope` policy is a `gatewayscript` policy that modifies the scope depending on the calling application.

The underlying OpenAPI source YAML is as follows:

```
- gatewayscript:
  version: 2.0.0
  title: modify_scope
  source: |-
    let admin_id = '1f1a2aa4-db9f-4423-b2f1-e2572b12123a';

    // Check application and modify the scope
    let app = context.get("oauth.processing.client_id");
    let scope = context.get("oauth.processing.scope");
    if (app === admin_id) {
      context.set("oauth.processing.scope", scope + " admin");
    } else {
      context.set("oauth.processing.scope", scope + " customer");
    }
  }
```

## Branch conditionally according to the OAuth path

---

The `path_branch` policy is a `switch` policy that branches according to the different OAuth paths to process the resource owner.

The underlying OpenAPI source YAML is as follows:

```
- switch:
  version: 2.0.0
  title: path_branch
  case:
    - condition: ($operationPath() = '/oauth2/token')
      execute:
        .
        .
        .
        definition of the user_security and other_grants policies
        .
        .
        .
    - condition: ($operationPath() = '/oauth2/authorize')
      execute:
        .
        .
        .
        definition of the user_password and implicit_authcode policies
        .
        .
        .
    - otherwise:
      .
      .
      .
      definition of the other_endpoints policy
      .
      .
      .
```

## Process the user name and password, and enable grant type component

---

The following two policies operate on the token endpoint.

- The `user_password` policy for password grant type processes the user name and password from the `x-www-form-urlencoded` body. The authentication method is derived from the User Security settings in the OAuth provider; see [Configuring user security for a native OAuth provider](#).
- The `other_grants` policy is an `oauth` policy that enables the `OAuthGenerateAccessToken`, `OAuthVerifyAZCode`, `OAuthVerifyRefreshToken`, and `OAuthCollectMetadata` components to perform the operations for client credentials, authorization code, refresh token, and password grant types.

The underlying OpenAPI source YAML is as follows:

```
- user-security:
  title: user_password
  version: 2.0.0
  description: ''
  factor-id: default
  extract-identity-method: context-var
  user-context-var: request.parameters.username.values
  pass-context-var: request.parameters.password.values
  ei-stop-on-error: false
  user-auth-method: auth-url
  au-stop-on-error: false
  auth-url: 'http://httpbin.org/basic-auth/user/pass'
  user-az-method: authenticated
  az-stop-on-error: true
  auth-response-headers-pattern: (?x-api*
  auth-response-header-credential: X-API-Authenticated-Credential
- oauth:
  title: other_grants
```



```

version: 2.0.0
description: >-
  This oauth policy performs all OAuth/OpenID Connect
  protocol steps that are needed for token path by default.
  The inputs and outputs of each of the steps are driven by
  documented context variables. Add or remove the Supported
  OAuth Components as required.
oauth-provider-settings-ref:
  default: custom-form
supported-oauth-components:
  - OAuthGenerateAccessToken
  - OAuthVerifyAZCode
  - OAuthVerifyRefreshToken
  - OAuthCollectMetadata

```

## Perform authorization checks, and enable grant type components

---

These following two policies operate on the authorize endpoint.

- The `user_security` policy configuration is derived from the User Security settings in the OAuth provider; see [Configuring user security for a native OAuth provider](#).
- The `implicit_authcode` policy is an `oauth` policy that enables the `OAuthGenerateAZCode`, `OAuthGenerateAccessToken`, and `OAuthCollectMetadata` components to perform the operations for the implicit and authorization code grant types.

The underlying OpenAPI source YAML is as follows:

```

- user-security:
  title: user_security
  version: 2.0.0
  description: >-
    This user security policy performs EI(basic) and AU(auth
    url) check for oauth assembly. Change the security check
    method as required
  factor-id: default
  extract-identity-method: basic
  ei-stop-on-error: true
  user-auth-method: auth-url
  au-stop-on-error: true
  user-az-method: authenticated
  az-stop-on-error: true
  auth-response-headers-pattern: (?)*x-api*
  auth-response-header-credential: X-API-Authenticated-Credential
  auth-url: 'http://httpbin.org/basic-auth/user/pass'
- oauth:
  title: implicit_authcode
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect
    protocol steps that are needed for az code path by
    default. The inputs and outputs of each of the steps are
    driven by documented context variables. Add or remove the
    Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthGenerateAZCode
    - OAuthGenerateAccessToken
    - OAuthCollectMetadata

```

## Process all other endpoints

---

The `otherwise` condition catches all other endpoints such as the introspect and revoke endpoints. The `other_endpoints` policy in the `otherwise` condition is an `oauth` policy that enables the `OAuthIntrospectToken`, and `OAuthRevokeTokencomponents` components to perform the operations for introspect and revoke.

The underlying OpenAPI source YAML is as follows:

```

- oauth:
  title: other_endpoints
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect
    protocol steps that are needed for all other paths by
    default. The inputs and outputs of each of the steps are
    driven by documented context variables. Add or remove the
    Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthIntrospectToken
    - OAuthRevokeToken

```

## Related concepts

---

- [API policies and logic constructs](#)

## Related reference

---

- [OAuth context variables](#)

## Related information

- [Configuring a native OAuth provider](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Parse

Use the Parse policy to control the parsing of an input document.

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [parse](#).

## About

When the input document is a JSON string, the string is parsed instead of copied over. You can re-parse a JSON string as any of the allowed doc types (**detect**, **xml**, **json**, **binary**). For example, suppose that the JSON string "`<a></b></a>`" is parsed. You can re-parse it using doc type **xml** to extract the XML from that string.

With the parse policy, you can retrieve parse settings by using the following methods in increasing order of precedence:

- Specify a valid parse settings configuration from which to retrieve its properties as the default parse settings.  
Note: If you change the default name, you **must** separately configure a corresponding ParseSettings object on the DataPower API Gateway.
- Specify a literal string as serialized XML or JSON properties as parse settings. These properties take precedence over any existing default properties. The literal string must be XML or JSON formatted.
  - The XML format is "`<ParseSettings><propertyName>propertyValue</propertyName></ParseSettings>`".
  - The JSON format is "`{ \"ParseSettings\" : { \"propertyName\" : propertyValue } }`".
- A URL that represents a named context from which to retrieve the serialized XML or JSON properties as parse settings. These properties take precedence over any existing literal or default properties.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Parse policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>parse</b> .	string
Description	No	A description of the policy.	string
Use Content Type	No	If this setting is enabled and the parse setting is configured to detect the document type, the parse action uses the <b>Content-Type</b> specified in the request headers. If this setting is enabled and the document type in the parse setting is configured for either JSON or XML, the parse action uses the <b>Content-Type</b> specified in the request headers and fails if the <b>Content-Type</b> in the request headers does not match the parse setting.  Enabling this setting is applicable only when the expected <b>Content-Type</b> is either JSON or XML.  If this setting is not enabled, the parse action uses either the document type that is specified in the parse setting, or the detected document type if the parse setting is configured to detect the document type.  The check box is cleared by default.	boolean
Parse settings object reference	No	An existing valid object from which to retrieve default property values for the dynamic object.	string
Parse settings literal configuration	No	A literal string as serialized XML or JSON properties that are merged into the dynamic object. The literal property overrides any existing default properties.	string
Parse settings URL reference	No	A URL that represents a named context from which to retrieve the serialized XML or JSON properties that are merged into the dynamic object. These properties override any existing literal or default properties.	string
Message for parsing	No	The name of a variable in the API context. The content of the <b>body</b> field of the variable is the input to the policy. The default variable name is <b>message</b> .	string
Message for resulting parsed data	No	The name of a variable in the API context. The content of the body field of the variable is the output of the parse operation. The parse metrics of the parsed document can be stored in a different part of the message. The default variable name is the same as the input name, so by default the input message is overwritten by the output message.	string

For examples, see [parse](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Proxy

Apply the Proxy policy to invoke another API within your assembly, particularly if the separate API contains a large payload. The response from the backend is stored in the `message.body` and in the response object variable if it is defined. Only one policy is permitted to be run per unique assembly flow.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, functionality provided by <a href="#">Invoke</a>	

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [proxy](#).

## About

You can attach this policy to the following API flows:

- REST
- SOAP

Only one Proxy policy is permitted to be run per unique flow of your assembly. More than one Proxy policy can be applied, if they are contained in mutually exclusive branches of the assembly.

You can use the Proxy policy to return multipart form data, that is, when the response is set to `Content-Type: multipart/related`. However, Proxy must be the last policy in the assembly, otherwise the response that is received can be manipulated during subsequent steps, thereby causing the multipart form data to be lost.

The proxy policy, if inside a conditional policy, must be the **final** policy to be executed in the API. If you need further processing afterward, use the [invoke](#) policy rather than the proxy policy.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Proxy policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <b>proxy</b> .	string
Description	No	A description of the policy.	string
Invoke URL	Yes	Specifies a URL for the target service. For a SOAP API, a URL is added by default. Where possible, the Proxy URL value is pre-supplied from information that is defined in the imported WSDL.	string
TLS profile	No	Specifies a TLS profile to use for the secure transmission of data.	string
Timeout	Yes	The time to wait before a reply back from the endpoint (in seconds). The default value is <b>60</b> .	integer
Username	No	The username to use for HTTP Basic authentication.	string
Password	No	The password to use for HTTP Basic authentication.	string
HTTP Method	Yes	The HTTP method to use for the proxy. Valid values are: <ul style="list-style-type: none"> <li>• Keep</li> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• PATCH</li> <li>• HEAD</li> <li>• OPTIONS</li> </ul> The default value is <b>Keep</b> . By using <b>Keep</b> , or removing the property from the source, the HTTP method from the incoming request is used.	string
Compression	No	Select this check box to enable Content-Encoding compression on upload. The check box is cleared by default.	boolean

Property label	Required	Description	Data type
Cache Type	No	<p>The cache type determines whether to cache documents, honoring or overriding the HTTP Cache Control directives received in the response from the target URL. This property takes effect only when a response is received, otherwise the policy always returns the non-expired response that was previously saved in cache. Valid values are:</p> <p><b>Protocol</b> The cache behavior is determined by the Cache-Control headers on the response, in accordance with RFC 7234. To optimize performance, if the gateway receives more than one request for a resource that is not in the cache but could be cached when the response from the target URL is received, the gateway sends only one request to the target URL; the remaining requests are not processed until the response from the first request has been received and the cache behavior has been determined from this response. If the response indicates that caching is possible, the gateway responds to all waiting requests with the cached resource. If the response indicates that caching is not possible, the gateway sends all waiting requests to the target URL.</p> <p>Use this option only if you expect that responses from the target URL can be cached, in which case it should improve performance and limit the demand on the target URL. If, however, the target URL never indicates that the gateway should cache its response, performance might be impaired when compared to the No Cache option.</p> <p><b>No Cache</b> Responses from the target URL are not cached on the gateway regardless of any caching headers returned. In this case, every request from the client is sent to the target URL. Use this option if you do not want to cache any of the backend responses on the gateway, or if it is unlikely that a response from the target URL will allow caching through the Cache-Control header settings.</p> <p><b>Time to Live</b> This option is similar to the Protocol option except it allows you to specify the amount of time that you want the successful response from the invoke or proxy to remain in the cache. Use this option only if you expect that responses from the target URL can be cached.</p> <p>The default value is Protocol.</p>	string
Time to Live	No	Specifies the amount of time in seconds that the response stays in the cache. Applies only if the property Cache type is set to <b>Time to Live</b> . Enter a value in the range 5 - 31708800. The default value is 900.	integer
Cache key	No	Specifies the unique identifier of the document cache entry. If omitted, the entire URL string is used as the key.	string
Stop on error	No	Select the errors that, if thrown during the policy execution, cause the assembly flow to stop. If there is a <b>catch</b> flow configured for the error, it is triggered to handle the error thrown. If an error is thrown and there are no errors selected for the Stop on error setting, or if the error thrown is not one of the selected errors, the policy execution is allowed to complete, and the assembly flow continues.	string
Response object variable	No	The name of a variable that will be used to store the response data from the request. This variable can then be referenced in other actions, such as 'Map'.	string
<b>X-Forwarded</b> header	No	<p>This header can be provided by</p> <ol style="list-style-type: none"> <li>If <b>X-Forwarded-Host</b> exists, processing continues. If it does not exist prior to calling the proxy policy, it is set with the value of the <b>Host</b> header.</li> <li>The <b>X-Forwarded-For</b> header is always set, in all cases. This header maintains breadcrumbs, showing a comma-separated list of IPs, from the client through any preceding proxy.</li> <li>If all three of <b>X-Forwarded-Host</b>, <b>X-Forwarded-Port</b>, and <b>X-Forwarded-Proto</b> headers are missing at the time of calling the proxy policy, they are set automatically. To prevent this, set <b>X-Forwarded-Host</b> header to some value before calling the proxy policy.</li> </ol>	string

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related information

- [TLS profiles](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Rate Limit

Use the Rate Limit policy to apply one or more rate or burst limits at any point in your API assembly flow. Rate and burst limits restrict the number of calls made to an API in a specified time period.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [ratelimit](#).

## About

The defined rate and burst limits are applied to whatever follows in the assembly flow. For example, if a Rate Limit policy is placed before an Invoke policy, and the call made by the Invoke policy exceeds the limits defined by the Rate Limit policy, the API call itself fails.

Note: For information about rate limits and burst limits in API Connect, see [Understanding rate limits for APIs and Plans](#).

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Rate Limit policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <code>ratelimit</code> .	string
Description	No	A description of the policy.	string
Source	Yes	The location of all the rate limit and burst limit definitions that are included in this policy. Select one of the following options: <ul style="list-style-type: none"> <li>• Catalog by Name: the limits to be applied are defined in the appropriate <code>api-collection</code> object on the DataPower API Gateway, which is the object that represents your API Connect Catalog in the gateway configuration. For details of how to configure a rate or burst limit in the <code>api-collection</code> object, see <a href="#">Configuring a rate or burst limit on the DataPower API Gateway</a>.</li> <li>• Plan Default: the rate and burst limits that are applied are the default ones configured in the Plan to which the calling application is subscribed. For details on how to configure default Plan rate and burst limits, see <a href="#">Editing a draft Product</a>.</li> </ul>	string
Rate Limit Name	Yes <sup>1</sup>	The name of a rate limit as defined in the DataPower API Gateway configuration. To display the Rate Limit Name field, click Add Rate Limit; you can add as many rate limits as you want. <sup>1</sup> You must provide at least one rate limit or at least one burst limit.	string
Burst Limit Name	Yes <sup>2</sup>	The name of a burst limit as defined in the DataPower API Gateway configuration. To display the Burst Limit Name field, click Add Burst Limit; you can add as many burst limits as you want. <sup>2</sup> You must provide at least one rate limit or at least one burst limit.	string

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Configuring a rate or burst limit on the DataPower API Gateway

If you want to include a rate limiting policy in your API assembly flow, you must first configure the required rate and burst limits on the Gateway. Rate and burst limits restrict the number of calls that an application can make to API in specified time period.

## About this task

You configure rate and burst limits on the DataPower® API Gateway by defining them in one or more configuration files that you package and then add to the Gateway as a Gateway extension.

You define the rate and burst limits in the appropriate **api-collection** object on the DataPower API Gateway, which is the object that represents your API Connect Catalog in the gateway configuration. The defined rate and burst limits will be available to be included in the assembly flows of all the APIs that are published to that Catalog.

Note: Only one Gateway extension can be in use at any one time, so you should package all required rate and burst limits in a single Gateway extension.

## Procedure

1. Create a .cfg file that includes the DataPower API Gateway CLI commands that define the rate or burst limit.

For example:

```
top; config;
switch apiconnect;

api-collection myorg_mycatalog_collection
  assembly-rate-limit 30perlmin 30 1 minute on off on on off off ""
exit
```

In this example, the **assembly-rate-limit** command specifies a rate limit name, and the API call limits that you want to impose, 30 calls per minute in this case. To define a burst limit, use the **assembly-burst-limit** command. For details of the command parameters, see [assembly-rate-limit](#) and [assembly-burst-limit](#) in the DataPower product documentation.

The **api-collection** command specifies the **api-collection** object in which you want to configure the rate or burst limits. The name of the **api-collection** object has the following format:

```
org_catalog_collection
```

where:

- *org* is the name of the provider organization that contains your Catalog.
- *catalog* is the name of the Catalog.

This **api-collection** object name is assigned automatically by the Gateway the first time that a Product is published to the Catalog.

2. Package the .cfg file in a .zip file.  
You can package multiple .cfg files in a single .zip file if you want; the .cfg files must all be at the root of the .zip file.
3. Add the .zip file to the DataPower API Gateway as a Gateway extension; for details, see [Configuring your Gateway server extensions](#).

## What to do next

Apply the rate or burst limit to an API by adding a [Rate Limit](#) policy to the API assembly flow.

## Related information

- [Gateway extension guidelines - DataPower API Gateway](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Redaction - DataPower API Gateway

Use the Redaction policy to completely remove or to redact specified fields from the Request body, the Response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.

## Gateway support

Note: This page describes the Redaction policy implementation in the DataPower® API Gateway. If you are using the DataPower Gateway (v5 compatible), see [Redaction - DataPower Gateway \(v5 compatible\)](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [redact - DataPower API Gateway](#).

Note: With the DataPower API Gateway, the input to the Redaction policy must be parsed data. One way to produce parsed data is to use a [Parse](#) policy before a Redaction policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Redaction policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>redact</b> .	string
Description	No	A description of the policy.	string
Root	No	Specifies the data source that contains the content to redact or remove. If no value is entered in the Root field, the action is applied to the entire API context. You can use any supported JSONata path expression.  If you want to apply the action to either request or response data, specify a value of <code>message.body</code> . The actual content to which the action is applied then depends on the positioning of the Redaction policy in the overall assembly flow; for example: <ul style="list-style-type: none"><li>• If positioned at the beginning, the action is applied to the client request.</li><li>• If positioned after an Invoke policy, the action is applied to the response from the back end.</li><li>• If positioned at the end, the action is applied to the response that is returned to the client.</li></ul> If, in your assembly flow, the Redaction policy is used after a <a href="#">Log</a> policy that specifies Gather-only mode, specify a Root value of <code>log.request_body</code> for the logged request payload, or <code>log.response_body</code> for the logged response payload.	string
Path	Yes	Specifies a JSONata path expression that identifies the content to redact or remove from the source. For more information, see <a href="#">Constructing JSONata expressions to redact fields</a> .	string
Action	No	Specifies whether you want to remove or redact the content. Choose one of the following options: <ul style="list-style-type: none"><li>• Remove: Completely removes the specified fields.</li><li>• Redact: Redacts (obfuscates with "*"s) the fields to block out the data.</li></ul> The default value is Redact.  Note: If a numerical value is being redacted, the redacted value is depicted as <b>*****</b> and the type is changed to <b>string</b> .	string

Tip: You can optionally click Add action to specify JSONata expressions for additional fields that you want to remove or redact from the specified content source.

Note: IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

API Gateway only

## Constructing JSONata expressions to redact fields

To define a field for redaction or removal when using the Redaction policy with the DataPower® API Gateway, you supply a JSONata expression that defines the path to the field that you want to redact or remove.

The following JSONata functions are supported:

- Aggregation functions:
  - `$average(array)`
  - `$max(array)`
  - `$min(array)`
  - `$sum(array)`
- Array functions:
  - `$append(array1, array2)`
  - `$count(array)`
  - `$reverse(array)`
  - `$sort(array [, function])`
  - `$zip(array1, ...)`
- Boolean functions:
  - `$boolean(arg)`
  - `$exists(arg)`
  - `$not(arg)`
- Numeric functions:
  - `$abs(number)`
  - `$ceil(number)`
  - `$floor(number)`
  - `$formatBase(number [, radix])`
  - `$number(arg)`
  - `$power(base, exponent)`
  - `$round(number [, precision])`
  - `$sqrt(number)`
- Object functions:

- `$keys(object)`
- `$lookup(object, key)`
- `$merge(array<object>)`
- `$spread(object)`
- String functions:
  - `$contains(str, pattern)`
  - `$join(array[, separator])`
  - `$length(str)`
  - `$lowercase(str)`
  - `$match(str, pattern [, limit])`
  - `$pad(str, width [, char])`
  - `$replace(str, pattern, replacement [, limit])`
  - `$split(str, separator [, limit])`
  - `$substring(str, start[, length])`
  - `$substringAfter(str, chars)`
  - `$substringBefore(str, chars)`
  - `$trim(str)`
  - `$uppercase(str)`

You can use the following JSONata numeric operators:

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)
- `/` (division)
- `%` (modulo)

You can use the following JSONata comparison operators for number values or strings:

- `=`
- `!=`
- `<`
- `>`
- `<=`
- `>=`

You can define the path in either of the following ways:

- [Use a JSONata expression](#)
- [Use the `\$xpath\(\)` JSONata extension](#)

## Using a JSONata expression

The path specified by the JSONata expression is relative to any value specified for the `root` property of the Redaction policy. If the `root` property has no value or is absent, begin the expression with the absolute content path. If the `root` property has a value then you can either begin the expression with `$` to use the `root` path directly, or you can provide a sub-path relative to the `root` path.

JSONata expressions can be used with content that is in either JSON or XML format.

### Example 1

If the `root` property of the Redaction policy has no value or is absent, use the following expression to redact or remove all occurrences of the `price` field in the request and response data:

```
message.body.**.price
```

### Example 2

If the `root` property of the Redaction policy has the value `log.request_body`, use the following expression to redact or remove all occurrences of the `price` field, specifically within an `item` element, in the logged request payload:

```
$.item.price
```

### Example 3

If the `root` property of the Redaction policy has the value `log`, use the following expression to redact or remove all occurrences of the `price` field in the logged response payload:

```
response_body.**.price
```

The `**` descendant wildcard traverses all descendants at all hierarchical levels.

## Using the `$xpath()` JSONata extension

The `$xpath()` function has the following format:

```
$xpath(content_path, xpath_expression)
```

where:

- `content_path` is the path to the content that contains the field that you want to redact or remove.
- `xpath_expression` is the XPath expression that defines the field that you want to redact or remove.

The `content_path` is relative to any value specified for the `root` property of the Redaction policy. If the `root` property has no value or is absent, provide the absolute content path. If the `root` property has a value then you can either provide the value `$` for the `content_path` parameter to use the `root` path directly, or you can provide a sub-path relative to the `root` path.



The `$xpath()` function can be used only with content that is in XML format.

#### Example 1

If the `root` property of the Redaction policy has no value or is absent, use the following expression to redact or remove all occurrences of the `price` field in the request and response data:

```
$xpath(message.body, '//price')
```

#### Example 2

If the `root` property of the Redaction policy has the value `log.request_body`, use the following expression to redact or remove all occurrences of the `price` field, specifically within an `item` element, in the logged request payload:

```
$xpath($, 'item/price')
```

#### Example 3

If the `root` property of the Redaction policy has the value `log`, use the following expression to redact or remove all occurrences of the `price` field in the logged response payload:

```
$xpath(response_body, '//price')
```

The `//` expression in the second parameter selects all occurrences anywhere in the content.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible)

## Redaction - DataPower Gateway (v5 compatible)

Use the Redaction policy to completely remove or to redact specified fields from the Request body, the Response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.

### Gateway support

Note: This topic describes the Redaction policy implementation in the DataPower® Gateway (v5 compatible). If you are using the DataPower API Gateway, see [Redaction - DataPower API Gateway](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [redact - DataPower Gateway \(v5 compatible\)](#).

### About

You can attach this policy to the following API flows:

- REST
- SOAP

### Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Redaction policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>redact</b> .	string
Description	No	A description of the policy.	string
Path	Yes	Specifies an XPath expression that defines the field to remove or redact. You can construct an XPath expression that is based on JSON or XML depending on whether your API requests and responses use a JSON or an XML format. If the payload is JSON, use the DataPower XML representation of the JSON content (JSONx) to construct the expression.  Note: Use a JSONx representation only to identify the XPath expressions for the fields to remove or redact. Do not change the format of any response bodies in API Manager. To learn more about constructing XPath expressions that are based on JSON or XML, see <a href="#">Constructing XPath expressions to redact fields</a> .	string

Property label	Required	Description	Data type
Action	Yes	<p>Specifies whether you want to remove or redact the field.</p> <p>Valid values:</p> <ul style="list-style-type: none"> <li><b>remove</b>: Completely removes the specified field.</li> <li><b>redact</b>: Redacts (obfuscates with "*"s) the field to block out the data.</li> </ul> <p>The default value is <b>redact</b>.</p> <p>Note: If a numerical value is being redacted, the redacted value is depicted as ***** and the type is changed to <b>string</b>.</p>	string
From	Yes	<p>Specifies where to remove or redact the specified field from.</p> <p>Valid values:</p> <ul style="list-style-type: none"> <li><b>all</b>: Removes or redacts the specified field from the Request body, the Response body, and the activity logs.</li> <li><b>request</b>: Removes or redacts the specified field from the Request body.</li> <li><b>response</b>: Removes or redacts the specified field from the Response body.</li> <li><b>logs</b>: Removes or redacts the specified field from the activity logs.</li> </ul> <p>The default value is <b>all</b>.</p> <p>Optionally click Add item to specify additional values.</p>	string

Tip: You can optionally click Add item to specify XPath expressions for additional fields that you want to remove or redact from the Request body, Response body, and logs.

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Constructing XPath expressions to redact fields

To define a field for redaction when using the DataPower® Gateway (v5 compatible), you supply an XPath expression that specifies the field that you want to redact. If your API requests and responses use XML format, you can base your XPath statements directly on the XML content. If your API requests and responses use JSON format, you must use the DataPower XML representation of the JSON content, JSONx, to construct your XPath expression.

## About this task

The following sections provide examples for constructing XPath expressions to redact a field; examples are provided for API responses in both XML and JSON format:

## XPaths for XML

Consider the following example of an XML response:

```
<xml>
  <primaryAddress>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </primaryAddress>
  <secondaryAddress>
    <streetAddress>412 Brooklyn Avenue</streetAddress>
    <city>New Jersey</city>
    <state>NJ</state>
    <postalCode>12302</postalCode>
  </secondaryAddress>
</xml>
```

To redact "streetAddress" in the preceding example the XPath expression would be: `//streetAddress` where the components of the expression have the following meaning:

Expression	Meaning
<code>//</code>	search anywhere in the XML structure
<code>//streetAddress</code>	search anywhere in the XML structure for an element of type "streetAddress"

The XML response that results from applying this XPath expression is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <primaryAddress>
    <streetAddress>*****</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </primaryAddress>
  <secondaryAddress>
    <streetAddress>*****</streetAddress>
    <city>New Jersey</city>
    <state>NJ</state>
    <postalCode>12302</postalCode>
  </secondaryAddress>
</xml>
```

Note: Both incidences of "streetAddress" have been redacted.

To redact one specific incidence of "streetAddress" in the initial example, the XPath expression would be: //secondaryAddress/streetAddress where the components of the expression have the following meaning:

Expression	Meaning
//secondaryAddress	search anywhere in the XML structure for an element of type "secondaryAddress"
//secondaryAddress/streetAddress	search under the preceding element for a child element of type "streetAddress"

The XML response that results from applying this XPath expression is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <primaryAddress>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </primaryAddress>
  <secondaryAddress>
    <streetAddress>*****</streetAddress>
    <city>New Jersey</city>
    <state>NJ</state>
    <postalCode>12302</postalCode>
  </secondaryAddress>
</xml>
```

Note: Only the secondary address has been redacted.

XPaths for JSON

Consider the following example of a JSON response:

```
{
  "primaryAddress": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "secondaryAddress": {
    "streetAddress": "412 Brooklyn Avenue",
    "city": "New Jersey",
    "state": "NJ",
    "postalCode": 12302
  }
}
```

Unlike the preceding XML example, to construct an XPath for this example you must use the corresponding DataPower XML representation, JSONx. The JSONx equivalent of this JSON response is as follows:

```
<json:object xsi:schemaLocation="http://www.datapower.com/schemas/json jsonx.xsd"
xmlns:json="http://www.ibm.com/xmlns/prod/2009/jsonx" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <json:object name="primaryAddress">
    <json:string name="streetAddress">21 2nd Street</json:string>
    <json:string name="city">New York</json:string>
    <json:string name="state">NY</json:string>
    <json:number name="postalCode">10021</json:number>
  </json:object>
  <json:object name="secondaryAddress">
    <json:string name="streetAddress">412 Brooklyn Avenue</json:string>
    <json:string name="city">New York</json:string>
    <json:string name="state">NY</json:string>
    <json:number name="postalCode">10021</json:number>
  </json:object>
</json:object>
```

To redact the element "streetAddress" from the preceding JSONx structure, the XPath expression would be: //\*[@name='streetAddress']

where the components of the expression have the following meaning:

Expression	Meaning
//*	find any element anywhere in the structure
[@name='streetAddress']	this element has a 'name' property with value "streetAddress"

The JSON response that results from applying this XPath expression is as follows:

```
{
  "primaryAddress": {
    "streetAddress": "*****",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  }
}
```

```

    },
    "secondaryAddress": {
      "streetAddress": "*****",
      "city": "New York",
      "state": "NY",
      "postalCode": 10021
    }
  }
}

```

Note: All incidences of "streetAddress" have been redacted in the preceding JSON structure. To redact a specific occurrence of the "streetAddress" element, the XPath expression would be: `//*[ @name='secondaryAddress']/*[@name='streetAddress']` where the components of the expression have the following meaning:

Expression	Meaning
<code>/**</code>	find any element anywhere in the structure
<code>//*[ @name='secondaryAddress']</code>	find an element anywhere in the structure that is named "secondaryAddress"
<code>//*[ @name='secondaryAddress']/*[@name='streetAddress']</code>	find a child element of any type (/* ) where the child element has a name of "streetAddress"

The JSON response that results from applying this XPath expression is as follows:

```

{
  "primaryAddress": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "secondaryAddress": {
    "streetAddress": "*****",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  }
}

```

Note: Only the street address for the secondary address has been redacted.

## What to do next

See [Including components in your assembly](#) to learn how to apply a redact policy using XPath. **Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Set Variable

Use the Set Variable policy to set the value of a runtime variable, or to clear a runtime variable, or to add a header variable.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [set-variable](#).

## About

You can attach this policy to the following API flows:


- REST
- SOAP

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Set Variable policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>set-variable</b> .	string
Description	No	A description of the policy.	string

Property label	Required	Description	Data type
Action	Yes	<p>Defines what action to apply on a runtime variable. Choose one of the following values:</p> <ul style="list-style-type: none"> <li>Set: Indicates that you want to set a runtime variable to a string value. Can be used to set new headers or to override existing values.</li> <li>Add: Indicates that you want to add a header variable. Can be used only to set new headers or to append a new entry of the same header name.</li> <li>Clear: Indicates that you want to delete a runtime variable. Can be used to remove a header when the data is processed in the assembly flow.</li> </ul> <p>The default value is Set.</p>	string
Set, Add, or Clear	Yes	Specifies the name of the variable that you want to set , add or clear, depending on the selected Action.	string
Type	Yes	<p>Select the data type of the variable. Choose one of the following values:</p> <ul style="list-style-type: none"> <li> any</li> <li>string</li> <li>number</li> <li>boolean</li> </ul> <p>If you are using the DataPower API Gateway then, for all values other than any, the value is validated against the specified data type.</p>	string
Value	Yes*	<p>Allocates this value to the specified variable. Can be a literal value, or another variable. * Value is required only when <b>Set</b> or <b>Add</b> is specified as the action.</p> <p>For example, to <i>set</i> a named variable of <b>billing-hostname</b> to a literal value, you can specify the Value as <b>acme . com</b>.</p> <p>As another example, to <i>set</i> a named variable to the value of the Content-Type header in a request, you can specify the Value entry as <b>\$(request . headers . content-type)</b>.</p> <p>If the selected value of the Type field is boolean, select the Value check box to indicated a value of <b>true</b>.</p> <p>Note: You can only set single string elements. Values are retrieved as strings and therefore you cannot clone a complete nodeset.</p>	string

For examples, see [set-variable](#).

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## User Security

Use the User Security policy to extract a user's credentials, authenticate those credentials, and obtain authorization from the user.

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [user-security](#).

## About

When you define an assembly user security action, you can define the processing for identity-extraction, authentication, and authorization or you can selectively disable any of these this aspects of processing. When disabled, this processing aspect is skipped.

When identity-extraction is enabled, the following methods are supported.

- Use basic authentication, which requires no additional configuration.
- Use context variables. For this method, specify which variable contains the user name and password.
- Use a redirect. For this method, specify the URL fragment to redirect to, and the time allowed to process.
- Use an HTML login form. For this method, specify whether to use the default or custom form and the time allowed to submit the form. For a custom form, specify the location of the form and the TLS client profile to secure the connection to the remote server.

When authentication is enabled, the following methods are supported.

- Contact an LDAP server. For this method, specify which server to contact.
- Send a request to an authentication endpoint. For this method, specify the URL of the endpoint, the TLS client profile to secure the connection, the pattern to select which response header to add, and the response header that contains the authenticated credentials.

When authorization is enabled, the following methods are supported.

- Implicitly accept any previously authenticated users, which requires no additional configuration.
- Use an HTML authorization form. For this method, specify whether to use the default or custom form and the time allowed to submit the form. For a custom form, specify the location of the form and the TLS client profile to secure the connection to the remote server.

You can attach this policy to the REST API flow.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. User Security policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is <b>user-security</b> .	string
Description	No	A description of the policy.	string
Factor ID	No	The identity that identifies the results of factor-authentication in the API context.	string
Extract Identity Settings	Yes	<p>Select the method that is used to extract the user credentials. The following options are available:</p> <p><b>Basic</b> Use basic authentication; no additional configuration is required.</p> <p><b>Context Variable</b> The credentials are provided by API Connect context variables; specify the following properties:</p> <ul style="list-style-type: none"> <li>• Username content variable: the context variable that is used to obtain the user name.</li> <li>• password context variable: the context variable that is used to obtain the password.</li> </ul> <p><b>HTML Form</b> Use forms based identity-extraction. Select whether to use the default form or a custom form. For a custom form, specify the following properties:</p> <ul style="list-style-type: none"> <li>• Custom form endpoint: the location of the form.</li> <li>• Custom form TLS profile: the TLS client profile that is used to secure the connection to the remote server.</li> </ul> <p>In the HTML form time limit field, specify the time allowed to submit the form.</p> <p><b>Redirect</b> Use a redirect for identity-extraction; specify the following properties:</p> <ul style="list-style-type: none"> <li>• Redirect URL: the URL fragment to which to redirect the request to obtain user credentials.</li> <li>• Redirect time limit: the time allowed for the transaction to complete.</li> </ul> <p><b>Disabled</b> Identity-extraction is disabled; this aspect of processing is skipped.</p> <p>Select Stop on error to halt assembly processing in the event of identity-extraction failure.</p>	string
Authenticate User Settings	Yes	<p>Select the authentication method. The following options are available:</p> <p><b>Authentication URL</b> The credentials are authenticated by an external endpoint; specify the following properties:</p> <ul style="list-style-type: none"> <li>• Authentication URL: the URL of the authentication endpoint.</li> <li>• Authentication TLS profile: the TLS client profile that is used to secure the connection to the authentication endpoint.</li> <li>• Authentication response header pattern: the pattern that is used to select which response headers to add to the API context.</li> <li>• Authentication response header credential: the response header that contains the authenticated user credentials.</li> </ul> <p><b>LDAP</b> The credentials are authenticated by an LDAP user registry; from the LDAP registry list, select the required registry.</p> <p><b>Disabled</b> Authentication is disabled; this aspect of processing is skipped.</p> <p>Select Stop on error to halt assembly processing in the event of authentication failure.</p>	string

Property label	Required	Description	Data type
Authorize User Settings	Yes	<p>Select the authorization method. The following options are available:</p> <p>authenticated</p> <p>Implicitly accept any previously authenticated users; no additional configuration is required.</p> <p>HTML Form</p> <p>The user provides authorization through an HTML form. Select whether to use the default form or a custom form. For a custom form, specify the following properties:</p> <ul style="list-style-type: none"> <li>Custom form endpoint: the location of the form.</li> <li>Custom form TLS profile: the TLS client profile that is used to secure the connection to the remote server.</li> </ul> <p>In the Dynamic table entries field, enter the name of a context variable that specifies the scopes that are to be added automatically to the authorization consent form.</p> <p>In the HTML form time limit field, specify the time allowed to submit the form.</p> <p>Disabled</p> <p>Authorization is disabled; this aspect of processing is skipped.</p> <p>Select Stop on error to halt assembly processing in the event of authorization failure.</p>	string

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Validate - DataPower API Gateway

Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.

## Gateway support

Note: This page describes the Validate policy implementation in the DataPower® API Gateway. If you are using the DataPower Gateway (v5 compatible), see [Validate - DataPower Gateway \(v5 compatible\)](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [validate - DataPower API Gateway](#).

## About

Position this policy where required in the assembly flow as follows:

- To validate the original input, position a Validate policy at the start of your flow.
- To validate an intermediate response that is returned from other invoke actions or tasks, position a Validate policy after those actions or tasks.
- To validate the response that is returned to the client application, position a Validate policy after the task that collates the response.

Note: With the DataPower API Gateway, the input to the Validate policy must be parsed data. One way to produce parsed data is to use a [Parse](#) policy before a Validate policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Validate policy properties

Property label	Required	Description	Data type
----------------	----------	-------------	-----------

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <b>validate</b> .	string
Description	No	A description of the policy.	string
Input	No	Specifies the name of a variable in the API context. The content of the <b>body</b> field of the variable, which is represented by <b>variable_name.body</b> , is the input data to validate. By default, the variable name is <b>message</b> .	string
Output	No	Specifies the name of a variable in the API context. <ul style="list-style-type: none"> <li>If the validation passes, the body field of the output variable, which is represented by <b>variable_name.body</b>, stores the output of the assembly validate action.</li> <li>If the schema to validate is a JSON schema, the validation also adds any default values that are missing from the payload.</li> <li>If the validation fails, no output is stored.</li> <li>If an output variable is not specified, the results of the assembly validate action are not saved.</li> </ul>	string
Validate against	Yes	Specifies the schema to be used for validating the payload. Select one of the following values: <ul style="list-style-type: none"> <li><b>definition</b>: Select this value if you want to use a previously defined schema to validate the payload that is returned from other invoke actions or tasks in the assembly flow. From the Definition Object list, select the required schema. For information on defining a schema for an API definition, see <a href="#">Editing an API definition</a>.</li> <li><b>url</b>: the schema is identified by a URL location. <ul style="list-style-type: none"> <li>In the JSON Schema URL field, enter the URL of the JSON schema to be used for validating a JSON payload.</li> <li>From the XML Validation Mode list, select how XML validation is to be performed: <ul style="list-style-type: none"> <li><b>None</b>: no XML payload validation is required.</li> <li><b>xsd</b>: an XML schema will be used to validate an XML payload. In the XML Schema URL field, enter the URL of the XML schema.</li> <li><b>wSDL</b>: a WSDL schema will be used to validate a SOAP payload. In the WSDL URL field, enter the URL of the WSDL schema.</li> <li><b>soap-body</b>: validate the body of a SOAP message against the XML schema only.</li> </ul> </li> </ul> </li> <li><b>wSDL</b>: use the XML schema in the WSDL file associated with the API operation or the API path.</li> <li><b>body-param</b>: validate the request input against the schema definition that is specified in the TYPE field for the request parameter for this operation. For information about how to create a request parameter, see <a href="#">Configuring an operation</a>.</li> <li><b>response-param</b>: validate the response to be returned to the client application, against the schema definition that is specified in the SCHEMA field for the response parameter for this operation. For information about how to create a response parameter, see <a href="#">Configuring an operation</a>.</li> </ul>	string

For examples, see [validate - DataPower API Gateway](#).

## Related tasks

- [Defining Paths for a REST API](#)
- [Creating a new REST OpenAPI definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible)

## Validate - DataPower Gateway (v5 compatible)

Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.

## Gateway support

Note: This topic describes the Validate policy implementation in the DataPower® Gateway (v5 compatible). If you are using the DataPower API Gateway, see [Validate - DataPower API Gateway](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [validate - DataPower Gateway \(v5 compatible\)](#).

Restriction:

- The schema that represents the XML can reference only one XML namespace.



- The schema cannot reference polymorphic XML elements.
- The validation works on the `message.body` variable and not any other output/context variable. If the invoke policy contains a configured response object variable, then `message.body` is not set, and validate is not able to act.
- If you use the `multipleOf` keyword in a schema definition for the API then, due to rounding behavior, the specified value must satisfy the following conditions, otherwise the validation fails when the API is called:
  - The value must not be less than `0.000009999999999999999848869`.
  - If the value is greater than 1, the amount before the decimal point must not be greater than `99999999999999934463`.

## About

You can attach this policy to the following API flow:

- REST

Position this policy where required in the assembly flow as follows:

- To validate the original input, position a Validate policy at the start of your flow.
- To validate an intermediate response that is returned from other invoke actions or tasks, position a Validate policy after those actions or tasks.
- To validate the response that is returned to the client application, position a Validate policy after the task that collates the response.

You can apply a different OpenAPI schema definition to each Validate policy either by choosing from the set of schema definitions that is specified at the API level, or the schema definition at the operation level.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Validate policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>validate</code> .	string
Description	No	A description of the policy.	string
Definition	Yes	Specifies the schema definition to be used for validating the payload. Valid values: <ul style="list-style-type: none"> <li>• <b>request</b>: Select this value to validate the request input against the schema definition that is specified in the TYPE field for the request parameter for this operation. For information about how to create a request parameter, see <a href="#">Configuring an operation</a>.</li> <li>• <b>response</b>: Select this value to validate the response to be returned to the client application, against the schema definition that is specified in the SCHEMA field for the response parameter for this operation. For information about how to create a response parameter, see <a href="#">Configuring an operation</a>.</li> <li>• <b>#/definitions/definition_name</b>: Select this value for a previously defined schema definition to be used to validate the payload that is returned from other invoke actions or tasks in the assembly flow. For information on defining a schema for an API definition, see <a href="#">Editing an API definition</a>.</li> </ul>	string

## Related tasks

- [Defining Paths for a REST API](#)
- [Creating a new REST OpenAPI definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Validate Username Token

Use the Validate Username Token policy to validate a Web Services Security (WS-Security) UsernameToken in a SOAP payload before allowing access to the protected resource.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
	1.1.0 (IBM API Connect® Version 2018.4.1.4 or later)

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [validate-username-token](#).

## About

A WS-Security UsernameToken enables a user identity to be passed securely over a multi-point message path. The Validate Username Token policy extracts the UsernameToken element from the request payload, authenticates the extracted username and password, and provides access to the protected resource based on the authentication result. The policy has two authentication methods: Lightweight Directory Access Protocol (LDAP) user registry, or Authentication URL.

The Validate Username Token policy supports both passwordText and passwordDigest types of password. When the authentication method is Authentication URL with a passwordDigest, a basic authentication header that contains a Base64 encoded username and passwordDigest is sent to the URL. In addition, a custom header named X-IBM-PasswordType is set with a value of digest. The following table shows the authentication process based on password type:

Table 2. Authentication URL process by password type

passwordText	passwordDigest
Authentication: Basic base64(username:password)	Authentication: Basic base64(username:passwordDigest) X-IBM-PasswordType: 'digest'

You can attach this policy to the following API flows:

- REST
- SOAP

Position this policy where required in the assembly flow as follows:

- To validate the original input, position a Validate Username Token policy at the start of your flow.
- To validate an intermediate response that is returned from other invoke actions or tasks, position a Validate Username Token policy after those actions or tasks.
- To validate the response that is returned to the client application, position a Validate Username Token policy after the task that collates the response.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 3. Validate Username Token policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <b>validate-usernameToken</b> .	string
Description	No	A description of the policy.	string
Authentication type (policy version 1.0.0 only)	Yes	The authentication type to use to validate the UsernameToken. Valid values: <ul style="list-style-type: none"> <li>• <b>Authentication URL</b>: Select this value to validate the user credentials against an authentication URL.</li> <li>• <b>LDAP registry</b>: Select this value to validate the user credentials against an LDAP user registry.</li> </ul> The default value is: <b>Authentication URL</b> .	string
Authentication URL (policy version 1.0.0 only)	Yes	The authentication URL to use to validate the UsernameToken user credentials against. Note: This property is required only if Authentication type is set to <b>Authentication URL</b> .	string
TLS profile (policy version 1.0.0 only)	No	The TLS profile to use for the secure transmission of data to the authentication URL. Note: This property is available only if Authentication type is set to <b>Authentication URL</b> .	string
LDAP registry name (policy version 1.0.0 only)	Yes	The name of the LDAP user registry to validate the UsernameToken user credentials against. You can select a name from the drop-down list, or type a name manually. Note: This property is required only if Authentication type is set to <b>LDAP registry</b> .	string
User Registry Name (policy version 1.1.0 and later)	Yes	Select the LDAP or Authentication URL registry to use to validate the UsernameToken.	string
LDAP search attribute <sup>1</sup>	Yes	The name of the LDAP user password attribute. Note: This property is required only for an LDAP user registry.	string

## Examples

The following example shows an LDAP user registry authentication:

```
- validate-usernameToken:
  version: 1.0.0
  title: "validate-usernameToken"
  auth-type: "LDAP Registry"
  ldap-registry: "wstest"
  ldap-search-attribute: "userPassword"
```

The following example shows an Authentication URL definition:

```
- validate-usernameToken:
  version: 1.0.0
  title: "validate-usernameToken"
  auth-type: "Authentication URL"
  auth-url: "https://www.google.com"
  tls-profile: "default-ssl-profile"
```

## Errors

The policy returns an HTTP 200 status code when successful, and the input payload is copied to the output flow. For all failure types the policy returns an HTTP 500 status code, and the output contains the SOAP fault.

Tip: If there are authentication failures, try verifying the LDAP user registry configuration as follows:

- Ensure Search (DN) is set as the communication method.
- Ensure Authenticated Bind is set so that specific permissions are required to search the registry.
- Ensure the Admin DN and Password fields are correctly completed for the Distinguished Name (DN) of a user authorized to carry out searches in the LDAP directory.
- Ensure that a combination of Base DN, Prefix, and Suffix are set, such that they fully describe the user DN. For example:
  - For a user named: `cn=alice, dc=ibm, dc=com`

```
BaseDN: dc=ibm
Prefix: cn=alice
Suffix: dc=com
```

where the user DN is calculated as: Prefix + BaseDN + Suffix.

## Related concepts

- [LDAP authentication](#)

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related information

- [Creating an LDAP registry](#)
- [Authenticating by using your enterprise user registry](#)

<sup>1</sup> When authenticating with LDAP and `passwordText`, the policy uses the username and password as LDAP bind credentials. However, when authenticating with LDAP and `passwordDigest`, the digest itself cannot be used for authentication. Instead, an LDAP search for the username is performed by using the administrator's distinguished name (DN) and password, and an attribute corresponding to the contents of the `ldap-search-attribute` is retrieved. A hash of the contents of this attribute (along with the `Nonce` and `Created` attributes, as in the WS-Security UsernameToken profile specification) is then compared to the `passwordDigest`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## XML to JSON

Use the XML to JSON policy to convert the context payload of your API from the extensible markup language (XML) format to JavaScript Object Notation (JSON).

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [xml-to-json](#).

## About

The XML to JSON policy uses a simple convention, based on BadgerFish, to convert your API context payload from XML to JSON. The XML content is preserved, including the attributes and namespaces. No additional configuration is required. For more information about the BadgerFish convention, including some examples, see [BadgerFish](#).

You can attach this policy to the following API flows:

- REST
- SOAP

Use the API Manager assembly view when you are creating your API definition to add a built-in policy to the flow.

The policy must be attached to the flow at the point at which you require the conversion to be performed. For example, if you need to convert an XML-formatted request into a JSON-formatted request, the policy must be attached to the request flow.

The policy reads input from the `message.body`, if that context exists, otherwise from the `request.body`, and then writes the output to the `message.body`.

**Note:** If you are using the DataPower API Gateway, the input to the XML to JSON policy must be parsed data. One way to produce parsed data is to use a [Parse](#) policy before an XML to JSON policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. Policy properties

Property label	Required	Description	Data type
----------------	----------	-------------	-----------

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>xml-to-json</code> .	string
Description	No	A description of the policy.	string
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Input	No	The input message to convert. Specify the name of a variable in the API context. <code>variableName.body</code> , the message payload, represents the XML input to convert. The default value of the variable is <code>message</code> and <code>message.body</code> is the default input.	string
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Output	No	The output message to store the conversion result. Specify the name of a variable in the API context. <code>variableName.body</code> represents the result of conversion from XML format to JSON format. When the specified input message is the default message, the default output is <code>message.body</code> . Otherwise, when the input message is the variable <code>my-message-variable</code> , for example, the default output is <code>my-message-variable.body</code> . The variable cannot be any read-only in the API context.	string
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Conversion type	No	The conversion type that determines the target format of the output. The following options are available: <ul style="list-style-type: none"> <li>badgerFish: BadgerFish convention is used to determine the target conversion format of the output.</li> <li>apicv5: apicv5 convention is used to determine the target conversion format of the output.</li> </ul>	string

## Examples

For example, the following simple XML object

```
<a>hello</a>
```

becomes

```
{ "a": { "$" : "hello" } }
```

The following XML object with an attribute

```
<a type="world">hello</a>
```

becomes

```
{ "a": { "$" : "hello", "@type" : "world" } }
```

For examples, see [xml-to-json](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## XSLT

Use the XSLT policy to apply an XSLT transform to the payload of the API definition.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [xslt](#).

## About

You can attach this policy to the following API flows:

- REST
- SOAP

Note: If you are using the DataPower API Gateway, the input to the XSLT policy must be parsed data. One way to produce parsed data is to use a [Parse](#) policy before an XSLT policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table lists the policy properties, indicates whether a property is required, specifies the valid and default values for input, and specifies the data type of the values.

Table 2. XSLT policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is <code>xml</code> .	string

Property label	Required	Description	Data type
Description	No	A description of the policy.	string
Use context current payload	No	Indicates whether this XSLT input document uses the context current payload, or if there is no input. The check box is cleared by default, which indicates that there is no input.	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Serialize output	No	If you select this option, the output tree that is generated by the XSLT policy is serialized. The content of <code>message.body</code> is updated with the serialized binary data rather than the XML tree. The check box is cleared by default, which indicates that the output tree is not serialized.	boolean
Source	Yes	The XSLT transform source to execute.	string

For examples of the OpenAPI definitions of XSLT policies, see [XSLT policy examples](#).

For more examples of how to use XSLT to access and modify properties and context, see [Implementation code examples](#) and, if you are using the DataPower API Gateway, [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

## Errors

The following error can be thrown while the policy is being executed:

- **TransformError** - a generic error that captures all errors that occur during the execution of the policy.

## Related concepts

- [Variable references in API Connect](#)

## Related tasks

- [Creating a new REST OpenAPI definition](#)

## Related reference

- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## XSLT policy examples

Examples of the OpenAPI definitions of XSLT policies.

- [Simple example with no context current payload](#)
- [Concatenation and transformation](#)
- [Obtain query parameter values and refer to context variables](#)

Note: The XML specification <https://www.w3.org/TR/xml/> does not specify a preferred order for XML namespace (XMLNS) attributes. Best practice is to not rely upon the sequence of XMLNS attributes if you write custom parsing code.

## Simple example with no context current payload

The following is an example of where the XSLT input document does not use the context current payload (there is no input):

```
- xslt:
  title: example xslt
  source: |
    <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
      <xsl:template match="/">
        <Hello>World!</Hello>
      </xsl:template>
    </xsl:stylesheet>
```

## Concatenation and transformation

The following example shows a more complex XSLT transform source, where the stylesheet concatenates two input strings and transforms the third input string to the IP address of the client:

```
- xslt:
  title: xslt
  input: true
  source: |
    <?xml version="1.0" encoding="UTF-8"?>
    <xsl:stylesheet
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
      xmlns:xalan="http://xml.apache.org/xslt"
      xmlns:fn="http://www.w3.org/2005/xpath-functions"
      xmlns:dp="http://www.datapower.com/extensions"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```

xmlns:xs4xs="http://www.w3.org/2001/XMLSchema"
xmlns:io="http://xformMessage"
xmlns:map="http://xformMessage/xform"
xmlns:mssl="http://www.ibm.com/xmlmap"
exclude-result-prefixes="fn dp dp map xalan mssl"
version="1.0">
<xsl:output method="xml" encoding="UTF-8" indent="no"/>

<!-- root wrapper template -->
<xsl:template match="/">
  <mssl:datamap>
    <xsl:choose>
      <xsl:when test="not(mssl:datamap/dataObject[1]/@xsi:nil)">
        <xsl:element name="dataObject">
          <xsl:attribute name="xsi:type">
            <xsl:value-of select="'io:data'"/>
          </xsl:attribute>
          <xsl:call-template name="map:xform">
            <xsl:with-param name="data" select="mssl:datamap/dataObject[1]"/>
          </xsl:call-template>
        </xsl:element>
      </xsl:when>
      <xsl:otherwise>
        <xsl:element name="dataObject">
          <xsl:attribute name="xsi:type">
            <xsl:value-of select="'io:data'"/>
          </xsl:attribute>
          <xsl:attribute name="xsi:nil">
            <xsl:text>true</xsl:text>
          </xsl:attribute>
        </xsl:element>
      </xsl:otherwise>
    </xsl:choose>
  </mssl:datamap>
</xsl:template>

<!-- This rule represents a type mapping: "data" to "io:data". -->
<xsl:template name="map:xform">
  <xsl:param name="data"/>
  <!-- a simple data mapping: "$data/StringOne" (string) to "StringOne" (string) -->
  <xsl:if test="$data/StringOne">
    <StringOne>
      <xsl:value-of select="concat($data/StringOne, $data/StringTwo)"/>
    </StringOne>
  </xsl:if>
  <!-- a simple mapping with no associated source: to "StringTwo" (string) -->
  <StringTwo>
    <xsl:value-of select="dp:client-ip-addr()"/>
  </StringTwo>
  <!-- a simple data mapping: "$data/NumberOne" (int) to "NumberOne" (int) -->
  <xsl:if test="$data/NumberOne">
    <NumberOne>
      <xsl:value-of select="$data/NumberOne"/>
    </NumberOne>
  </xsl:if>
  <!-- a simple data mapping: "$data/NumberTwo" (int) to "NumberTwo" (int) -->
  <xsl:if test="$data/NumberTwo">
    <NumberTwo>
      <xsl:value-of select="$data/NumberTwo"/>
    </NumberTwo>
  </xsl:if>
  <!-- a simple data mapping: "$data/NumberThree" (int) to "NumberThree" (int) -->
  <xsl:if test="$data/NumberThree">
    <NumberThree>
      <xsl:value-of select="$data/NumberThree"/>
    </NumberThree>
  </xsl:if>
</xsl:template>

<!-- ***** Utility Templates ***** -->
<!-- copy the namespace declarations from the source to the target -->
<xsl:template name="copyNamespaceDeclarations">
  <xsl:param name="root"/>
  <xsl:for-each select="$root/namespace::*[not(name() = '')]">
    <xsl:copy/>
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

## Obtain query parameter values and refer to context variables

The following example shows a complete OpenAPI source file. The API includes an XSLT policy that obtains a query parameter value in XSLT, and also uses the `getvariable` method to retrieve the value of the context variable `request.headers.user-agent`.

```

swagger: '2.0'
info:
  x-ibm-name: xslt
  title: xslt
  version: 1.0.0
schemes:
  - https
host: $(catalog.host)
basePath: /xslt
consumes:

```

```

- application/json
produces:
- application/json
securityDefinitions:
  clientIdHeader:
    type: apiKey
    in: header
    name: X-IBM-Client-Id
security:
- clientIdHeader: []
x-ibm-configuration:
  testable: true
  enforced: true
  cors:
    enabled: true
  assembly:
    execute:
      - operation-switch:
          title: operation-switch
          case:
            - operations:
                - verb: get
                  path: /hello
                  execute:
                    - xslt:
                        title: SayHello
                        input: false
                        source: |
                          <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
                            <xsl:template match="/">
                              <xsl:element name="APIC">
                                <xsl:text>Hello World!</xsl:text>
                              </xsl:element>
                            </xsl:template>
                          </xsl:stylesheet>
            - operations:
                - verb: get
                  path: /getContextQueryVar
                  execute:
                    - xslt:
                        title: GetContextQueryVar
                        input: false
                        source: |
                          <xsl:stylesheet version="1.0"
                            xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                            xmlns:apim="http://www.ibm.com/apimanagement">
                            <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl"/>
                            <xsl:template match="/">
                              <xsl:call-template name="apim:output">
                                <xsl:with-param name="mediaType" select="'application/xml'"/>
                              </xsl:call-template>
                              <APIC>
                                <xsl:element name="apim:getVariable">
                                  <xsl:element name="useragent">
                                    <xsl:value-of select="apim:getVariable('request.headers.user-agent')"/>
                                  </xsl:element>
                                  <xsl:element name="query">
                                    <xsl:value-of select="apim:getVariable('request.querystring')"/>
                                  </xsl:element>
                                </xsl:element>
                              </APIC>
                            </xsl:template>
                          </xsl:stylesheet>
            - operations:
                - verb: get
                  path: /getQuery
                  execute: []
            otherwise:
                - throw: null
                  title: handling unknown operation
                  name: Unsupported
          catch:
            - errors:
                - Unsupported
            execute:
              - set-variable:
                  actions:
                    - set: message.body
                      value: '<error>Not Supported</error>'
          phase: realized
paths:
/hello:
  get:
    responses:
      '200':
        description: 200 OK
/getContextQueryVar:
  get:
    responses:
      '200':
        description: 200 OK
definitions: {}
tags: []

```

For more examples of how to use XSLT to access and modify properties and context, see [Implementation code examples](#) and, if you are using the DataPower® API Gateway, [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Implementation code examples

Example XSLT and GatewayScript code snippets.

Note: If you are using GatewayScript, you must include one or other of the following commands depending on your gateway type:

```
DataPower Gateway (v5 compatible) var apic = require('./apim.custom.js');
API Gateway var apic = require('apim');
```

where *apic* is the common name used for the GatewayScript examples in this topic. However, *apic* could be any given name of your choice, for example you could use:

```
var apim = require('./apim.custom.js');
```

and then you would start your calls with **apim**.

- [Access to input properties code snippet](#)
- [Access to runtime context code snippet](#)
- [Access to input payload code snippet](#)
- [Access to HTTP headers code snippet](#)
- [Modify the payload code snippet](#)
- [Configure error information code snippet](#)
- [Accessing the caught exception in a catch block](#)
- [Set variables code snippet](#)

## Access to input properties code snippet

The following code block shows an example of how to access the input properties by using the XSLT `policyProperties()` function. The example defines a property that is named `a_property`, which is declared as an integer value, but is retrieved in XSLT as text.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  DataPower Gateway (v5 compatible) <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  API Gateway <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="p" select="apim:policyProperties()" />
    <xsl:message>
      The value of my input property is
      <xsl:value-of select="$p/a_property" />
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>
```

If you are using GatewayScript, call the following function:

```
apic.getPolicyProperty(propertyName)
```

where *propertyName* is the name of the input property that you want to access. If the input property name is blank, the action will return all input properties.

## Access to runtime context code snippet

The following code block shows an example of how to access the runtime context by using the XSLT `getContext()` function.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  DataPower Gateway (v5 compatible) <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
```



```

    API Gateway <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xml" />

<xsl:template match="/">
  <xsl:variable name="client-id" select="apim:getContext('client.app.id') " />
  <xsl:message>
    The calling application is
    <xsl:value-of select="$client-id" />
  </xsl:message>
</xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, call the following function:

```
apic.getContext(varName)
```

where `varName` is the name of the context variable that you want to access.

For a complete list of context variables, see [API Gateway context variables](#). If you are using the DataPower® API Gateway, see also [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

## Access to input payload code snippet

The following code block shows an example of how to access the input payload by using the XSLT `payloadRead()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  DataPower Gateway (v5 compatible) <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xml" />
  API Gateway <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xml" />

  <xsl:template match="/">
    <xsl:variable name="input" select="apim:payloadRead()" />
    <xsl:message>
      The input payload is
      <xsl:copy-of select="$input" />
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, call the following function:

```
apic.readInput(callback)
```

A callback is required because the actual payload read is asynchronous. The callback method is called when the payload is ready.

This function returns an XML node-set that contains the payload of the request. If the payload is in JSON format, a JSONx node-set is returned that can then be manipulated within an XSLT or GatewayScript stylesheet. If the payload is not in JSON or XML format, the node-set that is returned is empty.

The following example shows how to use the `payloadType()` function to determine what type of payload (XML or JSONx) will be returned by the XSLT `payloadRead()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  DataPower Gateway (v5 compatible) <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xml" />
  API Gateway <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xml" />

  <xsl:template match="/">
    <xsl:variable name="payloadType" select="apim:payloadType()" />
    <xsl:message>
      <xsl:text>Payload type is [</xsl:text>
      <xsl:value-of select="$payloadType" />
      <xsl:text>]</xsl:text>
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

## Access to HTTP headers code snippet

The following code block shows an example of how to access the HTTP headers in XSLT by using the `getContext()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
    href="#" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  <xsl:include
    href="#" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="content-type" select="apim:getContext('request.headers.content-type')"/>
    <xsl:message>
      The request content type is
      <xsl:value-of select="$content-type" />
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, call the following function:

```
apic.getContext(request.headers.headerName)
```

where **headerName** maps to the name of the header you want to access.

Note: Access or modification of HTTP headers by using DataPower extensions, such as **dp:set-request-header**, is not advisable, as such actions might yield unexpected results when the policy is combined with other policies and assembly steps.

## Modify the payload code snippet

The output from a user-defined policy must be an XML node-set, which represents an XML or SOAP message, or a JSON message by using JSONx. The following code block shows an example of how to modify the payload in XSLT. To assist the API Gateway policy framework to accept the new or transformed message, call the **apim:output** template, as shown in the following example.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:apim="http://www.ibm.com/apimanagement"
  xmlns:jsonx="http://www.ibm.com/xmlns/prod/2009/jsonx">

  <!-- Contains the APIM functions -->
  <xsl:import
    href="#" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  <xsl:include
    href="#" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <!-- Creates a JSON document (empty object is for simplicity) -->
    <jsonx:object>
    </jsonx:object>

    <!-- Indicates the media type of the output being produced -->

    <xsl:call-template name="apim:output">
      <xsl:with-param name="mediaType" select="'application/json'"/>
    </xsl:call-template>
  </xsl:template>

</xsl:stylesheet>

```

where **mediaType**:

- 'application/json' is when the output is written in JSONx format.
- 'application/xml' is when the output is written in XML format.

If you are using GatewayScript, call the following function:

```
apic.output(mediaType)
```

where **mediaType** is:

- **application/json** is when the output is written in JSONx format.
- **application/xml** is when the output is written in XML format.

Specifying the media type allows the next steps in the assembly flow to understand how to process the new payload.

Tip: The output from a user-defined policy must be XML or JSONx. JSONx is an IBM standard format to represent JSON as XML. One way to convert output GatewayScript JSON data into JSONx, is to add a **Convert Query Params to XML** action to follow the GatewayScript action within the same policy rule. The **Convert Query Params to**

**XML** action must have an **Input Conversion** with the **Default**

**Encoding** set to **JSON**. The output from the GatewayScript action must be the input for the **Convert Query Params to XML** action for JSONx to be produced.

## Configure error information code snippet

The following XSLT code block shows an example of how to configure the policy implementation to produce error information by calling the `apim-error` template.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:apim="http://www.ibm.com/apimanagement">

  <!-- Contains the APIM functions -->
  <xsl:import
    href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  <xsl:include
    href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <!-- Indicates this policy has a failure and provides
    additional information for the client application -->
  <xsl:template match="/">
    <xsl:call-template name="apim:error">
      <xsl:with-param name="httpCode" select="'401'" />
      <xsl:with-param name="httpReasonPhrase" select="'Unauthorized'" />
      <xsl:with-param name="errorMessage" select="'Please select a Plan'" />
    </xsl:call-template>
  </xsl:template>
</xsl:stylesheet>
```

where:

- `httpCode` is the code of the required error message.
- `httpReasonPhrase` is the reason for the error.
- `errorMessage` is the suggested action for the user.

If you are using GatewayScript, call the following function:

```
apic.error(name, httpCode, httpReasonPhrase, message)
```

where:

- `name` is the name of the error.
- `httpCode` is the code of the required error message.
- `httpReasonPhrase` is the reason for the error.
- `message` is the suggested action for the user.

## Accessing the caught exception in a catch block

The following XSLT code block shows an example of how, in the `catch` block of an API assembly, you can obtain the details of the current caught exception. A possible use would be to create a custom error response using the details of the caught exception.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:apim="http://www.ibm.com/apimanagement"
  xmlns:apigw="http://www.ibm.com/xmlns/datapower/2017/11/apigateway"
  extension-element-prefixes="dp"
  exclude-result-prefixes="dp apim">
  <xsl:output method="xml" />

  <!-- Contains the APIM functions -->
  <xsl:import
    href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  <xsl:include
    href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="exception" select="apim:getError()" />
    <xsl:variable name="exception" select="apigw:get-error()" />
    <!-- output desired error message based on the exception -->
    <myError>
      <errorReason><xsl:value-of select="$exception/error/message"/></errorReason>
    </myError>

    <!-- Propagate the HTTP status code and reason phrase from the exception -->
    <xsl:call-template name="apim:setVariable">
      <xsl:with-param name="varName" select="'message.status.code'" />
      <xsl:with-param name="value" select="$exception/error/status/code"/>
      <xsl:with-param name="action" select="'Set'" />
    </xsl:call-template>

    <xsl:call-template name="apim:setVariable">
      <xsl:with-param name="varName" select="'message.status.reason'" />
    </xsl:call-template>
  </xsl:template>
</xsl:stylesheet>
```

```

    <xsl:with-param name="value" select="$exception/error/status/reason"/>
    <xsl:with-param name="action" select="'Set'" />
  </xsl:call-template>
</xsl:template>

```

```
</xsl:stylesheet>
```

The `apim:getError()` and `apigw:get-error()` functions return an XML node set; for example:

```

<?xml version="1.0" encoding="UTF-8"?>
<error>
  <name>RuntimeException</name>
  <message>This is a thrown Runtime Error</message>
  <policyTitle>Throw Runtime Error</policyTitle>
  <status>
    <code>500</code>
    <reason>Internal Server Error</reason>
  </status>
</error>

```

If you are using GatewayScript, call the following function:

```
apim.getError()
```

which returns a JSON object; for example:

```

{
  "name": "OperationError",
  "message": "This is a thrown Operation Error",
  "policyTitle": "Throw Operation Error",
  "status": {
    "code": "500",
    "reason": "Internal Server Error"
  }
}

```

## Set variables code snippet

The following XSLT code block shows an example of how to set a runtime variable to a specified string value by calling the `setVariable` template.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local_isp_policy_apim.custom.xsl" />
  <xsl:include href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:call-template name="apim:setVariable">
      <xsl:with-param name="varName" select="'serviceEndpoint'" />
      <xsl:with-param name="value" select="'https://endpoint.host.com/data'" />
    </xsl:call-template>
    <xsl:message>
      <xsl:text>Variable [ </xsl:text>
      <xsl:value-of select="'serviceEndpoint'" />
      <xsl:text>] set to [ </xsl:text>
      <xsl:value-of select="'https://endpoint.host.com/data'" />
      <xsl:text>] </xsl:text>
    </xsl:message>
  </xsl:template>
</xsl:stylesheet>

```

where:

- `varName` is the name of the runtime variable that you want to set a value to.
- `value` is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, you would specify the `value` as `$(request.headers.content-type)`.

If you are using GatewayScript, call the following function:

```
apic.setvariable(varName, varValue, action)
```

where:

- `varName` is the name of the runtime variable that you want to set a value to, or that you want to add or clear.
- `varValue` is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, you would specify the `varValue` as `request.headers.content-type`. This property is only required when `set` or `add` is specified as the action.
- `action` is the action that you want to apply to the variable. Valid options are:
  - `set`
  - `add`
  - `clear`

If no option is set, the default option of `set` is applied.

The following XSLT example shows how to retrieve the value of a runtime variable by using the `getVariable()` function.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanager" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <include href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_local:_isp_policy_apim.custom.xsl" />
  <include href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_store:_dp_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="varValue" select="apim:getVariable('serviceEndpoint')" />
    <xsl:message>
      <xsl:text>Variable [ </xsl:text>
      <xsl:value-of select="'serviceEndpoint'" />
      <xsl:text>] = [ </xsl:text>
      <xsl:value-of select="$varValue" />
      <xsl:text>] </xsl:text>
    </xsl:message>
  </xsl:template>
</xsl:stylesheet>
```

where

- `varValue` is the name of the runtime variable that you want to retrieve a value for.

If you are using GatewayScript, call the following function:

```
apic.getvariable(varName)
```

where `varName` is the name of the runtime variable that you want to retrieve a value for.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Logic Constructs

IBM® API Connect includes a number of logic constructs that you can use to apply preconfigured logic to an assembly to control the flow of data through your assembly when the API is called.

Logic constructs are configured in the context of an API. You can use the API Manager assembly editor to add a logic construct to an API and to configure the properties for that construct.

The logic constructs are described in the following subtopics:

- [if](#)  
Use the if construct to apply a section of the assembly when a condition is fulfilled.
- [operation-switch](#)  
Use the operation-switch component to apply a section of the assembly to a specific operation.
- [switch](#)  
Use the switch component to execute one of a number of sections of the assembly based on which specified condition is fulfilled.
- [throw](#)  
Use the throw policy to throw an error when it is reached during the execution of an assembly flow.

---

## Related concepts

- [The Assemble view](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## if

Use the if construct to apply a section of the assembly when a condition is fulfilled.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0, functionality provided by <a href="#">switch</a>	

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [if](#).

An if construct provides a way to branch an API's assembly when a specified condition is fulfilled. Each if construct contains a section of the assembly that is only executed when the script within the construct returns a **true** value.

When using the assemble view's property sheet, use the Condition field to write your condition that returns **true** or **false**.

If you want one or more policies or constructs to be executed when the condition of the if construct is fulfilled, drag the new policy or construct onto one of the dashed boxes that are displayed within the if construct. Constructs and policies included in the if construct are part of the case that is executed when the condition of the if construct is returned as true.

For information about the OpenAPI implementation of an if construct, see [if](#).

In the Condition field, use the form `apim.getvariable('context.location.variable')` to reference your variables, where *context* is the context that you want to reference, *location* is the location of the variable within that context, and *variable* is the name of the variable.

## Construct property details

You can configure a construct's properties in the property sheet in the assemble view.

Table 2. The properties of an if construct

Property	Required	Description
Title	No	A custom title for your construct when it is displayed in the canvas. If a title is not specified, <b>if</b> is used by default.
Description	No	A description of your construct, it is not displayed on the canvas.
Condition	Yes	Use GatewayScript to provide conditions. A list of context variables that you can use to generate conditions can be found in <a href="#">API Connect context variables</a> .

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## operation-switch

Use the operation-switch component to apply a section of the assembly to a specific operation.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0, functionality provided by <a href="#">switch</a>	

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [operation-switch](#).

An operation-switch component provides a way to branch an API's assembly depending on the operation that is called. For each case, a separate piece of the assembly is applied to the operations that belong to the case.

When using the assemble view's property sheet, click the Case field to view suggested operations. Type in the Case field to refine the list of suggested operations.

Each case behaves as an assembly. To add components to a case, drag the component from the palette to the area included in the operation-switch component. Dashed boxes are displayed where the component can be included in the case.

For information about the OpenAPI implementation of an operation-switch component, see [operation-switch](#).

Restriction: Nesting an operation-switch component inside an if or switch construct, or another operation-switch component, is **not** supported.

## Component property details

You can configure a component's properties in the property sheet in the assemble view.

Table 2. The properties of an operation-switch component

Property	Required	Description
Title	No	A custom title for your component when it is displayed in the canvas. If a title is not specified, <b>operation-switch</b> is used by default.
Description	No	A description of your component, it is not displayed on the canvas.
Case	Yes	Each case includes one or more operations to which the branch provided by the operation switch applies.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## switch

Use the switch component to execute one of a number of sections of the assembly based on which specified condition is fulfilled.

### Gateway support



Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [switch](#).

A switch construct provides a way to branch an assembly based on multiple conditions. Each switch component contains multiple cases, each corresponding to a section of the assembly that is only executed when the condition or operation specified by the case is met or used. Additionally, an otherwise case executes when no other case is fulfilled.


Add new cases by clicking + Case and add an "otherwise" case by clicking + Otherwise.

If multiple cases are fulfilled, the highest priority case will be executed. Change the priority of cases by clicking the Move up  and Move down  icons.

To configure a case to be executed if a specific operation is called, use the search operations field and select your operation from the list. You can refine the results of the search by typing in the search operations field.

To configure a case to be executed based on a GatewayScript condition, click edit condition and enter your script in the "Condition editor" window. When you have supplied a script, you can edit your script either in the Condition field or by clicking edit condition.

If you are using the DataPower Gateway (v5 compatible), you enter your condition as GatewayScript directly in a code area in the Condition editor. If you are using DataPower API Gateway, the Condition editor provides a script builder to help you construct your condition script; for more information, see [Using the switch policy condition editor](#).

To delete a case, click the Remove case icon .



If you want one or more policies or constructs to be executed when the condition of a case is fulfilled, drag the new policy or construct onto one of the dashed boxes that are displayed within the case's section of the switch construct.

Note: A switch case **must** contain at least one policy, otherwise the Gateway server returns an error.

### Construct property details

You can configure a construct's properties in the property sheet in the assemble view.

Table 2. The properties of a switch construct

Property	Required	Description
Title	No	A custom title for your construct when it is displayed in the canvas. If a title is not specified, <b>switch</b> is used by default.
Description	No	A description of your construct, it is not displayed on the canvas.
case	Yes (one or more)	Specify one or more operations or write a script for a condition.  Use the Condition editor to construct your condition script. See <a href="#">Using the switch policy condition editor</a> .  Use GatewayScript to provide conditions. You can reference variables by using the form <code>apim.getvariable('context.location.variable')</code> , where <i>context</i> is the context that you want to reference, <i>location</i> is the location of the variable within that context, and <i>variable</i> is the name of the variable. A list of context variables that you can use to generate conditions can be found in <a href="#">API Connect context variables</a> .
otherwise	No	Add an otherwise case if you want to execute a section of the assembly when no other cases are fulfilled.

- [Using the switch policy condition editor](#)

The switch policy condition editor provides a user interface to help you build the conditions for the **case** clauses in a switch policy.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Using the switch policy condition editor

The switch policy condition editor provides a user interface to help you build the conditions for the `case` clauses in a switch policy.

You build a condition script by using the Add Group and Add Condition options, together with operator selections, and pre-supplied function selections. The Output field shows the resultant condition script, and updates dynamically as you build your condition.

### Adding conditions

When you open the policy condition editor, an initial condition is pre-supplied ready for you to configure.

Use the drop-down list to select the function that you want to begin your condition.

The following JSONata functions are supported:

- Aggregation functions:
  - `$average(array)`
  - `$max(array)`
  - `$min(array)`
  - `$sum(array)`
- Array functions:
  - `$append(array1, array2)`
  - `$count(array)`
  - `$reverse(array)`
  - `$sort(array [, function])`
  - `$zip(array1, ...)`
- Boolean functions:
  - `$boolean(arg)`
  - `$exists(arg)`
  - `$not(arg)`
- Numeric functions:
  - `$abs(number)`
  - `$ceil(number)`
  - `$floor(number)`
  - `$formatBase(number [, radix])`
  - `$number(arg)`
  - `$power(base, exponent)`
  - `$round(number [, precision])`
  - `$sqrt(number)`
- Object functions:
  - `$keys(object)`
  - `$lookup(object, key)`
  - `$merge(array<object>)`
  - `$spread(object)`
- String functions:
  - `$contains(str, pattern)`
  - `$join(array[, separator])`
  - `$length(str)`
  - `$lowercase(str)`
  - `$match(str, pattern [, limit])`
  - `$pad(str, width [, char])`
  - `$replace(str, pattern, replacement [, limit])`
  - `$split(str, separator [, limit])`
  - `$substring(str, start[, length])`
  - `$substringAfter(str, chars)`
  - `$substringBefore(str, chars)`
  - `$trim(str)`
  - `$uppercase(str)`

You can use the following JSONata numeric operators:

- `+` (addition)
- `-` (subtraction)
- `*` (multiplication)
- `/` (division)
- `%` (modulo)

You can use the following JSONata comparison operators for number values or strings:

- `=`
- `!=`
- `<`
- `>`
- `<=`
- `>=`

You can also use the following functional extensions to standard JSONata notation. Each extension corresponds to a part of the API context.

Extension	Variable	Description
-----------	----------	-------------



Extension	Variable	Description
<code>\$header (name)</code>	<code>message.headers.name</code>	Message header
<code>\$httpVerb ()</code>	<code>request.verb</code>	HTTP method of the request
<code>\$operationID ()</code>	<code>api.operation.id</code>	ID of the operation
<code>\$operationPath ()</code>	<code>api.operation.path</code>	Path of the operation
<code>\$queryParameter ('name')</code>	<ul style="list-style-type: none"> <li><code>request.parameters.name.locations</code></li> <li><code>request.parameters.name.values</code></li> </ul>	Returns the value of a request query parameter given the parameter name. The function searches the <code>request.parameters.name.locations</code> array for the index position of the value <code>query</code> , and returns <code>request.parameters.name.values [index]</code> , where <code>index</code> is the index position. Parameter values are not URL decoded.
<code>\$statusCode ()</code>	<code>message.status.code</code>	Status code
<code>\$storageType ([arg])</code>	<p><code>variable.body</code></p> <p>You can specify any variable in the API context. When no variable is specified, the default variable <code>message.body</code> is used.</p>	<p>Storage type of the message. The supported values are:</p> <ul style="list-style-type: none"> <li><code>binary</code></li> <li><code>json</code></li> <li><code>stream</code></li> <li><code>xml</code></li> </ul>
<code>\$urlParameter ('name')</code>	<ul style="list-style-type: none"> <li><code>request.parameters.name.locations</code></li> <li><code>request.parameters.name.values</code></li> </ul>	<p>Given a request parameter name, returns the parameter values for all occurrences of that parameter as a path parameter or query parameter.</p> <p>The function searches the <code>request.parameters.name.locations</code> array for the index positions of the values <code>path</code> and <code>query</code>, and returns, in a single array, the <code>request.parameters.name.values [index]</code> values for all identified index positions. When the URL contains both path and query parameter values, the array includes the path values first followed by the query values. The values of each parameter type are added in the order that they are received. Parameter values are URL decoded.</p> <p>For example, the following URL contains both path and query parameter values:</p> <pre>http://example.com/petstore/cats/adopt?breed=Sphynx&amp;breed=Siamese</pre> <p>The function call <code>\$urlParameter ('breed')</code> returns the following array of values:</p> <pre>[cats, adopt, Sphynx, Siamese]</pre> <p>In this example, the URL includes an API path that is configured as <code>/petstore/{breed}/{breed}</code>, where <code>breed</code> is configured to be a path parameter of the API path. As a result, <code>cats</code> and <code>adopt</code> are included in the output.</p>
<code>\$xpath (path, xpathExpression)</code>	You can specify any writable variable in the API context. The <code>xpathExpression</code> must be a literal string.	Allows use of XPath expressions

After you select the function, additional fields are displayed, depending on the selected function, so that you can complete the condition. For example, if you select the function `$httpVerb ()`, a comparison operator selection list is displayed (= or !=), together with a list of HTTP verbs to select on the right hand side of the expression (`GET`, `PUT`, `POST`, `DELETE`, `HEAD`, `OPTIONS`, or `PATCH`).

If you select NOT, your condition is negated with a `$not` function.

To add further conditions, click Add Condition; you then select whether to insert an `and` or an `or` operator between this condition and the preceding one.

To group two or more conditions inside parentheses, click Add Group, then add your conditions to the group; you then select whether to insert an `and` or an `or` operator between this group and the preceding condition or group.

To write your own condition from scratch, select Custom, then enter your script in the field provided.

## Example

To build the following condition:

```
($statusCode () != 200 and ($httpVerb () = 'GET' or $httpVerb () = 'PUT'))
```

complete these steps in the condition editor:

1. Select `$statusCode ()`, select != for the comparison operator, and enter `200` in the field on the right hand side of the condition.
2. Click Add Group.
3. In the new group sub pane that is displayed, click Add Condition.
4. Select `httpVerb ()`, leave the comparison operator as =, and select `GET` on the right hand side of the condition.
5. Click Add Condition in the group sub pane again.
6. Select `httpVerb ()`, leave the comparison operator as =, and select `PUT` on the right hand side of the condition.
7. For the comparison operator between the two conditions, select `or`.

## Simple condition statements

The following examples show conditions that use a single function.

This example uses the `$httpVerb ()` extension to specify the HTTP method of the request.

```
$httpVerb ()="GET"
```

This example uses the `$operationPath()` extension to specify the path of the operation.

```
$operationPath()="/base/path-2"
```

This example uses the `$operationID()` extension to specify the operation ID.

```
$operationID()="test-gatewayscript-GET"
```

This example uses the `$statusCode()` extension to specify the message status code.

```
$statusCode()=200
```

This example uses the `$header(name)` extension to specify the content type of the message header.

```
$header("Content-Type")="application/json"
```

## Combining conditions with logical operators

You can use `and` or `or` operators to combine multiple functions in a single condition.

This example specifies an HTTP GET request and an API operation path equal to `test-gatewayscript-GET`.

```
$httpVerb()="GET" and $operationPath()="test-gatewayscript-GET"
```

This example specifies either a `POST` or `PUT` request.

```
$httpVerb()="POST" or $httpVerb()="PUT"
```

This example specifies an API operation ID equal to `test-gatewayscript_POST` and a message status code equal to `200`, or an API operation ID equal to `test-gatewayscript-GET` and a message status code equal to `500`.

```
($operationID()="test-gatewayscript-POST" and $statusCode()=200) or ($operationID()="test-gatewayscript-GET" and $statusCode()=500)
```

This example specifies an API operation ID equal to `test-gatewayscript-POST` and a message status code equal to `200` with an API operation path equal to `/base/path-2`.

```
($operationID()="test-gatewayscript-POST" and $statusCode()=200) and $operationPath()="/base/path-2"
```

This example specifies a message header content type equal to `text/plain` and a message header length equal to `300`, or a message status code equal to `200`.

```
($header("Content-Type")="text/plain" and $header("Content-Length")=300) or $statusCode()=200
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## throw

Use the throw policy to throw an error when it is reached during the execution of an assembly flow.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in the assembly user interface; for details on how to configure the policy in your OpenAPI source, see [throw](#).

When the throw policy is encountered, the specified error and error message is produced.

If a catch has been configured that the error produced by the throw policy fulfills, the catch will be triggered.

If no catch is triggered by the thrown error, then a `500 Internal Server Error` is returned to the API caller.

## Component property details

You can configure a component's properties in the property sheet in the assemble view.

Table 2. The properties of a throw component

Property	Required	Description
Title	No	A custom title for your component when it is displayed in the canvas. If a title is not specified, <code>throw</code> is used by default.
Name	Yes	The error name that is thrown by the policy.
Message	Yes	The error message that is returned with the error name.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway

In the code for GatewayScript and XSLT policies in your API assemblies, you can use API context variables to work with the messages that are generated when an API is called.

### Introduction

All API context variables are stored in a JSON-like tree; each context variable is a node in this tree. A non-leaf node can be either an object or an array, while a leaf node can be a number, string, boolean, or document; a document can contain an XML document, BLOB, Stream, or another JSON document.

Each context variable has a name and a value. You can access a context variable by name when using the GatewayScript and XSLT APIs. Dot notation is used to access a context variable, reflecting the position of the corresponding node in the JSON-like tree; for example, `request.headers.Host`. Where a node name contains special characters, bracket notation is supported; for example, `request.headers['Content-Type']`.

For the GatewayScript policy, all the APIs for the DataPower® API Gateway are in the `context` module, which is a global module. You can therefore access the `context` module directly, you do not need to use the `require()` function first.

For the XSLT policy, the APIs are in the `http://www.ibm.com/xmlns/datapower/2017/11/apigateway` namespace; to use these APIs you must declare this namespace.

Note: The mechanisms described here for interacting with the API Connect context are preferred if you are writing new GatewayScript policy, XSLT policy, or user-defined policy code, and provide the best performance. For mechanisms that are compatible with Version 5, see [GatewayScript code examples](#).

### Accessing context variables - examples

#### GatewayScript examples

To read the `client.app.id` context variable:

```
var clientID = context.get('client.app.id');
```

To set the value of a context variable:

```
context.set('my.vars.amount', 100);
```

The resulting tree has the following structure:

```
{ "my": { "vars": { "amount": 100 } } }
```

To delete the node `my.vars`, and all its child nodes:

```
context.clear('my.vars');
```

The resulting tree has the following structure:

```
{ "my": { } }
```

#### XSLT examples

To read the `client.app.id` context variable:

```
<xsl:variable name="clientID" select="apigw:get-variable('client.app.id')"/>
```

To set the value of a context variable:

```
<apigw:set-variable name="my.vars.amount" value="100"/>
```

The resulting tree has the following structure:

```
{ "my": { "vars": { "amount": 100 } } }
```

To delete the node `my.vars`, and all its child nodes:

```
<apigw:clear-variable name="my.vars"/>
```

The resulting tree has the following structure:

```
{ "my": { } }
```

### About payloads

When the assembly flow for an API call begins, the following two message objects are predefined in the API context tree:

- **request**: holds the information for the original payload; the **request** message is a read-only object.
- **message**: holds the information for the current payload.

Typically, a message object has **headers** and **body** properties. For the current payload, the **message** object might have **variables** and **status** properties, as well as other properties created during the execution of the assembly.

The following example shows the general structure of a typical message:

```

{
  "headers": {
    "Content-Type": "application/xml",
    "X-Client-ID": "1234567890"
  },
  "variables": {
    "amount": 123,
    "name": "xyz"
  },
  "status": {
    "code": 200,
    "reason": "OK"
  },
  "body": <XML/Blob/Stream/JSON Document>
}

```

## Manipulating the message headers - examples

Note: Do not attempt to use this method to add fields to Analytics data. Using an [Analytics filter](#) is the only supported way to add top-level Analytics fields.

GatewayScript examples

To delete a header:

```
context.message.header.remove('X-My-Header');
```

To read a header:

```
var reqType = context.request.header.get('Content-Type');
var currType = context.message.header.get('Content-Type');
```

To set a header:

```
context.message.header.set('Content-Type', 'application/json');
```

To iterate through the headers:

```
var reqHdrs = context.request.headers;
for (var h in reqHdrs) {
  console.log('>>> Reading the request header: ', h, '=', reqHdr[h]);
}

var currHdrs = context.message.headers;
for (var h in currHdrs) {
  console.log('>>> Reading the current header: ', h, '=', currHdr[h]);
}

```

To create a message context called `headersContext`, if one doesn't already exist, and add to it a boolean context variable called `HelloWorld`, with the value `true`:

```
var ctx = context.getMessage("headersContext") || context.createMessage("headersContext");
ctx.setVariable('HelloWorld', true);
```

The `setVariable` call writes the variable into `headersContext.variables>HelloWorld`; therefore a condition statement in any subsequent `switch` policy should check for `headersContext.variables>HelloWorld` rather than `headersContext>HelloWorld`.

To read the context variable in any subsequent `gatewayscript` policy, use the following call:

```
ctx.getVariable("HelloWorld")
```

Note: If you are either migrating an API from IBM® API Connect Version 5.0, or from the DataPower Gateway (v5 compatible), to the DataPower API Gateway, or you are configuring a new API for the DataPower API Gateway, creating a message context, rather than using the context variable mechanism described in [Accessing context variables - examples](#), is the preferred option. In the case of migration, it most closely matches the behavior of IBM API Connect Version 5.0, and `setVariable` and `getVariable` calls will continue to work without modification. Creating a message context is also the preferred option if you are creating a new API for the DataPower API Gateway because it most closely aligns the API context with the way that various native DataPower API Gateway policies process it.

XSLT examples

To delete a header:

```
<apigw:clear-header message="'message'" name="'X-My-Header'"/>
```

To read a header:

```
<xsl:variable name="reqType" select="apigw:get-header('request', 'Content-Type')"/>
<xsl:variable name="currType" select="apigw:get-header('message', 'Content-Type')"/>
```

To set a header:

```
<apigw:set-header name="'Content-Type'" value="'text/xml'"/>
```

To get all headers:

```
<xsl:variable name="allReqHdrs" select="apigw:all-headers('request')"/>
<xsl:variable name="allCurrHdrs" select="apigw:all-headers('message')"/>
```

## Manipulating the message status - examples

Note: Only the `message` object has the status property, not the `request` object.

GatewayScript examples

To read the status code and reason phrase:

```
var code = context.message.statusCode;
var reason = context.message.reasonPhrase;
```

To update the status code:

```
context.message.statusCode = 400; //integer, update the code, with default reason
```

To update the status code and reason phrase in a single operation:

```
context.message.statusCode = '200 Wonderful'; //string, update code along with customized reason
```

Note: The `context.message.reasonPhrase` context variable is read-only.

XSLT examples

To read the status code and reason phrase:

```
<xsl:variable name="code" select="apigw:get-variable('message.status.code')"/>
<xsl:variable name="reason" select="apigw:get-variable('message.status.reason')"/>

<apigw:set-variable name="message.status.code" value="200"/>
<apigw:set-variable name="message.status.reason" value="OK"/>
```

## Accessing the message body - examples

---

Note: The request body is read-only.

GatewayScript examples

To read the request body:

```
context.request.body.readAsJSON(function(error, json) {});
context.request.body.readAsXML(function(error, xml) {});
context.request.body.readAsBuffer(function(error, buffer) {});
```

To read the message body:

```
context.message.body.readAsJSON(function(error, json) {});
context.message.body.readAsXML(function(error, xml) {});
context.message.body.readAsBuffer(function(error, buffer) {});
```

To write the message body and update the content type accordingly:

```
// text response
context.message.body.write("Hello world.");
context.message.header.set('Content-Type', "text/plain");

// JSON response
var pet = { "kind": "dog", "age": 3, "color": "black" };
context.message.body.write(pet);
context.message.header.set('Content-Type', "application/json");

// XML response
var resp = "<order><id>100</id><timestamp>20181231</timestamp><status>fulfilled</status></order>";
context.message.body.write(resp);
context.message.header.set('Content-Type', "text/xml");
```

XSLT examples

Note:

- There is no XSLT API to write the message body; the output of the stylesheet transformation is used to update `message.body`.
- To read the request body or the message body, you must parse `message.body` with a [Parse](#) policy before the XSLT policy is executed. At the beginning of the assembly flow, `message.body` contains `request.body` so, after parsing `message.body`, you can then read the request from `message.body`.
- If the XSLT input document uses the context current payload, `message.body` is used for the input document `'/'` that can be accessed from the XSLT policy if `message.body` has been parsed as XML in advance; this is determined by the `input` property of the XSLT policy, see [xslt](#).

Read the message body:

```
<xsl:variable name="currentPayload" select="apigw:read-payload('message')"/>
```

Read the message body when the `input` property of the XSLT policy is set to `true`:

```
<xsl:variable name="currentPayload" select="/" />
```

## Rejecting the session - examples

---

Note: After the session is rejected, the API assembly flow is stopped and a fault response is returned, unless a catch handler is defined.

GatewayScript examples

To reject the session with an error name and message, and update the status reason and code:

```
context.reject('CustomError', 'You are not authorized to make this API call');
context.message.statusCode = '401 Unauthorized';
```

XSLT examples

To reject the session with an error name and message, and update the status reason and code:

```
<apigw:reject identifier="CustomError" status-code="401" reason="Unauthorized">You are not authorized to make this API call</apigw:reject>
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tags in API Connect

There are a number of forms of tagging in IBM® API Connect.

- You can create tags in an API that are visible in its OpenAPI definition and can be used to filter operations in the Developer Portal.
- You can tag items in the Developer Portal so that they can be searched for.

---

## Tags within an API

The OpenAPI specification allows the inclusion of tags in an API definition, and you can create these tags through the API Manager user interface. You can assign tags to an operation within an API.

Tags that are assigned to an operation can be used to group the operations of an API in the Developer Portal, by using the filter options for the API. These tags are visible to anyone who can view any of the Products to which the API belongs.

For more information, see [Editing an API definition](#) and [Configuring an operation](#).

Note: Though you can tag the API itself in the API Setup section, by using this method the tag is not visible in the Developer Portal.

---

## Tags in the Developer Portal

An administrator can assign tags to items in the Developer Portal, which are then available through the API Products search function in the Developer Portal.

For more information, see [Editing tags for a specific item](#) and [Managing tags in the Developer Portal](#).

---

## Related concepts

- [Working with Products](#)

---

## Related tasks

- [Creating an API definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring API security

You configure security for an API by creating one or more security definitions by using IBM® API Connect that specify various aspects of security configuration. You then select which definitions you want to apply to your API, and to the operations in your API.

---

## About this task

By default, the security definitions that you apply to your API are also applied to the operations in the API, but for each API operation you can override the default setting by specifying the types of security definition that you want the operation to inherit from the containing API.

---

## Procedure

You configure API security by completing the following steps:

1. Create one or more security definitions.
2. Apply one or more of those security definitions to the API.
3. Optional: Specify the security definitions that you want each API operation to inherit.  
For details of configuring API security, see the following subtopics:

- [Creating a security definition](#)  
A security definition specifies all the settings for a particular aspect of API security; for example, the user registry that you use to authenticate access to the API.
- [Applying security definitions to an API](#)  
The security definition contains security settings that you enforce to define access control requirements for the operations in the API, by applying the security definition to an API.
- [Applying security definitions to an API operation](#)  
You can specify whether or not an API operation inherits the security definitions that have been created in the containing API.
- [Enabling CORS support for an API](#)  
You can enable cross-origin resource sharing (CORS) support for your API. CORS allows embedded scripts in a web page to call the API across domain boundaries.
- [LDAP authentication](#)  
The Lightweight Directory Access Protocol (LDAP) is an internet protocol for accessing and maintaining distributed directory information services over a network. If you rely on LDAP to authenticate users for web applications, take a minute to review the contents of this topic before beginning.

- [Authentication URL user registry](#)

You can use an Authentication URL user registry to specify a REST authentication service to manage user authentication, and to optionally provide additional metadata to be embedded in the token.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a security definition

A security definition specifies all the settings for a particular aspect of API security; for example, the user registry that you use to authenticate access to the API.

### About this task

You can create security definitions of the following types:

Type	Description
Basic authentication	Use a basic authentication security definition to specify a user registry or an authentication URL to be used to authenticate access to the API.
API key	Use an API key security definition to specify what application credentials are required to call an API.
OAuth	Use an OAuth security definition to specify settings for OAuth token based authentication for your API.

The following subtopics describe how to create security definitions of each type:

- [Creating a basic authentication security definition](#)  
When you create a basic authentication security definition in an API, you provide details of a user registry to be used to authenticate access to the API operations.
- [Creating an API key security definition](#)  
When you create an API key security definition in an API, you specify the credentials that an application must provide to identify itself when calling the API operations.
- [Creating an OAuth security definition](#)  
When you create an OAuth security definition, you provide settings for controlling access to the API operations through the OAuth authorization standard.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a basic authentication security definition

When you create a basic authentication security definition in an API, you provide details of a user registry to be used to authenticate access to the API operations.

### Before you begin

IBM® API Connect supports two types of user registries: Authentication URL user registry and LDAP user registry.

Before you can create a basic authentication security definition in an API, the user registry must exist. To create a user registry, you can use either API Manager or Cloud Manager. When you create a registry in API Manager, it is visible only to your provider organization. When you create a registry in Cloud Manager, you can make it visible to multiple provider organizations.

To create a user registry with API Manager, see [Working with user registries](#).

To create a user registry with Cloud Manager, see [User registries overview](#).

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

When you use basic authentication, you require API users to provide a valid user name and password to access selected operations. The application developer must also provide an HTTP authorization header in requests that are sent to operations that require basic authentication.

When you use an authentication URL, the user credentials that are provided in the authorization header are validated by the endpoint specified in the URL. If the user is authenticated, IBM API Connect expects an authentication URL to return an HTTP 200 OK response status code. All other HTTP response status codes result in an authentication failure and access is denied.


You cannot apply more than one basic security definition to an API. If you apply a basic security definition, you cannot also apply an OAuth security definition. For information on applying security definitions, see [Applying security definitions to an API](#).

For more information about using an LDAP user registry for authentication, see [LDAP authentication](#).

For information about using an Authentication URL, see [Authentication URL user registry](#).

### Procedure

To create a basic authentication security definition, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
  2. To create the security definition in an existing API, click the API you want to work with. To create a new API to add the security definition to, see [Creating an API definition](#).
  3. Select Security Definitions and click Add.
  4. Enter a name for the security definition and an optional description.
  5. For Type, select Basic.
  6. Click Choose one... from the section Authenticate using User Registry (optional). Select a user registry.  
Note: The user registries that you can select from are those that are specified in the Sandbox Catalog for the management server and provider organization that you are connected to.  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).  
If you are working offline in API Designer, you must type in the exact name of the user registry.  
You will need to specify the selected user registry in any Catalog to which the API is to be published.  
For details of how to specify the user registries for a Catalog, see [Creating and configuring Catalogs](#).
7. Click Save to save your changes.

## Results

---

A Basic security definition is now added to the Security Definitions.

## What to do next

---

Apply your security definition to the API, or to one or more API operations. For more information, see [Applying security definitions to an API](#) and [Applying security definitions to an API operation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating an API key security definition

When you create an API key security definition in an API, you specify the credentials that an application must provide to identify itself when calling the API operations.

## About this task

---


You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

You can require that, when calling an API operation, an application must provide either a client ID, or a client ID and client secret; you create an API key security definition to specify a credentials requirement. If you require that an application must provide both a client ID and client secret, you must create two API key security definitions, one for each type of credentials.

## Procedure

---

To create an API key definition, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. To create the security definition in an existing API, click the API you want to work with. To create a new API to add the security definition to, see [Creating an API definition](#).
3. Select Security Definitions and click Add.
4. Enter a name for the security definition and an optional description.
5. For Type, select API Key.
6. Specify whether the credentials are sent in the request header, or as query parameters, by selecting one of the following Located In options:

### Header


The credentials are sent in the request header. This is the default setting.

### Query

The credentials are sent as query parameters. This method is less secure because the client secret could be exposed in a log file.

The selected option is enforced, and API calls fail if the credentials is included in the wrong location by the caller.

Note: You must specify the same location for the client ID and client secret, either Header or Query.

7. Optional:  Select the Key Type; the options are Client ID and Client Secret. Select the option that is applicable to the type of API key security definition you are creating.  
The field is mapped to the `x-key-type` property for the security definition in the OpenAPI source for the API, where the corresponding values are `client_id` and `client_secret`; use these values if you modify the OpenAPI source directly. If you do not select a value for the Key Type field, you instead use the Parameter name to specify the key type, as explained in step [8](#).

Important: The `client_id` is not confidential and functions as a "SecurityScheme" only in the meaning of the OpenAPI specification.



If you use only the `client_id` (without the `client_secret`) as your API Key, be aware that the `client_id` is not considered a confidential value in the API Manager UI; for example, it might be displayed to other authenticated members of the provider organization. The `client_id` is treated as a public identifier for apps and cannot be kept private.

If authentication of the identity of the app calling an API is needed, then the API Key should use both a `client_id` to identify the app, plus a `client_secret` to authenticate it. The `client_secret` is designed to be known only to the app and the authorization server, and it should be protected. IBM API Connect protects the `client_secret`: it is only available when first generated, it can be stored as a one-way hash to prevent retrieval or leaks, and it can be changed and reset without changing the corresponding `client_id`. When an API Key is not marked as a `client_secret`, it will not be protected in the API Manager UI.

8. Specify the Parameter name.

If your API is not enforced by the API Connect gateway, enter the parameter name required by your gateway.

If your API is enforced by the IBM® API Connect gateway, the required value depends on the gateway type setting for the API, as follows:

- DataPower API Gateway: if you did not select a value for the Key Type in step 7, then for a key of type client secret you must include the string "secret", ignoring case, in the value you specify in the Parameter name field; if the value does not contain the string "secret" then the key is assumed to be of type client ID.

If you did select a value for the Key Type then you can provide any value you like for the Parameter name field; the value is mapped to the `name` property for the security definition in the OpenAPI source for the API, allowing you set the `name` to any desired value while still retaining the required key type.

- DataPower Gateway (v5 compatible): the value must be as specified in the following table, depending on where the client credentials are located, and the type of credentials. The field is automatically populated, according to value you select in the Located In field in step 6, as follows:

- If there are no existing API key definitions of type client ID, the key type is assumed to be client ID.
- If there is an existing API key definition of type client ID, the key type is assumed to be client secret.

The value is mapped to the `name` property for the security definition directly in the OpenAPI source. If the pre-populated value is not what you require, modify the value of the `name` property directly in the OpenAPI source on the Source tab.

Table 1. Client ID and Client secret parameter name values for the DataPower® Gateway (v5 compatible)

Location of credentials	Type of credentials	Parameter name
Header	Client ID	X-IBM-Client-Id
Header	Client secret	X-IBM-Client-Secret
Query	Client ID	client_id
Query	Client secret	client_secret

When you first create an API, a default API key security definition, of type client ID, is provided.

For information about including API key parameters in an API call, see [Calling an API](#).

Note:

- You cannot apply more than two API key security definitions to an API.
- If you apply an API key security definition for client secret, you must also apply an API key security definition for client ID.
- If you require the application developer to supply both client ID and client secret, you must apply two separate API key security definitions.
- You can have at most one API key definition of type client ID, regardless of whether the client ID is sent in the request header or as a query parameter.
- You can have at most one API key definition of type client secret, regardless of whether the client secret is sent in the request header or as a query parameter.
- The client ID and client secret headers that are specified in the request when the API is called are **not** added automatically to the message context. If you need these headers in the message context for subsequent processing, include a `set-variable` policy in your API assembly that adds the headers to the message content, taking the values specified in the request; you can do this in either of the following ways:
  - In the Assemble tab, add a Set Variable policy to your API assembly that defines the appropriate actions; for example:

Table 2.

Action	Set	Type	Value
Set	message.headers.X-IBM-Client-Id	string	\$(request.headers.X-IBM-Client-Id)
Set	message.headers.X-IBM-Client-Secret	string	\$(request.headers.X-IBM-Client-Secret)

For more information on configuring a Set Variable policy, see [Set Variable](#).

- In the Source tab, add a `set-variable` policy directly to the assembly section in your OpenAPI source; for example:

```
assembly:
  execute:
    - set-variable:
      version: 2.0.0
      title: set-variable
      actions:
        - set: message.headers.X-IBM-Client-Id
          value: $(request.headers.X-IBM-Client-Id)
          type: string
        - set: message.headers.X-IBM-Client-Secret
          value: $(request.headers.X-IBM-Client-Secret)
          type: string
```

For more information on defining a `set-variable` policy in your OpenAPI source, see [set-variable](#).

You should position the `set-variable` before any policy that needs to access the API key headers; for example, an invoke policy that calls a back-end service that is required to identify the application that made the initial request.

9. Click Save to save your changes.

## What to do next

Apply your security definition to the API, or to one or more API operations. For more information, see [Applying security definitions to an API](#) and [Applying security definitions to an API operation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating an OAuth security definition

When you create an OAuth security definition, you provide settings for controlling access to the API operations through the OAuth authorization standard.

### Before you begin

---

Before you can create an OAuth security definition, you must:

1. Create an OAuth provider.
  - To use Cloud Manager, see [Configuring a native OAuth provider](#) or [Configuring a third-party OAuth provider](#).
  - To use API Manager, see [Configuring a native OAuth provider](#) or [Configuring a third-party OAuth provider](#).
2. Add the OAuth provider to a catalog. If you have not created any catalogs, use the Sandbox Catalog. See the [OAuth instructions step](#) in [Creating and configuring Catalogs](#).

### About this task


---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

### Procedure

---

To create an OAuth security definition, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. To create the security definition in an existing API, click the API you want to work with. To create a new API to add the security definition to, see [Creating an API definition](#).
3. Select Security Definitions and click Add.
4. Enter a name for the security definition and an optional description.
5. For Type, select OAuth2.
6. From the OAuth Provider menu, select the provider that you want to use in this security definition.  
Note: The OAuth providers that you can select from are those that are specified in the Sandbox Catalog for the management server and provider organization that you are connected to.  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

If you are working in offline API Designer, you must you must type in the exact name of the OAuth provider.

You will need to specify the selected OAuth provider in any Catalog to which the API is to be published.

For details of how to specify the OAuth providers in a Catalog, see [Creating and configuring Catalogs](#).

7. From the Flow menu, select the grant type for the provider.  
The supported flow types are Application, Resource owner, Access Code, or Implicit. The values for the endpoints are automatically displayed in entries for Token URL and Authorization URL, as applicable to the flow type. For example:

Setting	Value
Flow	Application
Token URL	<a href="https://example.com/samplenative/oauth2/token">https://example.com/samplenative/oauth2/token</a>

Note: The Authorization URL and Token URL are maintained only for informative purposes, no validation or other action is applied to them by API Connect.

8. Optionally, specify additional scopes by clicking Add. For each additional scope, specify Name and Description.
9. Click Save to save your changes.

### What to do next

---

Apply your security definition to the API, or to one or more API operations. For more information, see [Applying security definitions to an API](#) and [Applying security definitions to an API operation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Applying security definitions to an API

The security definition contains security settings that you enforce to define access control requirements for the operations in the API, by applying the security definition to an API.

### Before you begin

---

Create one or more security definitions in your API. For more information, see [Creating a security definition](#).

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.


The following restrictions exist when you apply security definitions to an API:

- You cannot apply more than two API key security definitions to an API.
- If you apply an API key security definition for client secret, you must also apply an API key security definition for client ID.
- If you require the application developer to supply both client ID and client secret, you must apply two separate API key security definitions.
- You can have at most one API key definition of type client ID, regardless of whether the client ID is sent in the request header or as a query parameter.
- You can have at most one API key definition of type client secret, regardless of whether the client secret is sent in the request header or as a query parameter.
- You cannot apply more than one basic security definition to an API. If you apply a basic security definition, you cannot also apply an OAuth security definition.
- You can apply at most one OAuth security definition to an API.

## Procedure

---

To apply security definitions to an API, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. To apply the definitions to an existing API, click the API that you want to work with.  
To create a new API before you apply the definitions to it, see [Creating an API definition](#).

3. Navigate to the Security section.

4. In the Security section, select the security definitions that you want to apply.

When you apply a security definition to an API, the user interface presents the set of all existing security definitions. You can select one or more of the definitions from the set, to specify the exact combination of definitions that you want this API to satisfy.

In addition, you can specify multiple combinations of definitions. To specify a second combination, click Add again, and the interface presents a second set of all existing security definitions. Select the check box for each definition that you want included in the second combination. You can add additional sets, and specify additional combinations, until you've created all the valid combinations for the API.

An application can call your API if it satisfies any of the combinations you have defined.

Note: The following additional requirement applies to security definitions that will be used with an OAuth third party provider. If you select an OAuth security definition for protecting a consumer API, you must also include an API key security definition, as the `x-IBM-Client-Id` or `client_id` must be included in the security credentials so that the correct Plan configuration settings can be enforced.

The selected definitions are now applied to the API.

5. If the selected security definition is of type OAuth2, select the required scopes; the scopes available for selection are those that were specified in the security definition; for more information, see [Creating an OAuth security definition](#).  
Note: If you are using the DataPower® Gateway (v5 compatible), you must select at least one scope, and the scope sent in an API request must match one of the selected scopes, otherwise the call fails.  
If you are using the DataPower API Gateway, you only need select any scopes if Advanced scope check after token generation is not enabled in the native OAuth provider associated with the security definition. If a default scope has been set in the native OAuth provider and the API request doesn't contain any scope, the default scope is used; for more information, see [Configuring scopes for a native OAuth provider](#).
6. To remove a security definition so that it is no longer applied to the API, clear the selection for that security definition.
7. Click Save to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Applying security definitions to an API operation

You can specify whether or not an API operation inherits the security definitions that have been created in the containing API.

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.


You can choose to inherit all the security definitions, or you can individually select the security definitions that you want to inherit.

For information on creating security definitions in an API, see [Creating a security definition](#).

## Procedure

---

To specify the security definition inheritance settings for an API operation by using IBM® API Connect, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. To specify the security definition inheritance settings for an operation in an existing API, click the API you want to work with.  
To create a new API and API operation before specifying the security definition inheritance settings, see [Creating an API definition](#) and [Defining Paths for a REST API](#).

3. Click Paths, and then click the required path.
4. In the **Operations** section, click the required operation to display its details.
5. Specify which security definitions to apply to your operation. By default, all the security definitions that have been configured for the API are applied to the operation. To select which of the API security definitions you want to apply to the operation, complete the following steps:
  - a. Select Override API Security Definitions.  
An Add button is displayed.
  - b. Click Add, then select the required security definitions.  
When you apply a security definition to an API operation, the user interface presents the set of all existing security definitions. You can select one or more of the definitions from the set, to specify the exact combination of definitions that you want this API operation to satisfy.  
  
In addition, you can specify multiple combinations of definitions. To specify a second combination, click Add again, and the interface presents a second set of all existing security definitions. Select the check box for each definition that you want included in the second combination. You can add additional sets, and specify additional combinations, until you've created all the valid combinations for the API operation.  
  
An application can call your API operation if it satisfies any of the combinations you have defined.
- c. If the selected security definition is of type OAuth2, select the required scopes; the scopes available for selection are those that were specified in the security definition; for more information, see [Creating an OAuth security definition](#).  
Note: If you are using the DataPower® Gateway (v5 compatible), you must select at least one scope, and the scope sent in an API request must match one of the selected scopes, otherwise the call fails.  
If you are using the DataPower API Gateway, you only need select any scopes if Advanced scope check after token generation is not enabled in the native OAuth provider associated with the security definition. If a default scope has been set in the native OAuth provider and the API request doesn't contain any scope, the default scope is used; for more information, see [Configuring scopes for a native OAuth provider](#).  
  
Note: The following additional requirement applies to security definitions that will be used with an OAuth third party provider. If you select an OAuth security definition for protecting a consumer API, you must also include an API key security definition, as the **x-IBM-Client-Id** or **client\_id** must be included in the security credentials so that the correct Plan configuration settings can be enforced.
6. Click Save to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Enabling CORS support for an API


You can enable cross-origin resource sharing (CORS) support for your API. CORS allows embedded scripts in a web page to call the API across domain boundaries.

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

### Procedure

To enable CORS support for an API, complete the following steps:

1. In the navigation pane, click  Develop, then select the APIs tab.
2. To enable CORS support for an existing API, click the API that you want to work with.  
To create a new API before enabling CORS, see [Creating an API definition](#).
3. Select API Setup. Scroll to the Lifecycle section, and select CORS.
4. Click Save to save your changes.
5. Optional: To implement your own CORS solution using custom OPTIONS operations, complete the following steps:
  - a. Add the following headers to your HTTP responses:

```
Access-Control-Allow-Origin: https://<portalhostname>  
Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept
```

Where <portalhostname> is your Developer Portal host name.

- b. Optional: You can proxy your API through API Connect as an enforced invoke API so that CORS is handled automatically.

Important:

- If you implement your own CORS solution, you **must** disable the CORS option described in step 3
- CORS preflight requests are sent by using the HTTP **OPTIONS** method. Therefore, if you require these requests to be handled by the API Connect gateway then you must enable the **OPTIONS** method for all APIs that will handle preflight requests; see [Defining Paths for a REST API](#).
- **OPTIONS** requests are counted as API calls against any configured rate limit. Note that you can apply rate limits to individual operations; see [Defining rate limits for an API operation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## LDAP authentication

The Lightweight Directory Access Protocol (LDAP) is an internet protocol for accessing and maintaining distributed directory information services over a network. If you rely on LDAP to authenticate users for web applications, take a minute to review the contents of this topic before beginning.

## Programming guidelines

When authenticating with LDAP, observe the following guidelines:

- Use only a LDAP interface -- Users can only be authenticated via a connection to an LDAP server.
- Active Directory -- Use of an Active Directory interface is prohibited, however, LDAP authentication with Active Directory is supported.
- Administrator properties -- The LDAP administrator must have access to all user's LDAP properties.

LDAP attributes read by IBM® API Connect are as follows:

Attribute	Definition
mail	User's SMTP email address.
cn	Used internally to determine user's common name.
sn	Used internally to determine user's last name or surname.
givenname	Used internally to determine user's first name.
Prefix from Search dn	Used as username.

## Using an LDAP Server for User Authentication

Every entry in an LDAP directory server has a distinguished name (DN). The DN is the name that uniquely identifies an entry in the directory. A DN is made up of **attribute=value** pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

Any of the attributes defined in the directory schema can be used to make up a DN. The order of the component attribute value pairs is important. The DN contains one component for each level of the directory hierarchy from the root down to the level where the entry resides. LDAP DNs begin with the most specific attribute (usually some sort of name) and continue with progressively broader attributes, often ending with a country attribute. The first component of the DN is referred to as the Relative Distinguished Name (RDN). It identifies an entry distinctly from any other entries that have the same parent. In the previous examples, the RDN "cn=Ben Gray" separates the first entry from the second entry, (with RDN "cn=Lucille White"). These two example DNs are otherwise equivalent. The attribute=value pair making up the RDN for an entry must also be present in the entry. (This is not true of the other components of the DN.)

## Examples

In the following examples, a user search is performed from Base DN. The search is done anonymously, or if authenticated bind is used, an authenticated user is used. Search DN appends the prefix, the given user name, and the suffix. If the prefix used is **(uid=** and the suffix used is **)**, **uid** becomes the user name attribute. The default search filter used is: **(|(cn={filter}\*)(sn={filter}\*)(mail={filter}\*)(givenName={filter}))** and attributes used for prefix are also added to the search filter. In this case, where **'uid=** is the prefix, when searching for users, the search filter becomes:

```
(|(cn={filter}*)(sn={filter}*)(mail={filter}*)(givenName={filter}*)(uid={filter}*))
```

where **{filter}** is replaced by actual text.

The authenticated bind DN is a user on the external LDAP server permitted to get base DNs and search the LDAP directory within the defined search base. It should also be able to read other user properties and be used if anonymous access to LDAP to get base DNs and to search and get access to user attributes is not allowed. When a search is performed for **Steve**, the LDAP query filter shown in the following example is used and search is done from base DN specified in UI. When the user's DN is returned, the DN and password are used to authenticate the user:

```
(|(cn=Steve*)(sn=Steve*)(mail=Steve*)(givenName=Steve*)(uid=Steve*))
```

For bind during login calls, the search string used is the prefix. For example, if the prefix is **'uid=**, the search string used to search for a user during log in becomes: **(uid=Steve)**.

Using the mail attribute as the user name

When you want to use an email address as the user name, (for example *steve@company.com*), you typically use the mail attribute as the prefix **'mail=**. In this case, use the following search string to perform the search internally (assuming **'mail=** as prefix):

```
(|(cn=steve@company.com*)(sn=steve@company.com*)(mail=steve@company.com*)(givenName=steve@company.com*)(mail=steve@company.com*))
```

In the previous example, if DN is found, the password is used with the DN to perform a bind. The first result is taken, so be sure to use a unique attribute for username. Next, the LDAP properties for the user are read (**cn, sn, mail, givenName**). If LDAP properties cannot be read using the logged-in user, they are read from the user's DN using authenticated bind credentials. If the user name attribute differs, it is also queried for. For example, if the prefix is **'uid=**, the **uid** attribute is also read from the user's DN object:

```
(|(cn={filter}*)(sn={filter}*)(mail={filter}*)(givenName={filter}*))
```

Note: The prefix and suffix cannot be used to get a user's DN directly. For example, the following attempt to directly get a user's DN fails:

```
Prefix: "uid=_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.toolkit.doc_con_ldap_requirements_"
Suffix: ",ou=users,dc=company,dc=com"
```

## LDAP referrals

LDAP referrals allow a directory tree to be partitioned and distributed between multiple LDAP servers. An LDAP referral is a domain controller's way of indicating to a client application that it does not have a copy of a requested object, while giving the client a location that is more likely to hold the object. The client then uses the object as the basis for a DNS search for a domain controller.

API Connect support for LDAP referral includes:

- Searching for users that are part of multiple Active Directory trees and forests.
- Authenticating users in the Cloud Manager, API Manager and the Developer Portal.

Note: API Connect LDAP referral support is dependent on the following conditions:

- A single LDAP host/port is configured with administrator credentials and all users are referred to from the tree/server's base DN.
- As part of the user search following the LDAP referral, the same administrator credentials are used in the downstream trees/forests.
- LDAP API authentication does not support external LDAP referral. Internal LDAP referral is supported with IBM API Connect Version 2018.4.1.12 and later releases.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Authentication URL user registry

You can use an Authentication URL user registry to specify a REST authentication service to manage user authentication, and to optionally provide additional metadata to be embedded in the token.

This support can optionally enable any of the following:

- Providing the authenticated credential to IBM® API Connect. For example, the user logs-in with user name: `spoon`, and password: `fork`. When the user is authenticated, the credential becomes `cn=spoon,o=eatery`. The credential is kept in the OAuth `access_token` to represent the user.
- Providing metadata support. Allow extra metadata to be stored in the `access_token`.
- Overriding the `scope` that the application receives after a successful OAuth protocol processing. By responding with a specific header, the Authentication URL endpoint can replace the `scope` value that the application receives. For example, you can provide a specific resource owner an account number within the `scope` header response for use in future processing steps.

When you call the Authentication URL user registry, the API Connect gateway sends a GET request with HTTP headers and then processes any HTTP response from the URL. For authentication, a REST authentication service is expected at the Authentication URL.

The following response from the REST authentication service indicates that user authentication is successful and that API Connect will use `cn=spoon,o=eatery` as the user identity.

```
HTTP/1.1 200 OK
Server: example.org
X-API-Authenticated-Credential: cn=spoon,o=eatery
```

For information on how to configure a User Security policy in an API assembly for use with an Authentication URL user registry, see [User Security policy](#).

For an example of an OAuth provider configuration that uses an Authentication URL user registry, see [Example - using multiple OAuth policies in an OAuth provider assembly](#).

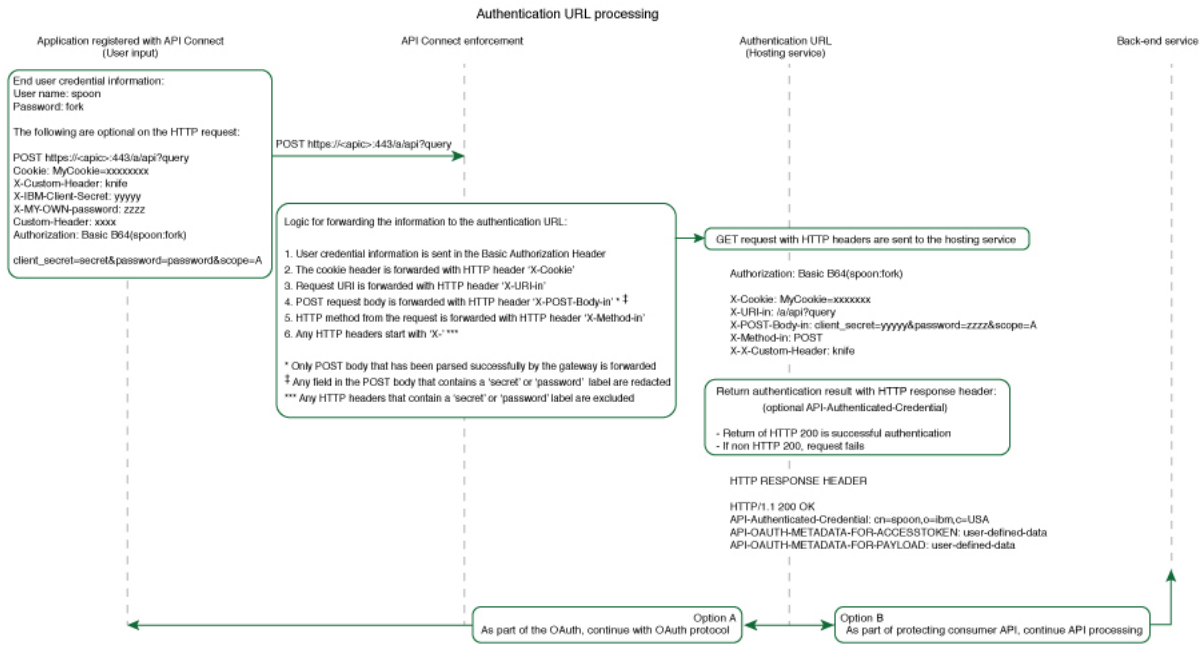
API Connect considers any non-200 HTTP response code a failed user authentication attempt.

When an Authentication URL user registry is invoked, two HTTP response headers are available that include metadata in the access token or the response payload that contains the access token. For more information, see [OAuth external URL and authentication URL](#). The two metadata response headers are:

```
API-OAUTH-METADATA-FOR-ACCESSTOKEN
API-OAUTH-METADATA-FOR-PAYLOAD
```

When an Authentication URL is invoked, an HTTP response header is available to override the requested `scope` from the application. For more information, see [Scope](#). The response header is:

```
x-selected-scope
```



If you are using the DataPower® API Gateway rather than the DataPower Gateway (v5 compatible), this diagram is provided for guidance only and is not fully accurate for this release.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

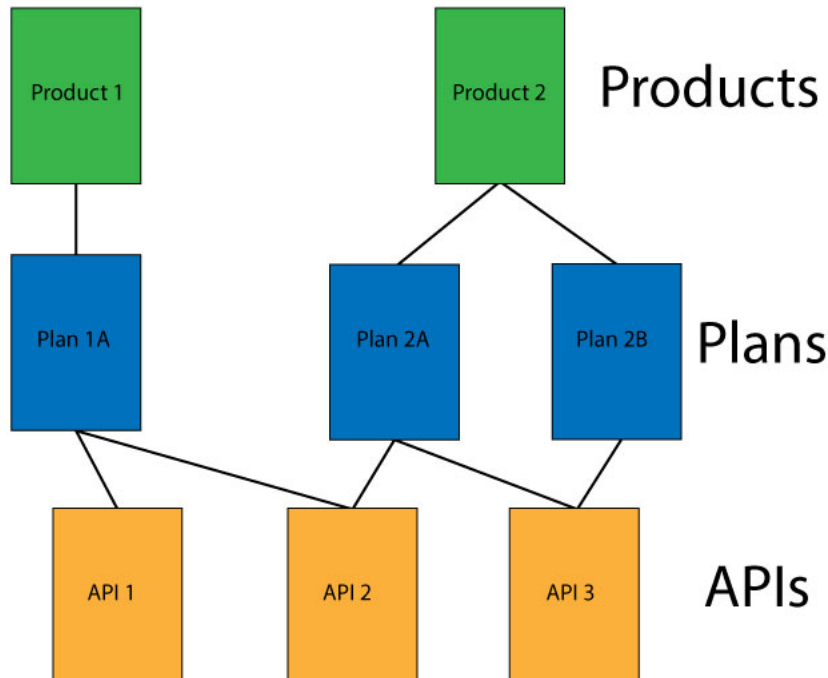
## Working with Products

In IBM® API Connect, Plans and APIs are grouped together in Products, with which you can manage the availability and visibility of APIs and Plans.

The following diagram demonstrates how Products, Plans, and APIs relate to one another.

Note: Plans belong to only one Product, but they can possess different APIs to other Plans within the same Product, and they can share APIs with Plans from any Product.

Figure 1. The hierarchy of Products, Plans, and APIs.



Products provide a method by which you can group APIs into a package that is intended for a particular use. Additionally, they contain Plans, which can be used to differentiate between different offerings. Plans can share APIs, but whether subscription approval is required depends upon the Plan itself. Additionally, you can enforce rate limits through these Plans, or through operations within the APIs of a Plan that override the rate limit of the Plan. You can also assign different subscription costs to your Plans to differentiate the rates of API calls.

To make an API available to an application developer, it must be included in a Plan. You can create Plans only within Products, and these Products are then published in a Catalog. A lifecycle manager can then control the availability and visibility of APIs and Plans through the API Manager. By using the API Manager, you can specify the Plans that an application developer is able to subscribe to through the Developer Portal. The application developer can only subscribe to one Plan from a specific Product. Multiple Plans within a single Product are useful in that they can fulfill similar purposes but with differing levels of performance and cost. For example, you can have a "Demo Plan", which makes a single API available free of charge, and a "Full Plan" which makes several APIs available with a monthly subscription cost.

As well as controlling which APIs an application developer can use, different Plans can be used to implement different rate limits. A rate limit can be implemented as a default rate that is shared across an entire Plan, or can be set for specific operations of an API within that Plan, exempting them from the shared Plan rate limit. Different Plans can have differing rate limits, both between operations and for the overall limit. This is useful for providing differing levels of service to customers. For example, a "Demo Plan" might enforce a rate limit of ten calls per minute, while a "Full Plan" might permit up to 1000 calls per second.

In addition, you can apply burst limits to your Plans, to prevent usage spikes that might damage infrastructure. Multiple burst limits can be set per Plan, at second and minute time intervals. You can also set multiple rate limits per Plan and per operation, at second, minute, hour, day, and week time intervals.

Note:

- Applying a rate limit at the Plan level creates a default rate limit that is shared across all the operations within the Plan. If you need to set specific rate limits for specific operations, you must set these within the operations themselves and these settings will override the setting at the Plan level.
- The test application that is used by the API Manager test tool is not subject to rate limits if you have enabled automatic subscriptions for the Catalog in which you are testing. For more information, see [Working with Catalogs](#)

API Connect also supports the implementation of multiple versions of Products. You can choose version numbers and use them to aid the development of your Products and Plans.

Note: The version for a Product is distinct from the version of any APIs that are contained in the associated Plans. Plans cannot themselves have their own version, they use the version of their parent Product.

- [Creating a draft Product](#)  
Create a draft Product to collect a set of APIs and Plans into one offering that you make available to your developers. A Plan includes rate limit settings that can be applied to the Plan as a whole, or specified for each operation in an API. Through Products and Plans, you have greater control over what APIs your developers have access to, and the subscription terms.
- [Downloading a draft Product](#)  
You can download the details of your draft Product so that you can store them for later recovery.
- [Importing a draft Product](#)  
You can create a Product by importing a Product definition in a .yaml file.
- [Editing a draft Product](#)  
After initially creating a draft Product, you can continue to further configure the Product, or you can make configuration changes later.
- [Staging a draft Product](#)  
Stage a draft Product to a Catalog to create a specific version of that Product, before publishing. When a Product is in the staged state, it is not yet visible to, or subscribable by, any developers.. The syndication feature in IBM API Connect means that if Spaces are enabled for a Catalog, Products can be staged only to a Space within that Catalog.
- [Publishing a draft Product](#)  
Publish a draft Product to make the APIs in that Product visible on the Developer Portal for use by application developers. The syndication feature in IBM API Connect means that if Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog.
- [Creating a new version of your Product](#)  
You can have multiple versions of a Product in IBM API Connect. These versions can occupy any of the lifecycle stages, which facilitates development.
- [Specifying the gateway type for a Product](#)  
You can edit the Product configuration to specify a gateway type.
- [Deleting a draft Product](#)  
You can delete draft Products that are no longer required.

## Related information

---

- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a draft Product

Create a draft Product to collect a set of APIs and Plans into one offering that you make available to your developers. A Plan includes rate limit settings that can be applied to the Plan as a whole, or specified for each operation in an API. Through Products and Plans, you have greater control over what APIs your developers have access to, and the subscription terms.

## Before you begin

---

Define your APIs. For more information, see [Creating an API definition](#).

Tip: As well as using the method described in this task, you can also create a Product when you create an API. If you create an API by using the developer toolkit command line editor, a Product is automatically created for you. You can then change any of the Product settings by opening your new Product in the Products page of the API Manager.



## About this task

---



You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

When you first create your Product, it has *draft* status. When you have finished configuring your draft Product, you stage it to a Catalog so that the APIs in the Product can be made available to your application developers.

## Procedure

---

To create a Product, complete the following steps:

1. In the navigation pane, click  Develop, then click Add > Product.  
The Add Product page opens.
2. Select New Product, then click Next.
3. On the Info page, complete the following steps:
  - a. Enter the Title and Version of the Product and, optionally, a description. A Name is entered automatically.  
Note: The value in the Name field is a single string that is used to identify the Product in developer toolkit CLI commands. The Title is used for display.  
To view the CLI commands to manage draft Products, see [apic draft-products](#).
  - b. Click Next.
4. On the APIs page, select the APIs that you want to include in the Product, then click Next.
5. On the Plans page, complete the following steps:
  - a. Optional: Change the Title, Description, or Rate limit settings for the default Plan.
  - b. Optional: To add further Plans, click Add, then scroll down and modify the Plan settings as required.
  - c. Click Next.
6. To have the Product published to the Sandbox Catalog automatically after creation, select Publish product.
7. In the Visibility section, specify the users that you want the Product to be visible to. You can choose Public to make the Product visible to all users, Authenticated to make the Product available to users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that you want the Product to be visible to.  
If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).
8. In the Subscribability section, specify the users that can subscribe to the Plans in the Product. You can choose Authenticated to make the Plans in the Product subscribable by users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product.  
If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product. If you selected custom visibility, only the consumer organizations and consumer organization groups selected there are available for adding to the custom subscribability list. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).  
Note: If you select custom visibility or subscribability, the consumer organizations and consumer organization groups that you can choose from are those in the Sandbox Catalog for the management server and provider organization that you are connected to. If you want to specify consumer organizations or groups that are in another Catalog, or in a Space, publish the Product to that Catalog or Space, then configure the visibility or subscribability there; for details, see [Changing the availability of a Product](#).  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).  
  
If you are working offline in API Designer, you must type in the exact name of the organization or group.  
  
For more information about consumer organizations in a Catalog, see [Administering Consumer organizations](#).
9. Click Next to create your Product.
  - a. To further configure your Product, click Edit Product.  
For details, see [Editing a draft Product](#)
  - b. If you do not want to configure your Product further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task.  
For details on how to configure your Product later, see [Editing a draft Product](#).

## Results

---

You have created a draft Product, and specified a set of APIs and Plans into one offering that you can now make available to your developers.

## What to do next

---

Stage your Product to a Catalog. For more information, see [Staging a draft Product](#).

## Related tasks

---

- [Creating a new version of your Product](#)

## Related reference

---

- [API and Product definition template examples](#)

## Related information

---

- [Publishing a Product](#)
- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Downloading a draft Product

You can download the details of your draft Product so that you can store them for later recovery.

### Before you begin

---

To complete this task, you must have created a draft Product. For more information, see [Creating a draft Product](#).

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.



**API Manager UI only:** To complete this task, you must be assigned a role that has the `Product-Drafts:Edit` permission. The pre-supplied Developer role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).

You can download the details of your Product as a .yaml file. This file can be used to re-create your Product, although it includes only references to APIs, not the API definitions themselves. This function does not provide a way to export an entire implementation. For information about creating a Product from a .yaml file, see [Importing a draft Product](#).

### Procedure

---

To download a .yaml file containing details of your Product, complete the following instructions:

1. In the navigation pane, click  Develop.
2. Alongside the Product version that you want to work with, click the options icon  and then click Download.
3. Save the file to the required location.

### Results

---

You have downloaded a .yaml representation of your Product.

### Related tasks

---

- [Importing a draft Product](#)

### Related information

---

- [Creating a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Importing a draft Product

You can create a Product by importing a Product definition in a .yaml file.

### Before you begin

---

To complete this task, you must have a .yaml representation of a Product. This can be created as described in the [Downloading a draft Product](#) task.

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.



You can import details of your Product as a .yaml file. This file can be used to re-create your Product, although it includes only references to APIs, not the API definitions themselves. This function does not provide a way to import an entire implementation.

You can import the file from your local file system.

## Procedure

---

To create a Product from a .yaml file, complete the following instructions:

1. In the navigation pane, click  Develop.
2. Click Add Product.
3. Select Existing Product, then click Next.
4. To import a file from your local file system, you can either drag and drop your files, or click Browse and select the file that you want to use. The wizard checks the validity of the YAML, and displays a message indicating successful validation.
5. Click Next.
6. To have the Product published to the Sandbox Catalog automatically after creation, select Publish product.
7. Click Next to create your Product.
  - a. To further configure your Product, click Edit Product.  
For details, see [Editing a draft Product](#)
  - b. If you do not want to configure your Product further at this time, click the left arrow icon  to return to the APIs and Products page; you can then move on immediately to another task.  
For details on how to configure your Product later, see [Editing a draft Product](#).

## Results

---

You have imported a Product from a .yaml file. If the same APIs and versions of APIs as referenced in the file are present, then they will be included in the Product.

Importing a Product in this way does not include the API definitions, only references to them.

## Related tasks

---

- [Downloading a draft Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Editing a draft Product

After initially creating a draft Product, you can continue to further configure the Product, or you can make configuration changes later.

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

During the initial creation of a Product, the Product creation wizard guides you to enter the minimum configuration settings, and then provides an Edit Product button. You have the option to specify additional configuration immediately, or to return to Product to edit it later. This topic describes how to specify additional configuration in either case.

## Procedure

---

To edit a draft Product, complete the following steps:

1. Access the Product Setup page in one of the following ways:
  - After completing the initial creation of the Product, click Edit Product.
  - In the navigation pane, click Develop, then click the name of the Product that you want to edit.
2. On the Product Setup page, you can make the following configuration changes:
  - a. In the Info section, change the Title, Version, or Summary.
  - b. Use the Contact, Terms of Service, and License sections to enter the corresponding details as required.
  - c. Change the gateway type. For more information, see [API Connect gateway types](#) and [Specifying the gateway type for a Product](#).
  - d. Click Save to save your changes.
3. Click Visibility, then make the following changes as required:
  - a. In the Visibility section, specify the users that you want the Product to be visible to. You can choose Public to make the Product visible to all users, Authenticated to make the Product available to users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that you want the Product to be visible to.  
If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that you want the Product to be visible to. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).
  - b. In the Subscribability section, specify the users that can subscribe to the Plans in the Product. You can choose Authenticated to make the Plans in the Product subscribable by users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product.  
If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product. If you selected custom visibility, only the consumer organizations and consumer organization groups selected there are available for adding to the custom subscribability list. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).  
Note: If you select custom visibility or subscribability, the consumer organizations and consumer organization groups that you can choose from are those in the Sandbox Catalog for the management server and provider organization that you are connected to. If you want to specify consumer organizations or



groups that are in another Catalog, or in a Space, publish the Product to that Catalog or Space, then configure the visibility or subscribability there; for details, see [Changing the availability of a Product](#).


If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

For more information about consumer organizations in a Catalog, see [Administering Consumer organizations](#).

- a. Click Save to save your changes.
4. To specify the APIs that you want to include in the Product, click APIs, then complete the following steps:
  - a. Click Edit.  
All draft APIs are listed.
  - b. Select the APIs that you want to include. You can include only APIs whose gateway type matches the gateway type of the Product; if you select an incompatible API, a warning message is displayed and you cannot save your changes until you clear the incompatible selection. For more information on gateway types, see [API Connect gateway types](#), [Specifying the gateway type for a Product](#), and [Specifying the gateway type for an API definition](#).
  - c. Click Save when done.  
The selected APIs are listed.

Note: To make an API available to application developers, you must include it in a Plan.

5. Optional: Add one or more Plans to the Product, or modify an existing Plan. Note that a default plan is automatically created for you, with a rate limit of 100 API calls per hour.
  - a. Click Plans.
  - b. To add a new Plan, click Add. To modify an existing Plan, click the options icon  alongside the required Plan, then click Edit.
  - c. Specify the Title of the Plan and, optionally, a description. A Name is entered automatically.  
Note: The value in the Name field is a single string that is used to identify the Plan in developer toolkit CLI commands. The Title is used for display.
  - d. Specify whether your Plan requires subscription approval. If you want subscriptions by developers to require approval, select the Approval check box; otherwise, ensure the check box is cleared.
  - e. In the Plan Rate Limits section, you can modify a rate limit, and click Add to add further rate limits. You can set multiple rate limits, at second, minute, hour, day, and week time intervals. To remove a rate limit, click the corresponding delete icon .  
Note: For information about rate limits and burst limits in API Connect, see [Understanding rate limits for APIs and Plans](#).
  - f. In the Plan Burst Limits section, you can modify a burst limit, and click Add to add further burst limits. You use burst limits to prevent usage spikes that might

damage infrastructure. You can set multiple burst limits, at second and minute intervals. To remove a burst limit, click the corresponding delete icon . Rate and burst limits work together to manage network traffic for the APIs covered under a Plan. A Plan can have multiple rate and burst limits, but it is recommended that each time interval be assigned only one set of limits. Adjust the rate and burst limits to allow for maximum traffic without overloading your network. The **rate limit** sets the maximum amount of sustainable, ongoing traffic for accessing the APIs on your network within a time interval (for example, one day). The **burst limit** sets the maximum short-term traffic volume for your network within a time interval (per second or minute).

The **burst limit** allows for short bursts of increased traffic. When the **burst limit** is exceeded, all subsequent API calls are rejected until the start of the next **burst limit interval**. The **burst limit counter** is reset to zero at the start of the next interval, which allows API calls to be accepted again. These API calls count toward the **rate limit counter**, but the resetting of the **burst limit counter** does not affect the **rate limit counter**.

The **rate limit** is the number of API calls allowed in a time interval, for example, 1000 calls per day. When the **rate limit** is exceeded, and Hard limit is enabled, all subsequent API calls will be rejected. The **rate limit counter** is reset to zero at the start of the next **rate limit interval**, which allows API calls to be accepted again. If Hard limit is disabled, all subsequent API calls are still accepted, and a message is logged stating that the **rate limit** has been exceeded. This is referred to as a "soft limit."

Hard limit affects only the **rate limit**, as illustrated by the following scenarios:

Scenario A

Table 1. Hard limit enabled

Hard limit	Burst limit	Rate limit
ON	100 calls/minute	1000 calls/day

- If a user calls an API 100 times in one minute, the **burst limit** is reached. The 101st call (and any subsequent calls) within the same minute will be rejected. Once the minute is up, the **burst limit counter** is reset. All API calls are tallied toward the **rate limit** of 1000 calls per day. The resetting of the **burst limit counter** does not affect the **rate limit counter**.
- If a user calls the API 99 times per minute, they will not hit the **burst limit**. They will eventually hit the 1000 calls per day rate limit, and the 1001st call will be rejected until the end of the same one day **rate limit interval**. During the period of time when API calls are rejected due to the daily **rate limit** being exceeded, the **burst limit** will not be activated since the calls are already rejected.
- Both **burst limits** and **rate limits** are applied per consumer.



Scenario B

Table 2. Hard limit disabled

Hard limit	Burst limit	Rate limit
OFF	100 calls/minute	1000 calls/day

- As in Scenario A, if a user calls the API 100 times in one minute, the 101st call within the same minute will be rejected, until that minute is up and the counter resets. These calls count toward the 1000 calls per day **rate limit** as well.
- If a user calls the API 99 times per minute, they will not hit the **burst limit**. They will eventually hit the 1000 calls per day **rate limit**, and the 1001st call will be accepted (since there is no hard limit). A message will be logged for each subsequent call until the time interval (one day) is up and the counter resets. During the remainder of the day, the **burst limit** will still be enforced, and calls will be rejected once the number of calls exceeds the 100 calls per minute within any given minute.
- Both burst limits and rate limits are applied per consumer.  
Note: When Hard limit is unchecked, the **rate limit** is considered a "soft limit." With a soft limit, calls are not rejected after the **rate limit** is reached. Instead, a message is recorded in the log file. With a soft limit, the **burst limit** still rejects API calls after it is exceeded.

- g. To include in the Plan all the APIs that were included in the Product in step 4, select Same as product in the Plan APIs section.
- h. To select the APIs, and API operations, that you want to include in the Plan, complete the following steps:
  - i. In the Plan APIs section, select Customize the plan API list, then click Edit. The Edit plan APIs window opens. The APIs that are available for selection are those that were included in the Product in step 4. Use the check boxes to specify the required APIs, then click Save when done.

- ii. To select specific API operations to include in the Plan, click the options icon  alongside the required API, then click Edit operation list. The Select operations you want to include window opens. Use the check boxes to specify the required operations, then click Save when done.
  - iii. To remove an API from the Plan, click the options icon  alongside the required API, then click Remove.
  - iv. To define rate limits for specific API operations, select Override plan rate limits for individual operation; for further details, see [Defining rate limits for an API operation](#).
  - i. Click Save to save your changes.
6. Optional: Click Categories, then define any categories that you want to organize your Product into. Click Save to save any updates. By organizing your Products into categories, you can provide a hierarchical display for your Products in the Developer Portal. For more information, see [Organizing your APIs and Products into categories](#).

---

## What to do next

Stage your Product to a Catalog. For more information, see [Staging a draft Product](#).

- [Defining rate limits for an API operation](#)  
For any Plan in a draft Product, you can apply one or more rate limits to a specific operation on any of the APIs in that Plan.

---

## Related tasks

- [Creating a new version of your Product](#)

---

## Related reference

- [API and Product definition template examples](#)

---

## Related information

- [Publishing a Product](#)
- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Defining rate limits for an API operation

For any Plan in a draft Product, you can apply one or more rate limits to a specific operation on any of the APIs in that Plan.

---

## About this task



You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

During the initial creation of a Product, the Product creation wizard guides you to enter the minimum configuration settings, and then provides an Edit Product button. You have the option to specify additional configuration immediately, or to return to Product to edit it later. This topic describes how to specify additional configuration in either case.

Note: For information about rate limits and burst limits in API Connect, see [Understanding rate limits for APIs and Plans](#).

---

## Procedure

1. Access the Product Setup page in one of the following ways:
  - After completing the initial creation of the Product, click Edit Product.
  - In the navigation pane, click Develop, then click the name of the Product that you want to edit.
2. Click Plans.
3. Click the options icon  alongside the required Plan, then click Edit.
4. Scroll down to the Plan APIs section, then select Override plan rate limits for individual operation. The Override Rate Limits section is displayed.
5. Click Add. The Override plan rate limits window opens.
6. Select the API, and the specific operation in that API, to which you want to apply rate limits. By default, calls to the operation are unlimited.
7. To apply one or more rate limits to the operation, select Rate Limits.
8. To define a rate limit, click Add.
9. Supply a name for the rate limit, and define the maximum number of calls allowed in a specified time period; for example, 100 calls per 1 minute. If you select HARD LIMIT and the limit is exceeded, subsequent calls are rejected, otherwise calls are still accepted and a message is logged.
10. Click Save when done. An entry for the operation is listed in the Override Rate Limits section.
11. To modify the rate limiting definition for an operation, click the options icon  alongside the required operation, then click Edit Operation; to delete it, click Remove.
12. Click Save to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Staging a draft Product

Stage a draft Product to a Catalog to create a specific version of that Product, before publishing. When a Product is in the staged state, it is not yet visible to, or subscribable by, any developers. The syndication feature in IBM® API Connect means that if Spaces are enabled for a Catalog, Products can be staged only to a Space within that Catalog.

---

### Before you begin

Ensure that you have a Catalog to stage to in the API Manager or API Designer user interfaces (UI). For more information, see [Creating and configuring Catalogs](#).  
Note: All references in this topic to a Catalog can also be applied to Spaces in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in IBM API Connect®](#).

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Product > Stage permission for the target Catalog or Space. For information on configuring Product management permissions for a Catalog or Space, see [Creating and configuring Catalogs](#) or [Managing user access in a Space](#).

---

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.  
Staging is not available when working offline in API Designer.

A Catalog is a staging target, and behaves as a logical partition of the DataPower® Gateway, and the Developer Portal.

You stage a Product so that the appropriate approvals, internally within the organization, can be given for it to then be published. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#). For information on publishing a Product, see [Publishing a Product](#).

You cannot re-stage an existing Product version to a production Catalog. If you update an API, you must create a new version of that API. Then, create a new version of the Product so that it includes the newer version of the API. For more information on creating new versions of APIs and Products, see [Creating a new version of an API definition](#) and [Creating a new version of your Product](#). If Spaces are enabled in a production Catalog, you cannot re-stage the same API or Product version to any Space in the Catalog.

You can, however, re-stage a Product version to the Sandbox Catalog, or other non-production Catalog, for ease of iterative testing. Note that when you re-stage a Product version to a non-production Catalog, any application subscriptions are deleted; this is done to facilitate scripted deployment in a continuous integration environment, where the re-creation of application subscriptions will be controlled by the scripts. If Spaces are enabled in a non-production Catalog and you re-stage a Product version to a different Space to the one in which it was previously staged, it is removed from the previous Space and staged to the newly specified Space.

Validation of the API OpenAPI definition file occurs during the staging or publishing process. The following validation occurs:



- Validation against the OpenAPI specification schema
- Validation against IBM extension properties
- Semantic validation, which includes the following types of validation:
  - Ensuring that if an API is enforced by an API Connect Gateway, then the scheme must be HTTPS, or the parameter name for an API key security scheme in the header must be either **X-IBM-Client-Id** or **X-IBM-Client-Secret**.
  - Ensuring that if the API is not enforced by an API Connect Gateway, then a "host" must be provided

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published (the `$ref` field is supported only if you are using the API Connect for IBM Cloud developer toolkit). For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

---

## Procedure

To stage a Product, complete the following steps:

1. In the navigation pane, click  Develop.  
The Develop: APIs and Products tab opens.
2. **Optional:** If you have accounts on multiple provider organizations, you can select a new provider organization for staging and publishing from the Organization menu.
3. Alongside the Product version that you want to work with, click the options icon  and then click Stage.  
If you have more than one version of the Product, ensure that you select the correct one.
4. Select the Catalog to which you want to stage the Product.
5. If Spaces have been enabled in the selected Catalog, select the Space that you require.  
Note: The Catalogs that you can select from are those that are defined for the management server and provider organization that you are connected to.  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).


For details of how to create a Catalog in a provider organization, see [Creating and configuring Catalogs](#).

6. If, when the staged Product is subsequently published, you want it to be published only to selected gateway services, select Publish to specific gateway services, then select the required gateway services. Only the gateway services whose type matches the gateway type setting for the Product are listed. For information on gateway types, see [API Connect gateway types](#).

7. Click Stage.

## Results

---

Your Product is staged to a Catalog. You can view the state of the Product in the Catalog in API Manager. If you staged the product from API Designer, ensure you are logged into API Manager with the same user name and password that you used for API Designer. Click  Manage in the API Manager UI, then select the required Catalog. The Product is shown with a state of Staged.

For information about the lifecycle of a product, see [The Product lifecycle](#).

If approval is required to stage Products in the Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is staged when the request is approved. If approval is not required, the Product is staged immediately.

For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

## What to do next

---

- Publish your Product for application developers to access it in the Developer Portal. For more information, see [Publishing a Product](#).

## Related information

---

- [Managing your Products](#)
- [Removing a Product from a Catalog](#)
- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Publishing a draft Product

---

Publish a draft Product to make the APIs in that Product visible on the Developer Portal for use by application developers. The syndication feature in IBM® API Connect means that if Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog.

## Before you begin

---

Ensure that you have a Catalog to stage to in the API Manager or API Designer user interfaces (UI). For more information, see [Creating and configuring Catalogs](#).

Note: All references in this topic to a Catalog can also be applied to Spaces in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in IBM API Connect®](#).

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Product Management permission for the target Catalog or Space. For information on configuring Product management permissions for a Catalog or Space, see [Creating and configuring Catalogs](#) or [Managing user access in a Space](#).

## About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI. Publishing is not available when working offline in API Designer.

You cannot re-publish a Product version to a production Catalog, you must create a new version of the Product for publication; see [Creating a new version of your Product](#). If Spaces are enabled in a production Catalog, you cannot re-publish the same Product version to any Space in the Catalog, neither the same Space nor a different Space.

You can, however, re-publish a Product version to the Sandbox Catalog, or other non-production Catalog, for ease of iterative testing. Note that when you re-publish a Product version to a non-production Catalog, any application subscriptions are deleted; this is done to facilitate scripted deployment in a continuous integration environment, where the re-creation of application subscriptions will be controlled by the scripts. If Spaces are enabled in a non-production Catalog and you re-publish a Product version to a different Space to the one in which it was previously published, it is removed from the previous Space and published to the newly specified Space.

Validation of the API OpenAPI definition file occurs during the staging or publishing process. The following validation occurs:



- Validation against the OpenAPI specification schema
- Validation against IBM extension properties
- Semantic validation, which includes the following types of validation:
  - Ensuring that if an API is enforced by an API Connect Gateway, then the scheme must be HTTPS, or the parameter name for an API key security scheme in the header must be either **X-IBM-Client-Id** or **X-IBM-Client-Secret**.
  - Ensuring that if the API is not enforced by an API Connect Gateway, then a "host" must be provided

Note: If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged or published (the `$ref` field is supported only if you are using the API Connect for IBM Cloud developer toolkit). For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).

## Procedure


---

To publish a Product, complete the following steps:

1. In the navigation pane, click  Develop.  
The Develop: APIs and Products tab opens.
  2. **Optional:** If you have accounts on multiple provider organizations, you can select a new provider organization for staging and publishing from the Organization menu.
  3. Alongside the Product version that you want to work with, click the options icon  and then click Publish.  
If you have more than one version of the Product, ensure that you select the correct one.
  4. Select the Catalog to which you want to publish the Product.
  5. If Spaces have been enabled in the selected Catalog, select the Space that you require.  
Note: The Catalogs that you can select from are those that are defined for the management server and provider organization that you are connected to.  
If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).
- For details of how to create a Catalog in a provider organization, see [Creating and configuring Catalogs](#).
6. If you want to publish the Product only to selected gateway services, select Publish to specific gateway services, then select the required gateway services. Only the gateway services whose type matches the gateway type setting for the Product are listed. For information on gateway types, see [API Connect gateway types](#).
  7. Click Publish.

## Results

---

Your Product is published to a Catalog. You can view the state of the Product in the Catalog in API Manager. If you published the product from API Designer, ensure you are logged into API Manager with the same user name and password that you used for API Designer. Click  Manage in the API Manager UI, then select the required Catalog. The Product is shown with a state of Published.

For information about the lifecycle of a product, see [The Product lifecycle](#).

If approval is required to publish Products in the Catalog, a publish approval request is sent, and the Product moves to the Pending state; the Product is published in the Catalog when the request is approved. If approval is not required, the Product is published immediately.

Note: If approval is required to stage Products to the Catalog, a stage approval request is sent. When the request is approved, the Product moves to the staged state and must be separately published in the Catalog. For information on publishing a staged Product in a Catalog, see [Publishing a Product](#).  
For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

## Related information

---

- [Managing your Products](#)
- [Removing a Product from a Catalog](#)
- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a new version of your Product

You can have multiple versions of a Product in IBM® API Connect. These versions can occupy any of the lifecycle stages, which facilitates development.

### Before you begin

---

Create your Product and assign it a version. For more information, see [Creating a draft Product](#).

### About this task

---



You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

You can create multiple versions of a Product, and these versions can be named according to your own preferences, and function as distinct Products when staged.

### Procedure

---

To create a new version of a Product, complete the following steps:

1. In the navigation pane, click  Develop.
2. Alongside the Product version that you want to work with, click the options icon , then click Save as a new version.  
The Save this product as a new version window opens.
3. Enter your new version number, then click Save.

## Results

---

Your Product is saved as a new version.



## Related information

- [Managing your Products](#)
- [Working with Products in the API Manager](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Specifying the gateway type for a Product

You can edit the Product configuration to specify a gateway type.

### Before you begin

IBM® API Connect supports two gateway types: DataPower® Gateway (v5 compatible) and DataPower API Gateway.

DataPower Gateway (v5 compatible) has been available with IBM API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

For more information on how to choose which gateway type to use, see [API Connect gateway types](#).

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

You can specify the gateway type for a Product.

Products can use only one type of gateway, and the APIs in the Product must use the same type of gateway. See [Specifying the gateway type for an API definition](#).

When you create a new Product, the gateway type defaults according to the gateway types of the gateway services that are enabled in your Sandbox Catalog, as indicated in the following table.


Table 1. Default gateway type for a new Product

Gateway Service enablement in Sandbox Catalog	Default gateway type for new Product
No gateway services enabled	DataPower Gateway (v5 compatible)
At least one gateway service of each type enabled	DataPower Gateway (v5 compatible)
One or more gateway services only of type DataPower API Gateway enabled	DataPower API Gateway
One or more gateway services only of type DataPower Gateway (v5 compatible) enabled	DataPower Gateway (v5 compatible)

For details of how to enable gateway services in a Catalog, see [Creating and configuring Catalogs](#).

### Procedure

To specify the gateway type for a Product, complete the following steps:

1. Ensure that you know the type of gateway that your APIs are configured to use.
2. In the navigation pane, click  Develop, then click Products.
3. Click the Product you want to update.
4. On the Product Setup page, go to the Gateway Type section. Select the gateway type for your Product:
  - DataPower Gateway (v5 compatible)
  - DataPower API Gateway
5. Click Save to retain your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting a draft Product

You can delete draft Products that are no longer required.

### About this task

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.



**API Manager UI only:** To complete this task, you must be assigned a role that has the `Product-Drafts:Edit` permission. The pre-supplied Developer role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).

Note:

- Deleting a draft Product does not remove it from any Catalogs to which it has been published. For details on how to remove a Product from a Catalog, see [Removing a Product from a Catalog](#).
- You cannot directly delete a Product that was generated automatically from an API with the Activate API option; these Products have the title `api_title` auto product. To delete such Products, delete the API from which it was generated; the Product is deleted together with the API. For more information, see [Deleting an API definition](#).

## Procedure

---

1. In the navigation pane, click  Develop.
2. Alongside the Product version that you want to delete, click the options icon  and then click Delete.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating and validating API and Product definitions by using the CLI

The developer toolkit of IBM® API Connect provides a command line interface that you can use to create and publish API and Product definitions, and also to validate YAML or JSON definitions.

The following topics explain how to use the command line interface to create an OpenAPI definition file, create a Product definition file, and validate a YAML or JSON definition.

- [Creating an OpenAPI definition file](#)  
APIs are defined in OpenAPI definition files, in YAML format. You can create a default OpenAPI definition file by using the `create` command and then modify it by using an editor of your choice.
- [Specifying a gateway type for an API definition](#)  
An API definition is specific to one or other of the gateway types, DataPower® API Gateway or DataPower Gateway (v5 compatible). API definitions must specify which type of gateway the API uses.
- [Referring to an extension in an API definition](#)  
To refer to an OpenAPI extension in your API definition YAML (or JSON) file, add an `extensions` key under `x-ibm-configuration`.
- [Using \\$ref to reuse code fragments in your OpenAPI files](#)  
If you deploy an API to an IBM API Connect Management server by using the developer toolkit command line, you can use the `$ref` field in your OpenAPI YAML and JSON API definition files to reference a fragment of OpenAPI code that is defined in a separate file. When API Connect processes the source API definition file, the `$ref` field is replaced with the contents of the target file.
- [Creating a Product definition file](#)  
You define a Product by creating a Product definition file.
- [Using x-ibm-languages to create multilingual API and Product documentation](#)  
Create multilingual API and Product documentation by using the `x-ibm-languages` extension in your API and Product OpenAPI definitions.
- [Using x-example to control the examples displayed in the Developer Portal](#)  
OpenAPI does not support the use of `example` attributes on parameters, only on request and response objects and their properties. To allow API Developers to be able to control the examples displayed in the portal you can use `x-example` in OpenAPI parameters.
- [Using x-embedded-doc to add additional documentation to products and APIs](#)  
You can use `x-embedded-doc` to add additional documentation to a product or an API as part of their definition.
- [Using x-pathalias to give consistent URLs for products and APIs](#)  
You can use `x-pathalias` to assign a URL to a product or an API.
- [Validating the YAML or JSON definition of an API or Product](#)  
You can validate a YAML or JSON definition by using the IBM API Connect developer toolkit.
- [Creating and using API and Product definitions templates](#)  
You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition.
- [OpenAPI 3.0 support in IBM API Connect](#)  
From Version 2018.4.1.4, IBM API Connect supports the OpenAPI 3.0 specification, with some limitations.

## Related reference

---

- [API development and management commands](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating an OpenAPI definition file

APIs are defined in OpenAPI definition files, in YAML format. You can create a default OpenAPI definition file by using the `create` command and then modify it by using an editor of your choice.

You can stage or publish the API directly to a Catalog in API Manager by referencing the API in a Product definition file, and then using the `apic products:publish` command to publish the Product. You can also create a draft API in API Manager by using the `apic draft-apis:create` command.

You can create an API in the CLI by running `apic create:api`, and supplying additional arguments on the command line.

Another option is to create an API interactively in the command line by running `apic create:api` and following the prompts.

You can see further details and available options for the `apic create:api` command by running:

```
apic create:api --help
```

IBM provides an extension to the OpenAPI specification; this extension is described in [IBM extensions to the OpenAPI specification](#).

Note: Products that contain an API with a Swagger property using `regex` that include lookahead assertions, such as "(?" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!
The multipart 'openapi' field contains an OpenAPI definition with validation errors.
  definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0, col:
0)
400
```

## Creating an API definition from a template

You can use a custom Handlebars template to create an API by using the following command:

```
apic create:api --template template_filename --title api_title
```

where *template\_filename* is the name of the Handlebars template to use, and *api\_title* is the title of your API.

An API template file must have a `.hbs` file name extension. You can create a template from scratch, or start with the example (default) API template provided in [API and Product definition template examples](#).

You can create multilingual API and Product documentation by using an `x-ibm-languages` extension directly in the OpenAPI definition. For more information, see [Using x-ibm-languages to create multilingual API and Product documentation](#).

- [IBM extensions to the OpenAPI specification](#)

You use the `x-ibm-configuration` object in your OpenAPI definition file to add extensions that are specific to IBM API Connect.

## Related reference

- [API development and management commands](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## IBM extensions to the OpenAPI specification

You use the `x-ibm-configuration` object in your OpenAPI definition file to add extensions that are specific to IBM® API Connect.

The `x-ibm-configuration` extension has the following structure:

```
x-ibm-configuration:
  enforced: enforced_boolean
  phase: Phase
  testable: test_boolean
  cors:
    enabled: cors_boolean
  activity-log:
    activity_logging_extension
  assembly:
    execute:
      - assembly_component
  gateway: gateway_type
  type: api_type
  properties:
    properties_extension
  catalogs:
    catalogs_extension
```

The following table lists the various extensions used by API Connect, whether they are required, a description of their use and behavior, and their type.

Table 1. IBM extensions

Extension	Required	Description	Type
phase	Yes	Use the phase extension to describe the maturity of the API. It can take the following values: <ul style="list-style-type: none"><li>• <b>identified</b>: the API is in the early conceptual phase and is neither fully designed nor implemented.</li><li>• <b>specified</b>: the API has been fully designed and passed an internal milestone but has not yet been implemented.</li><li>• <b>realized</b>: the API is in the implementation phase.</li></ul>	String
testable	Yes	Use the <code>testable</code> extension to specify whether the API can be tested using the test tool in the Developer Portal.	Boolean

Extension	Required	Description	Type
enforced	Yes	Use the <b>enforced</b> extension to specify if the API Connect gateway is used to enforce the API. <ul style="list-style-type: none"> <li>• <b>true</b> indicates that the API Connect gateway is used to enforce the API.</li> <li>• <b>false</b> indicates that the API Connect gateway is not used to enforce the API.</li> </ul>	Boolean
cors	Yes	Use the <b>cors</b> extension to specify whether CORS access control is used for the API. The extension has an <b>enabled</b> field which is a Boolean.	Object (with a single Boolean field)
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> activity-log	No	Use the <b>activity-log</b> extension to configure your logging preferences for the API activity that is stored in Analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.	Object
assembly	No	Use the <b>assembly</b> extension to describe the application of policies and logic to an API. It contains an <b>execute</b> field that contains an array of policies that are applied in order. It can contain a <b>catch</b> field that contains an array of error cases to be caught. For information about use of the <b>execute</b> field, see <a href="#">execute</a> . For information about use of the <b>catch</b> field, see <a href="#">catch</a> .	Object
gateway	No	Use the <b>gateway</b> extension to specify which type of gateway you want to use. If you are using the DataPower® Gateway (v5 compatible) or DataPower API Gateway, it must be included and take one of the following values: <ul style="list-style-type: none"> <li>• <code>datapower-gateway</code> (DataPower Gateway (v5 compatible))</li> <li>• <code>datapower-api-gateway</code> (DataPower API Gateway)</li> </ul> For information about types of gateways, see <a href="#">API Connect gateway types</a> .	String
type	No	The <b>type</b> extension takes one of the following values: <ul style="list-style-type: none"> <li>• <code>rest</code></li> <li>• <code>wSDL</code></li> </ul>	
properties	No	Use the <b>properties</b> extension to define properties for use in an API.	Object <a href="#">(properties)</a>
catalogs	No	Use the <b>catalogs</b> extension to define Catalog-specific values for properties defined in the <b>properties</b> extension.	Object <a href="#">(catalogs)</a>

The following example shows the **x-ibm-extension** section of an API that is enforced by API Connect, is in the realized state, is testable through the test tool in the Developer Portal, has CORS access control enabled, and has a simple assembly that invokes a URL and then redacts a field from the request or response.

```
x-ibm-configuration:
  enforced: true
  phase: realized
  testable: true
  cors:
    enabled: true
  assembly:
    execute:
      - invoke:
          title: Example Invoke
          target-url: 'https://example.com/api'
          description: Example description
      - redact:
          actions:
            - action: redact
              from:
                - request
                - response
              path: /**[@name='secondaryAddress']/**[@name='streetAddress']
  properties:
    ID:
      value: 1234
      description: An ID to be used for validating.
      encoded: false
  catalogs:
    Sandbox:
      properties:
        ID: 5678
```

- [catch](#)  
Use the **catch** extension to catch errors that occur during an API call.
- [properties](#)  
Use the **properties** extension to define properties for referencing in an API.
- [catalogs](#)  
Use the **catalogs** extension to assign catalog-specific values to properties defined in the **properties** section.
- [activity-log](#)  
If you're using the DataPower API Gateway, you can use the **activity-log** extension to configure your logging preferences for the API activity that is stored in Analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## execute

The execute field of an assembly has the following structure:

```
execute:
- Policy_1
- Policy_2
```

Note: Although some built-in policies can be used with both the DataPower® Gateway (v5 compatible) and the DataPower API Gateway, some policies are restricted to a particular Gateway. The following icons indicate which Gateway each policy can be used with:

-  Indicates that the policy can be run on the DataPower Gateway (v5 compatible).
-  Indicates that the policy can be run on the DataPower API Gateway.

For details of the two types of gateway, see [API Connect gateway types](#).

The following table describes the possible policies and logic constructs that can be included in an execute field.

Table 1. execute properties

Property	Required	Description	Data Type		
activity-log	No	Use the activity-log policy to log information that relates to the calling of API operations.	object ( <a href="#">activity-log</a> )	✓	✓ Functionality provided by the <a href="#">activity-log</a> extension
client-security	No	Provides a range of options for authenticating client access to your APIs, extending the capabilities of the OpenAPI specification.	object ( <a href="#">client-security</a> )	✗	✓ Available from V2018.4.1.5
gatewayscript	No	Include a GatewayScript program.	object ( <a href="#">gatewayscript</a> )	✓	✓ Available from V2018.4.1.0
if	No	Use the if policy to execute a section of the assembly only when a condition is fulfilled.	object ( <a href="#">if</a> )	✓	✓ Available from V2018.4.1.0; functionality provided by <a href="#">switch</a>
invoke	No	Use the invoke policy to call an API. The last invoke in your policy might be replaced by a proxy automatically to improve performance. To disable this, see: <a href="#">API properties</a> .	object ( <a href="#">invoke</a> )	✓	✓
json-to-xml	No	Convert payload from JSON to XML.	object ( <a href="#">json-to-xml</a> )	✓	✓
jwt-generate	No	Generate a JSON Web Token (JWT).	object ( <a href="#">jwt-generate</a> )	✓	✓
jwt-validate	No	Validate a JSON Web Token (JWT).	object ( <a href="#">jwt-validate</a> )	✓	✓
log	No	Use the log policy to customize or override the default activity logging configuration for an API.	object ( <a href="#">log</a> )	✗	✓ Available from V2018.4.1.7
map	No	Use the map policy to transform variables.	object ( <a href="#">map</a> )	✓	✓
operation-switch	No	Use the operation-switch policy when you want to execute alternative policy assemblies, conditional on the operation that is being called.	object ( <a href="#">operation-switch</a> )	✓	✓ Available from V2018.4.1.0; functionality provided by <a href="#">switch</a>
oauth	No	Use the oauth policy to policy to perform OAuth processing based on defined OAuth provider settings.	object ( <a href="#">oauth</a> )	✗	✓
parse	No	Use the parse policy to control the parsing of an input document. When the input document is a JSON string, the string is parsed instead of copied over.	object ( <a href="#">parse</a> )	✗	✓
proxy	No	Proxy a service.	object ( <a href="#">proxy</a> )	✓	✓ Functionality provided by <a href="#">invoke</a>
ratelimit	No	Use the ratelimit policy to apply one or more rate or burst limits at any point in your API assembly flow. Rate and burst limits restrict the number of calls that an application can make to an API in a specified time period.	object ( <a href="#">ratelimit</a> )	✗	✓ Available from V2018.4.1.7
redact	No	Use the redact policy to completely remove or to redact specified fields from the request body, the response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.	object ( <a href="#">redact - DataPower API Gatewayredact - DataPower Gateway (v5 compatible)</a> )	✓	✓ Available from V2018.4.1.7
set-variable	No	Use the set-variable policy to set a runtime variable to a string value, or to add or clear a runtime variable.	object ( <a href="#">set-variable</a> )	✓	✓ Available from V2018.4.1.0

Property	Required	Description	Data Type	DataPower Gateway	API Gateway
switch	No	Use the switch policy to execute one of a number of sections of the assembly based on which specified condition is fulfilled.	object ( <a href="#">switch</a> )	✓	✓ Available from V2018.4.1.0
throw	No	Use the throw policy to specify points at which an error should be thrown.	object ( <a href="#">throw</a> )	✓	✓ Available from V2018.4.1.0
user-defined-policy	No	You can apply your own user-defined policies to your APIs.	object	✓	✗
user-security	No	Extract a user's credentials, authenticate those credentials, and obtain authorization from the user.	object ( <a href="#">user-security</a> )	✗	✓
validate	No	Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.	object ( <a href="#">validate - DataPower API Gateway, validate - DataPower Gateway (v5 compatible)</a> )	✓	✓ Available from V2018.4.1.0
validate-usertoken	No	Validate a WS-Security UsernameToken.	object ( <a href="#">validate-usertoken</a> )	✓	✗
xml-to-json	No	Convert payload from XML to JSON.	object ( <a href="#">xml-to-json</a> )	✓	✓
xslt	No	Apply an XSLT transform to the payload.	object ( <a href="#">xslt</a> )	✓	✓ Available from V2018.4.1.0

The following example shows an execute field for an assembly that invokes a URL and then redacts a field from the request or response.

```
execute:
- invoke:
  title: Example Invoke
  target-url: 'https://example.com/api'
  description: Example description
- redact:
  actions:
    - action: redact
      from:
        - request
        - response
      path: /**[@name='secondaryAddress']/**[@name='streetAddress']
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## activity-log

Use the Activity Log policy to configure your logging preferences for the API activity that is stored in IBM® API Connect analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Activity Log](#).

Note: If you are using the DataPower API Gateway, you configure your logging preferences by using the [activity-log](#) extension.

## About

The activity-log policy has the following format:

```
- activity-log:
  version: version
  title: title
  description: description
  content: activity_to_log_if_call_successful
  error-content: activity_to_log_if_call_unsuccessful
```

Apply this policy by adding an assembly extension with an execute field to your OpenAPI definition file.

You can also apply an activity-log policy by using the API Manager assembly editor to add a built-in policy to the API. For more information, see [Activity Log](#) in the built-in policies section.

## Properties

The following table describes the policy properties:

Table 2. Activity Log policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	A title for the policy.	string
description	No	A description of the policy.	string
content	Yes	Defines the type of content to be logged when the operation is successful. Valid values: <ul style="list-style-type: none"> <li><b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li><b>activity</b>: Logs invocation only (only the resource URI is recorded).</li> <li><b>header</b>: Logs activity and header.</li> <li><b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <b>activity</b> .	string
error-content	No	Indicates what content to log if an error occurs. Valid values: <ul style="list-style-type: none"> <li><b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li><b>activity</b>: Logs invocation only (only the resource URI is recorded).</li> <li><b>header</b>: Logs activity and header.</li> <li><b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <b>payload</b> .	string

## Example 1

```
# use defaults
- activity-log:
  version: 1.0.0
  title: default activity logging
```

## Example 2

```
- activity-log:
  version: 1.0.0
  title: no logging for successful calls
  content: none
  error-content: activity
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## client-security

The client-security policy provides a range of options for authenticating client access to your APIs, extending the capabilities of the OpenAPI specification.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.5	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Client Security](#).

## About

You use the client-security policy in an API assembly to define how clients that call the API must supply authentication credentials.

The client-security policy has the following format:

```
- client-security:
  version: version
  title: title
  description: description
  stop-on-error: is_processing_stopped_on_client_security_failure
  secret-required: is_client_secret_required_in_request
  extract-credential-method: method_for_supplying_credentials
  id-name: parameter_that_specifies_client_id
  secret-name: parameter_that_specifies_client_secret
  http-type: authentication_type
  client-auth-method: method_for_client_authentication
```

## Properties

Table 2. client-security policy properties

Property	Required	Description	Data type
version	Yes	The policy version number.	string
title	No	The title of the policy.	string
description	No	A description of the policy.	string
stop-on-error	Yes	If set to <b>true</b> , assembly processing stops if client security fails, and an error is returned.	boolean
secret-required	Yes	If set to <b>true</b> , the client secret must be sent in the request. The secret is compared to the registered secret on the application that is identified by the client ID.	boolean
extract-credential-method	Yes	Specify one of the following values to define how the calling application authenticates: <ul style="list-style-type: none"> <li><b>header</b>: client ID and client secret credentials must be supplied in the request header.</li> <li><b>query</b>: client ID and client secret credentials must be supplied as query parameters in the request URL.</li> <li><b>form</b>: client ID and client secret credentials must be supplied as form data sent in a POST request.</li> <li><b>cookie</b>: client ID and client secret credentials must be supplied in a header called <b>Cookie</b>.</li> <li><b>http</b>: the calling application must authenticate by using basic authentication.</li> <li><b>context-var</b>: the credentials that are used for authenticating the client are obtained from context variables that you set in the assembly flow prior to the Client Security policy, by using a <a href="#">gatewayscript</a> policy for example. The names of these context variables are determined by the values that you supply in the id-name and secret-name properties of the client-security policy.</li> </ul>	string
id-name	Yes, unless extract-credential-method is set to <b>http</b>	The name of the parameter whose value specifies the client ID. For all values of extract-credential-method other than <b>context-var</b> and <b>http</b> , the calling application must supply a parameter with this name, in the location defined by the extract-credential-method setting. For <b>context-var</b> , this property specifies the name of a context variable. This option does not apply if the extract-credential-method property is set to <b>http</b> .	string
secret-name	Yes, if secret-required is set to <b>true</b> and extract-credential-method is other than <b>http</b>	The name of the parameter whose value specifies the client secret. For all values of extract-credential-method other than <b>context-var</b> and <b>http</b> , the calling application must supply a parameter with this name, in the location defined by the extract-credential-method setting. For <b>context-var</b> , this property specifies the name of a context variable. This option does not apply if the extract-credential-method property is set to <b>http</b> .	string
http-type	Yes, if extract-credential-method is set to <b>http</b>	The authentication type. Currently, this must be set to <b>basic</b> .	string
client-auth-method	Yes	Specify one of the following values: <ul style="list-style-type: none"> <li><b>native</b>: only client ID and client secret are used to authenticate the request. If extract-credential-method is set to <b>http</b> then the calling application must supply the client ID for the user name, and the client secret for the password.</li> <li><b>third-party</b>: a user registry is used to authenticate the client. If extract-credential-method is set to a value other than <b>http</b> then the calling application must supply the user name for the client ID, and the password for the client secret.</li> </ul>	string
user-registry	Yes, if the client-auth-method is set to <b>third-party</b>	Specify the value of the <b>name</b> property of the user registry that will be used to authenticate the client. The supported registry types are LDAP and authentication URL.	string

## client-security policy example

```
- client-security:
  version: 2.0.0
  title: client-security
  stop-on-error: true
  secret-required: true
  extract-credential-method: cookie
  id-name: my-client-id
  secret-name: my-client-secret
  client-auth-method: third-party
  user-registry: myauthurl
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## gatewayscript

Use the gatewayscript policy to execute a specified DataPower GatewayScript program.



## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [GatewayScript](#).

## About

The gatewayscript policy has the following structure:

```
- gatewayscript:  
  version: version  
  title: Title  
  description: Description  
  source: Script
```

## Properties

The following table describes the policy properties:

Table 2. gatewayscript policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	The title of the policy.	string
description	No	A description of the policy.	string
source	Yes	The GatewayScript source code to execute.	string

## Example

The following is an example of a simple gatewayscript policy:

```
- gatewayscript:  
  version: 1.0.0  
  title: Example_GatewayScript  
  source: console.debug('Hello World!');  
  description: A simple GatewayScript policy.
```

For more information about how to use a gatewayscript policy, see [GatewayScript](#) in the built-in policies section.

For more code examples, see [GatewayScript code examples](#) and, if you are using the DataPower API Gateway, [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

For general information on using GatewayScript, see the following topics in the DataPower product documentation:

- [GatewayScript APIs for API management](#)
- [Accessing and manipulating a variable in the API context](#)
- [OAuth context variables](#)

## Related concepts

- [Variable references in API Connect](#)

## Related reference

- [GatewayScript code examples](#)
- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## if

Use the if construct when you want to execute a portion of your assembly only when a specific condition is fulfilled.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0, functionality provided by <a href="#">switch</a>	

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [if](#).

## About

The if policy has the following format:

```
- if:
  version: version
  title: title
  description: description
  condition: 'condition_1'
  execute:
    policy_assembly_1 ...
```

In the condition field, use the form `apim.getvariable('context.location.variable')` to reference your variables, where *context* is the context that you want to reference, *location* is the location of the variable within that context, and *variable* is the name of the variable.

The **execute** section can define any policy assembly, including further if policies. For more information, see [execute](#).

## Properties

Table 2. if policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
condition	Yes	A script that returns <b>true</b> or <b>false</b> . Use GatewayScript for a DataPower Gateway implementation. For information about the context variables you can use and how to reference them in your script, see <a href="#">API Connect context variables</a> .	string
execute	Yes	The policy assembly that you want to execute if the condition returns <b>true</b> . For more information, see <a href="#">execute</a> .	string

## Example

```
# carry out different redaction actions depending on the operation
```

```
- if:
  version: 1.0.0
  title: clear_region_and_set_body
  condition: 'apim.getvariable('request.body.secret') == true'
  execute:
    - redact:
      title: remove secret field
      actions:
        - action: remove
          from: all
          path: /document/user/secret
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## invoke

Use the invoke policy to call an API.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Invoke](#).

## About

Note:

- The Invoke policy does not support responses with multipart form data, that is, when the response is set to `Content-Type: multipart/related`.
- The Invoke policy always uses chunked encoding, which is not supported by the HTTP 1.0 protocol.

The invoke policy has the following format:

```

- invoke:
  version: version
  title: title
  description: description
  target-url: URL_of_target_API
  tls-profile: TLS_profile_to_be_used
  verb: method_type
  timeout: timeout_value_in_seconds
  compression: is_data_to_be_compressed
  username: username_if_authentication_required
  password: password_if_authentication_required
  output: location_of_the_invoke_result
  cache-key: unique_identifier_of_the_document_cache_entry
  cache-response: cache_behavior
  cache-putpost-response: response_caching_behavior
  cache-ttl: cache_time_to_live
  inject-proxy-headers: are_proxy_headers_sent_to_target_url
  decode-request-params: are_request_parameters_decoded
  encode-plus-char: are_plus_characters_encoded
  keep-payload: is_payload_sent_on_delete
  use-http-10: are_transactions_restricted_to_http_1.0
  chunked-uploads: are_chunked_encoded_documents_sent_to_the_server
  header-control:
    .
    .
    .
    headers_to_copy_to_target_url
    .
    .
    .
  parameter-control:
    .
    .
    .
    parameters_to_copy_to_target_url
    .
    .
    .
  follow-redirect: url_redirection_behavior
  stop-on-error: errors_that_stop_the_flow

```

## Properties

The following table describes the properties of the invoke policy.

Table 2. Invoke policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
target-url	Yes	The URL of the target API.	string
tls-profile	No	The TLS profile to be used.	string
verb	No	The operation method type.  Valid values: <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• PATCH</li> <li>• HEAD</li> <li>• OPTIONS</li> </ul> The default value is <b>GET</b> . However, if the property is omitted from the source, the HTTP method from the incoming request is used.	string
timeout	No	The timeout value in seconds. The default value is <b>60</b> .	integer
compression	No	Specifies whether data is to be compressed by using compress before it is uploaded. The default value is <b>false</b> .	boolean
username	No	The user name, if authentication is required.	string
password	No	The password, if authentication is required.	string
output	No	The name of a variable that will be used to store the response data from the request. By default, the invoke response, that is the body, headers, statusCode and statusMessage, is saved in the variable <i>message</i> . Use this property to specify an alternate location to store the invoke response. This variable can then be referenced in other actions, such as <a href="#">map</a> . Note: If you want the response to be saved in <i>message</i> , leave the output property blank, do <b>not</b> supply the value <i>message</i> .	string
cache-key	No	Specifies the unique identifier of the document cache entry.	string

Property	Required	Description	Data type
cache-response	No	The cache response type. Valid values: <ul style="list-style-type: none"> <li><b>protocol</b>: The cache behavior is defined by the Cache-Control headers on the request and response.</li> <li><b>no-cache</b>: Specifies that there is no caching. However, if the document is already in the cache, the document is retrieved from the cache.</li> <li><b>time-to-live</b>: Specifies that the response stays in the cache for the specified time.</li> </ul> The default value is <b>protocol</b> .	string
cache-putpost-response	No	Specifies whether to cache the response from POST and PUT requests. Caching the response from POST and PUT requests can reduce server load and reduce latency in the response to the client request. The default value is <b>false</b> .	boolean
cache-ttl	No	Specifies the amount of time in seconds that the response stays in the cache. Applies only if the property cache-response is set to <b>time-to-live</b> . Enter a value in the range 5 - 31708800. The default value is <b>900</b> .	integer
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> inject-proxy-headers	No	When set to true, the <b>invoke</b> policy injects the <b>X-Forwarded-For</b> , <b>X-Forwarded-To</b> , <b>X-Forwarded-Host</b> , and <b>X-Forwarded-Proto</b> headers to the request that is sent to the <b>target-url</b> . The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> decode-request-params	No	When set to true, any request parameters that are referenced by a variable definition on the <b>target-url</b> of the <b>invoke</b> policy are URL-decoded. The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> encode-plus-char	No	When set to true, all "+" characters in the query parameter values of the <b>target-url</b> are encoded to %2F. The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> keep-payload	No	When set to true, the <b>invoke</b> policy sends a payload on an <b>HTTP DELETE</b> method. The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> use-http-10 (policy version 2.0.0 only)	No	When set to true, HTTP transactions are restricted to version 1.0. The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> chunked-uploads	No	If you set this property to <b>true</b> , chunked-encoded documents are sent to the server. When the HTTP 1.1 protocol is used, the body of the document can be delimited by either <b>Content-Length</b> or chunked encoding. While all servers can interpret <b>Content-Length</b> , many applications fail to understand chunked-encoded documents. For this reason, <b>Content-Length</b> is the standard method. The use of <b>Content-Length</b> interferes with the ability of the DataPower Gateway to fully stream. If you must stream full documents to the target server, enable this property.  When enabled, the server must be RFC 2616 compatible. Unlike all other HTTP 1.1 features that can be negotiated down at run time, you must know beforehand that the target server is RFC 2616 compatible.  The default value is <b>true</b> .  Note: Chunked encoding is not supported by the HTTP 1.0 protocol.	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> header-control: type values	No	Specifies the headers in <b>message.headers</b> that you want to copy to the target URL. If the <b>type</b> property is set to <b>blacklist</b> , the <b>values</b> property lists the headers that you don't want to be copied. If the <b>values</b> property is empty then all headers are copied.  If the <b>type</b> property is set to <b>whitelist</b> , the <b>values</b> property lists the headers that you want to be copied.  The items that are listed in the <b>values</b> property are in regular expression format. The values are not case-sensitive. The default value of the <b>header-control</b> property is  <b>header-control:</b> <b>type:</b> <b>blacklist</b> <b>values:</b> <b>[]</b>  See <a href="#">header-control examples</a>	string
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> parameter-control: type values	No	Specifies the parameters in the incoming request that you want to be copied to the target URL. If the <b>type</b> property is set to <b>blacklist</b> , the <b>values</b> property lists the parameters that you don't want to be copied.  If the <b>type</b> property is set to <b>whitelist</b> , the <b>values</b> property lists the parameters that you want to be copied. If the <b>values</b> property is empty then no parameters are copied.  The items that are listed in the <b>values</b> property are in regular expression format. The values are not case-sensitive. The default value of the <b>parameter-control</b> property is  <b>parameter-control:</b> <b>type:</b> <b>whitelist</b> <b>values:</b> <b>[]</b>  See <a href="#">parameter-control examples</a>	string
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> follow-redirect	No	Specifies the behavior if the back-end server returns the HTTP status code <b>301 Moved Permanently</b> . If this property is set to <b>true</b> , the <b>invoke</b> policy follows the URL redirection by making a further call to the URL specified in the <b>Location</b> header in the response. If this property is set to <b>false</b> , the <b>invoke</b> saves the <b>301</b> status code and the API call is considered to be complete. Note: The <b>follow-redirect</b> property is supported only by the DataPower API Gateway. If you are using the DataPower Gateway (v5 compatible), the <b>invoke</b> always follows the URL redirection; the <b>proxy</b> policy (not supported by the DataPower API Gateway) saves the <b>301</b> status code and completes the API call without following the URL redirection.	boolean

Property	Required	Description	Data type
stop-on-error	No	List the errors that, if thrown during the policy execution, cause the flow to stop. If there is a <code>catch</code> flow configured for the error, it is triggered to handle the error thrown. If an error is thrown and there are no errors specified for the Stop on error property, or if the error thrown is not one of the specified errors, the policy execution is allowed to complete, and the assembly flow continues.	string

## Example

```
- invoke:
  version: 2.0.0
  title: get the account status
  target-url: https://example.com/accounts/{id}?status={status}
  cache-response: time-to-live
  cache-putpost-response: true
  tls-profile: MyTLSProfile
  verb: POST
  timeout: 60
  compression: false
  username: MyUser
  password: MyPassword
  stop-on-error:
    - ConnectionError
    - OperationError
```

API Gateway only

## header-control examples

```
# copy all headers to the target URL

- invoke:
  target-url: http://myhost/mypath
  header-control:
    type: blacklist
    values: []

# copy all headers except X-Client-ID and Content-Type

- invoke:
  target-url: http://myhost/mypath
  header-control:
    type: blacklist
    values:
      - ^X-Client-ID$
      - ^Content-Type$

# copy no headers

- invoke:
  target-url: http://myhost/mypath
  header-control:
    type: whitelist
    values: []

# copy only the Content-Type header

- invoke:
  target-url: http://myhost/mypath
  header-control:
    type: whitelist
    values:
      - ^Content-Type$
```

API Gateway only

## parameter-control examples

```
# copy no request parameters to the target URL

- invoke:
  target-url: http://myhost/path?storeid=3
  parameter-control:
    type: whitelist
    values: []

# append the petid parameter to the target URL
# if the incoming request is http://apigw/org/sandbox/petstore/base?petid=100&display=detailed,
# the target URL at runtime will be http://myhost/mypath?storeid=3&petid=100

- invoke:
  target-url: http://myhost/path?storeid=3
  parameter-control:
    type: whitelist
    values:
      - ^petid$
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## json-to-xml

Use the json-to-xml policy to convert the context payload of your API from the JavaScript Object Notation (JSON) format to the extensible markup language (XML) format.

### Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [JSON to XML](#).

### About

The json-to-xml policy has the following structure:




```
- json-to-xml:  
  version: version  
  title: Title  
  description: Description
```

Note: If you are using the DataPower API Gateway, the input to the json-to-xml policy must be parsed data. One way to produce parsed data is to use a [parse](#) policy before a json-to-xml policy in your assembly flow, which provides explicit control of the parse action.

### Properties

The following table describes the policy properties:

Table 2. Policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	The title of the policy.	string
description	No	A description of the policy.	string
 input	No	The input message to convert. Specify the name of a variable in the API context. <b>variableName.body</b> , the message payload, represents the JSON input to convert. The default value of the variable is <b>message</b> and <b>message.body</b> is the default input.	string
 output	No	The output message to store the conversion result. Specify the name of a variable in the API context. <b>variableName.body</b> represents the result of conversion from JSON format to XML format. When the specified input message is the default message, the default output is <b>message.body</b> . Otherwise, when the input message is the variable <b>my-message-variable</b> , for example, the default output is <b>my-message-variable.body</b> . The variable cannot be any read-only in the API context.	string
 conversionType	No	The conversion type that determines the target format of the output. The following options are available: <ul style="list-style-type: none"><li>None: No conversion of the output takes place.</li><li>badgerFish: BadgerFish convention is used to determine the target conversion format of the output.</li></ul>	string
root-element-name	Yes	The root element name of the resultant XML document. This property is used only if the input JSON document is not hierarchical and has more than one top-level property, or if the always-output-root-element property is set to <b>true</b> .	
always-output-root-element	Yes	Select this property to <b>true</b> you always want the policy to output the root element, even if it is not required to make the XML document well formed.	boolean
unnamed-element-name	No	The XML element name to be used for JSON array elements.	string

### Example

The following is an example of a json-to-xml policy:

```
- json-to-xml:  
  version: 1.0.0  
  title: JSON to XML transform  
  description: Transforms JSON message body to XML format
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## jwt-generate

Use the Generate JWT security policy in IBM® API Connect to generate a JSON Web Token (JWT).

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Generate JWT](#).

## About

The jwt-generate policy has the following structure:

```
- jwt-generate:
  version: version
  title: title
  description: description
  jwt: json_web_token
  jti-claim: jwt_id_claim
  iss-claim: issuer_claim
  exp-claim: validity_period
  sub-claim: subject_claim
  aud-claim: audience_claim
  jws-jwk: sign_jwk_variable_name
  jws-alg: cryptographic_algorithm
  jws-crypto: sign_crypto_object
  jwe-enc: encryption_algorithm
  jwe-jwk: encrypt_jwk_variable_name
  jwe-alg: key_encryption_algorithm
  jwe-crypto: encrypt_crypto_object
```

## Properties

The following table describes the policy properties:

Table 2. Generate JWT policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	The title of the policy.	string
description	No	A description of the policy.	string
jwt	No	Runtime variable in which to place the JWT that is generated. The default value is: <b>generated.jwt</b> . However, if not set, the JWT that is generated is written to the Authorization Header as a Bearer token.	string
jti-claim	No	Indicates whether a JWT ID (jti) claim should be added to the JWT. If selected, the property is set to <b>true</b> , and a UUID is generated and set as the JTI claim value.	boolean
iss-claim	Yes	Runtime variable from which the Issuer (iss) claim string can be retrieved. This claim represents the Principal that issued the JWT. The default value is: <b>iss.claim</b>	string
sub-claim	No	Runtime variable from which the Subject (sub) claim string can be retrieved.	string
aud-claim	No	Runtime variable from which the Audience (aud) claim string can be retrieved. Multiple variables are set by using a comma-separated string.	string
exp-claim	Yes	The length of time (in seconds), that is added to the current date and time, in which the JWT is considered valid. The default value is <b>3600</b> .	integer
private-claims	No	Runtime variable from which a valid set of JSON claims can be retrieved. These claims are added to any set of claims specified previously.	string
jws-jwk	No	Runtime variable that contains the JWK that is used to sign the JWT. A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to sign the JWT. However, if both data types are specified, only the Crypto Object is used.	string
jws-alg	No	The cryptographic algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>• HS256</li> <li>• HS384</li> <li>• HS512</li> <li>• RS256</li> <li>• RS384</li> <li>• RS512</li> <li>• ES256</li> <li>• ES384</li> <li>• ES512</li> <li>• PS256</li> <li>• PS384</li> <li>• PS512</li> </ul>	string
jws-crypto	No	The cryptographic object to use to sign the JWT. A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to sign the JWT. However, if both data types are specified, only the Crypto Object is used.	string

Property	Required	Description	Data type
jwe-enc	No	The encryption algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>A128CBC-HS256</li> <li>A192CBC-HS384</li> <li>A256CBC-HS512</li> </ul>	string
jwe-jwk	No	Runtime variable that contains the JWK to use to encrypt the JWT.	string
jwe-alg	No	The key encryption algorithm to use. Valid values are: <ul style="list-style-type: none"> <li>RSA1_5</li> <li>RSA-OAEP</li> <li>RSA-OAEP-256</li> <li>dir</li> <li>A128KW</li> <li>A192KW</li> <li>A256KW</li> </ul>	string
jwe-crypto	No	The cryptographic object to use to encrypt the claim.	string

## Example

The following is an example of a jwt-generate policy:

```
- jwt-generate:
  version: 1.0.0
  title: jwt-generate
  iss-claim: iss.claim
  exp-claim: 3600
  jwt: generated.jwt
  jti-claim: true
  sub-claim: sub.claim
  aud-claim: aud.claim
  private-claims: private.claims
  jws-jwk: jws.jwk
  jws-alg: HS256
  jws-crypto: jwsCryptoObjectName
  jwe-enc: A128CBC-HS256
  jwe-jwk: jwe.jwk
  jwe-alg: A128KW
  jwe-crypto: jweCryptoObjectName
```

For more information about how to use a jwt-generate security policy, see [Generate JWT](#) in the built-in policies section.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## jwt-validate

Use the Validate JWT security policy to enable the validation of a JSON Web Token (JWT) in a request before allowing access to the APIs.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Validate JWT](#).

Note:

- If the original message was signed with a Shared Secret Key, the cryptographic object that is specified must also be a Shared Secret Key.
- If the original message was signed with a Private Key, the cryptographic object that is specified must be a Crypto Certificate (public certificate).
- The cryptographic material can be provided through a JSON Web Key (JWK).
- If a JWK header parameter is included in the header of the JWT, the parameter must match the JWK or cryptographic object that is specified in the policy, or the JWT validation will fail.
- If both a cryptographic object and a JWK are specified, the cryptographic object is used to decrypt or verify the JWT.
- The JWT validate action on the DataPower API Gateway can verify a JWT by using either a single JWK, or a JWK set.

## About

The jwt-validate policy has the following structure:

```
- jwt-validate:
  version: version
```



```

title: title
description: description
jwt: json_web_token
output-claims: output_full_set_of_jwt_claims
iss-claim: issuer_claim
aud-claim: audience_claim
jwe-crypto: decrypt_crypto_object
jwe-jwk: decrypt_crypto_jwk_variable_name
jws-crypto: verify_crypto_object
jws-jwk: verify_crypto_jwk_variable_name

```

## Properties

The following table describes the policy properties:

Table 2. Validate JWT policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	The title of the policy.	string
description	No	A description of the policy.	string
jwt	Yes	Context or runtime variable that contains the JWT to be validated. The default value is: <code>request.headers.authorization</code> . However, if this property is not set, the policy looks for the JWT in the <code>request.headers.authorization</code> location by default.  Note: The format of the authorization header must be:  " <b>Authorization: Bearer <code>jwt-token</code></b> "  where <code>jwt-token</code> is the encoded JWT.	string
output-claims	Yes	Runtime variable to which the full set of claims that are contained in the JWT is assigned. The default value is: <code>decoded.claims</code> .	string
iss-claim	No	The Perl Compatible Regular Expressions (PCRE) to use to validate the Issuer (iss) claim.	string
aud-claim	No	The PCRE to use to validate the Audience (aud) claim.	string
jwe-crypto	No	The cryptographic object (a shared key or certificate) to use to decode the claim. A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to decrypt the JWT. However, if both data types are specified, only the Crypto Object is used.	string
jwe-jwk	No	Runtime variable that contains the JWK to use to decrypt the JWT. A JWK and a Crypto Object are both valid ways of providing the cryptographic data necessary to decrypt the JWT. However, if both data types are specified, only the Crypto Object is used.	string
jws-crypto	No	The cryptographic object (a shared key or certificate) to use to verify the signature. A JWK, or JWK set, and a Crypto Object are both valid ways of providing the cryptographic data necessary to verify the JWT. However, if both data types are specified, only the Crypto Object is used.	string
jws-jwk	No	Runtime variable that contains the JWK, or JWK set, to use to verify the signature. A JWK, or JWK set, and a Crypto Object are both valid ways of providing the cryptographic data necessary to verify the JWT. However, if both data types are specified, only the Crypto Object is used.	string

## Example

The following is an example of a `jwt-validate` policy:

```

- jwt-validate:
  version: 1.0.0
  title: jwt-validate
  jwt: request.headers.authorization
  output-claims: decoded.claims
  iss-claim: "'^data.*'"
  aud-claim: "'^id.*'"
  jwe-crypto: jweCryptoObjectName
  jwe-jwk: jwe.jwk
  jws-crypto: jwsCryptoObjectName
  jws-jwk: jws.jwk

```

For more information about how to use a `jwt-validate` security policy, see [Validate JWT](#) in the built-in policies section.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## log

Use the log policy to customize or override the default activity logging configuration for an API.

## Gateway support

Table 1. Table showing which gateways support this policy, and the

corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Log](#).

## About

The log policy has the following format:

```
- log:  
  version: version  
  title: title  
  description: description  
  mode: activity_logging_actions
```

## Properties

The following table describes the properties of the log policy.

Table 2. log policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
mode	Yes	<p>Specify one of the following values:</p> <ul style="list-style-type: none"><li><b>gather-only</b>: gather all analytics data and write it to the <code>log</code> context variable, which populates the API event record on completion of the API execution. For more information on the fields in the <code>log</code> context variable, and the consequent API event record, see <a href="#">API event record fields</a>.</li><li><b>send-only</b>: perform the following actions:<ul style="list-style-type: none"><li>Read the data from the <code>log</code> context variable.</li><li>Truncate all message payloads and convert to a textual representation.</li><li>Send the data to the analytics server.</li></ul></li><li><b>gather-and-send</b>: perform a <b>gather-only</b> operation, immediately followed by a <b>send-only</b> operation.</li></ul> <p>If you use the Send-only or Gather-and-send option, data is buffered and sent to the analytics server in batches according to the time interval configured for the Analytics Endpoint on the DataPower API Gateway. For more information, see <a href="#">Configuring an analytics endpoint</a> in the DataPower knowledge center.</p> <p>Note: If you are offloading to a third party analytics server, you can redact any aspect of the event data. If you are using API Connect analytics, you can redact only request and response payloads.</p>	string

## Example

```
- log:  
  version: 2.0.0  
  title: Gather activity log data for processing  
  mode: gather-only
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## map

Use the map policy to transform your assembly flow and specify relationships between variables.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Map](#).

## About

For information about the use and structure of the map policy, see [The Map policy structure](#). For more information about API properties that affect the map policy, see [API properties](#).

The map policy has the following format:

```

- map
  version: version
  title: title
  description: description

  inputs:
  - input_1:
    variable: context_1
    $ref: '#/definitions/definition_1'
  - input_2
    variable: context_2
    type: type_2
    content: content_type

  outputs:
  - output_3
    variable: context_3
    type: type_3

  actions:
  - set: output_3.output_property_3
    from: input_1.input_property_1

  - set: output3.output_property_3
    from:
      - input1.input_property_1
      - input2.input_property_2
    value: 'script A'
    default: 'default A'

  - create: output3.output_property_3
    from: input1.input_property_1
    foreach: input1.input_property_1
    actions:
      - further_actions
  options:
    .
    .
    .
  advanced_XML_and_general_configuration_options
    .
    .
    .

```

## Properties

Table 2.

Property	Required	Description	Data type	Belongs to
version	Yes	The policy version number.	string	N/A
title	No	A title for the policy.	String	N/A
description	No	A policy description.	String	N/A
inputs	No	An array listing the inputs of the map policy.	Object	N/A
outputs	Yes	An array listing the outputs of the map policy.	Object	N/A
variable	Yes	A reference to the context variable that is the location of the input or output variable..	String	inputs or outputs
\$ref	Yes <sup>1</sup>	A reference to the definition of the type of the variable.	String	inputs or outputs
type	Yes <sup>1</sup>	The type of the variable.	String	inputs or outputs
content	No	The content type of the variable: <code>application/xml</code> or <code>application/json</code> . If <code>None</code> is selected, or the field is not included, then the type is treated as JSON.	String	inputs or outputs
actions	Yes	Lists actions to be performed by the map policy.	Object	N/A
set	Yes <sup>2</sup>	Specifies by name the output variable that is to be set to a value by the action.	String	actions
create	Yes <sup>2</sup>	Specifies by name the output variable that is to have a value appended to it by the action.	String	actions
from	No	Specifies by name any input variables used by the action.	String	actions
value	No	Contains a script that maps and transforms input variables into output variables.	String	actions
default	No	Contains a static value, or an inline variable reference, to be applied to the output when no input value is provided. For information on inline variable references, see <a href="#">inline references</a> .	String	actions
foreach	No	Specifies by name an array for which further actions should be performed for each element.	String	actions
includeEmptyXMLElements	No	If set to <code>true</code> , empty XML elements are included in the output of the map policy. Set to <code>false</code> if you do not want empty XML elements to be included in the output of the map policy. The default value is <code>true</code> .	Boolean	options
namespaceInheritance	No	If set to <code>true</code> , XML namespaces are inherited from the parent element. Set to <code>false</code> if you want the map policy to use explicit namespaces. The default value is <code>true</code> .	Boolean	options
inlineNamespaces	No	If set to <code>true</code> , XML namespaces will be inserted into the document where they are first used. Set to <code>false</code> if you want namespaces to all be defined on the root element. The default value is <code>true</code> .	Boolean	options
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> mapResolveXMLInputDataType	No	If set to <code>true</code> , XML elements whose schema is configured as type boolean or numeric will be converted to that data type. Set to <code>false</code> if you want all XML element values to be returned as a string.	Boolean	options

Property	Required	Description	Data type	Belongs to
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapXMLEmptyElement	No	<p>This property controls how the map policy handles the output of an empty XML element. Specify one of the following values:</p> <ul style="list-style-type: none"> <li><b>string</b> (the default option): An empty XML element is handled as an empty string.</li> <li><b>null</b>: An empty XML element is handled as a null value.</li> <li><b>none</b>: The data is ignored.</li> <li><div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> <b>string-badgerfish</b>: The value for an empty XML element is considered to be an empty string. The empty string value will be placed into a JSON badgerfish value property.</li> <li><div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> <b>null-badgerfish</b>: The value for an empty XML element is considered to be null. The null value will be placed into a JSON badgerfish value property. A mapping of this element to a JSON output property does not occur unless the <b>mapNullValue</b> property is set to <b>true</b>.</li> </ul>	String	options
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapArrayFirstElementValue	No	<p>If set to <b>true</b>, then if an array is encountered in the traversal of the input, only the first element is used. Set to <b>false</b> if you want the map policy to use all array elements. The default value is <b>false</b>.</p>	Boolean	options
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapResolveApicVariables	No	<p>If set to <b>true</b>, API Connect variable references found in the map properties are resolved. Set to <b>false</b> if you want the map policy to ignore API Connect variable references in the map policies. The default value is <b>true</b>.</p>	Boolean	options
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapNullValue	No	<p>If set to <b>true</b>, an input property value with a null value is mapped to the output document. Set to <b>false</b> if you want the map policy to ignore null input values. The default value is <b>false</b>.</p>	Boolean	options
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapOptimizeSchemaDefinition	No	<p>If set to <b>true</b>, complex schema types evaluation handles circular type references in an optimized manner. Set to <b>false</b> to evaluate these schema types in a standard manner. The default value is <b>false</b>.</p>	Boolean	options
<div style="border: 1px solid green; padding: 2px; display: inline-block;">API Gateway only</div> mapEmulateV4DefaultRequiredProps	No	<p>If set to <b>true</b>, default values are generated in the output for required properties that are either not mapped, or for which there is no input data present, in the following specific cases:</p> <ul style="list-style-type: none"> <li>An array consists of objects that contain one or more required properties.</li> <li>An object which is optional has one or more child properties that are required.</li> </ul> <p>By default, these required properties are not present in the output. If you set this property to <b>true</b>, these required properties will be present in the output. If the output schema defines a <b>default</b> property for the output property then the specified default value is used, otherwise default value is assigned dependent on the data type, as follows:</p> <ul style="list-style-type: none"> <li>String: empty string ("")</li> <li>Number: 0</li> <li>Boolean: false</li> <li>Object: empty object</li> <li>Array: empty array</li> </ul> <p>Example 1</p> <p>The input data has the following array of objects:</p> <pre>[{"a": "value1"}, {"a": "value2", "b": "value3"}]</pre> <p>The output schema defines the output object as having two properties, <b>a</b> and <b>b</b>, of which <b>b</b> is required. The map policy defines the following mappings:</p> <ul style="list-style-type: none"> <li><b>input.array.a</b> to <b>output.array.a</b></li> <li><b>input.array.b</b> to <b>output.array.b</b></li> </ul> <p>If this property is set to <b>true</b>, and <b>b</b> is either not mapped or has no input data present, then <b>b</b> is assigned a default value of an empty string, and the output is as follows:</p> <pre>[{"a": "value1", "b": ""}, {"a": "value2", "b": "value3"}]</pre> <p>Example 2</p> <p>The output schema defines the following structure:</p> <pre>{"a" : {"b" : {"c" : "value1", "d" : "value2"} } }</pre> <p>Property <b>b</b> is optional but property <b>d</b> within <b>b</b> is required.</p> <p>The map policy defines a mapping to <b>output.a.b.c</b>.</p> <p>If this property is set to <b>true</b>, and <b>d</b> is not mapped, then <b>d</b> is assigned a default value of an empty string, and the output is as follows:</p> <pre>{"a" : {"b" : {"c" : "value1", "d" : ""} } }</pre> <p>If this property is set to <b>false</b>, these required properties are <b>not</b> created in the output with their default values.</p> <p>The default value is <b>false</b>.</p>	Boolean	options

Property	Required	Description	Data type	Belongs to
<span>API Gateway only</span> mapEnablePostProcessingJSON	No	If set to <b>true</b> , mapped JSON output is post processed. The post processing of JSON output will use the output schema to ensure that property values are of the same data type as that defined in the schema. It will also normalize output property values that have a Badgerfish JSON syntax due to object mapping of an XML input. Set to <b>false</b> if you want no post processing of mapped JSON output. The default value is <b>false</b> .	Boolean	options
<span>API Gateway only</span> mapCreateEmptyArray	No	This property controls how the map policy handles the output of an empty array. Specify one of the following values: <ul style="list-style-type: none"> <li><b>all</b>: Output all empty arrays, including empty children arrays.</li> <li><b>parent</b>: Output only the current property's empty array value. Children map actions of this property are not attempted.</li> <li><b>none</b>: Prevent any empty output array values from being produced.</li> </ul> The default value is <b>all</b> .	String	options
<span>API Gateway only</span> mapReferenceLimit	No	Set the value of this property to an integer value that specifies the maximum allowed number of iterations of a circular schema definition. The default value is 1, which means that circular schema definitions are not followed. The maximum possible value is 5. If you specify a value greater than 5, a value of 5 is assumed. If you specify a non-numeric value, a value of 1 is assumed.	String	options
messagesInputData	No	This property defines the severity level for log messages that relate to input data. Specify one of the following values: <ul style="list-style-type: none"> <li><b>error</b></li> <li><b>warn</b></li> <li><b>info</b></li> </ul>	String	options
<span>API Gateway only</span> mapEmulateV4EmptyJSONObject	No	If a mapping fails because its input is not present and there is no default mapping configured, the default behavior is to not to make any change to the output mapping. Set this property to <b>true</b> to create an empty object for the parent of the target mapping, emulating the behavior of IBM® API Management Version 4.0.  Example The map policy defines a mapping to <b>output.a.b.c</b> . If input data is present, the output is as follows: <pre> {   "a": {     "b": {       "c": "inputvalue"     }   } } </pre> If there is no input data, and the mapEmulateV4EmptyJSONObject option is set to <b>true</b> , the output is as follows: <pre> {   "a": {     "b": {     }   } } </pre> Properties <b>a</b> and <b>b</b> are created but the value of <b>b</b> is an empty object. The default value is <b>false</b> .	Boolean	options

<sup>1</sup> There must be one of \$ref or type in the description of a variable.

<sup>2</sup> There must be one of set or create in an actions field.

## Example

```

- map:
  version: 1.0.0
  title: Output mapping
  inputs:
    Monthly_cost:
      schema:
        type: double
      variable: loan_invoke.body.monthly_payment
      content: application/json
    Duration:
      schema:
        type: integer
      variable: request.parameters.duration
  outputs:
    Quote_Output:
      schema:
        $ref: '#/definitions/Quote_Output'
      variable: message.body
      content: application/json
  actions:
    - set: Quote_Output.monthly_repayment
      from: Monthly_cost

```

```

value: ''
- set: Quote_Output.total_cost
  from:
    - Duration
    - Monthly_cost
  value: '$(Duration)*$(Monthly_cost)'
description: Maps and transforms contexts to the operation output.
options:
  includeEmptyXMLElements: false
  namespaceInheritance: false
  inlineNamespaces: false
  API Gateway only mapResolveXMLInputDataType: true
  API Gateway only mapXMLEmptyElement: null
  API Gateway only mapArrayFirstElementValue: false
  API Gateway only mapResolveApicVariables: false
  API Gateway only mapNullValue: true
  API Gateway only mapOptimizeSchemaDefinition: true
  API Gateway only mapCreateEmptyArray: parent
  API Gateway only mapReferenceLimit: 5
messagesInputData: warn
mapEmulateV4EmptyJSONObject: true

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## operation-switch

Use the operation-switch construct when you want to execute alternative policy assemblies, conditional on the operation that is being called.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0, functionality provided by <a href="#">switch</a>	

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [operation-switch](#).

## About

An operation can be described with a **verb/path** pair, or with an **operationId**. The **operationIds** are strings, or names, that are defined in the OpenAPI document.

The operation-switch policy has the following format:

```

- operation-switch:
  version: version
  title: title
  description: description
  case:
    - operations:
      - verb: operation_verb_1_1
        path: operation_path_1_1
      - verb: operation_verb_1_2
        path: operation_path_1_2
      .
      .
      further verb/path combinations
      .
      .
    execute:
      policy_assembly_1 ...
    - operations:
      - verb: operation_verb_2_1
        path: operation_path_2_1
      - verb: operation_verb_2_2
        path: operation_path_2_2
      .
      .
      further verb/path combinations
      .
      .

```

```

execute:
  policy_assembly_2 ...
  .
  .
- operations:
- operationID_3_1
- operationID_3_2
- operationID_3_3
  .
  .
  further operationIDs
  .
  .
execute:
  policy_assembly_3 ...
  .
  .
  further operations sections
  .
  .

```

For each **operations:** section, if the operation that is being called matches any of the **verb/path** combinations or **operationId** strings listed in that **operations:** section, then the policy assembly that is defined in the **execute:** section is executed. Therefore, each **operations:** section defines execution of a policy assembly conditional on the operation that is being called.

You can have as many **operations:** sections as you want, and each **operations:** section can have one or more **verb/path** combinations or **operationId** strings.

The **execute:** section can define any policy assembly, including further operation-switch policies.

## Properties

Table 2. operation-switch policy properties

Property	Required	Description	Data type	
version	Yes	The policy version number	string	
title	No	A title for the policy.	string	N/A
description	No	A policy description.	string	N/A
case	Yes	An array containing different cases, each entry contains an operations and execute field.	array (object)	N/A
operations	Yes	The operations to which a case applies.	object	case
verb	No	An operation verb. Valid values: <ul style="list-style-type: none"><li>• GET</li><li>• POST</li><li>• PUT</li><li>• DELETE</li><li>• HEAD</li><li>• PATCH</li><li>• OPTIONS</li></ul>	string	operations
path	No	A relative path to an individual endpoint. For example, <code>/account_status</code> .	string	operations
operationID	No	The operation is defined in OpenAPI. The policy operation refers to the OpenAPI operation.	string	operations
execute	Yes	The policy assembly that you want to execute if the operation that is being called matches any one of the verb/path combinations. For more information, see <a href="#">execute</a> .	string	case

## Example

Example that defines match operations with **verb/path** combinations:

**# carry out different redaction actions depending on the operation**

```

- operation-switch:
  version: 1.0.0
  title: clear_region_and_set_body
  case:
  - operations:
    - verb: GET
      path: /account_details
      execute:
        - redact:
            title: remove secret field
            actions:
              - action: remove
                from: all
                path: /document/user/secret
  - operations:
    - verb: GET
      path: /account_status
      execute:
        - redact:

```

```

    title: redact address
    actions:
      - action: redact
        from: response
        path: /**[@name='secondaryAddress']/**[@name='streetAddress']

```

Example that defines match operation with `operationIDs`:

```

# match on operationIDs

- operation-switch:
  title: customer_actions
  case:
    - operations:
      - getCustomerByName
      - deleteCustomer
      - addACustomer
    execute:
      .
      .
      .

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## oauth

Use the oauth policy to policy to perform OAuth processing based on defined OAuth provider settings.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [OAuth](#).

## About

The oauth policy has the following format:

```

- oauth:
  version: version
  title: title
  description: description
  oauth-provider-settings-ref:
    .
    .
    .
  references_to_oauth_settings
    .
    .
    .
  supported-oauth-components:
    - oauth_component_1
    - oauth_component_2
    .
    .
    .

```

Note: You add an oauth policy to the OpenAPI source in a native OAuth provider. For more information, see the following topics:

- [Editing the native OAuth provider configuration using the API Editor](#) (Cloud Manager UI)
- [Editing the native OAuth provider configuration using the API Editor](#) (API Manager UI)

## Properties

Table 2. oauth policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A description of the policy.	string
oauth-provider-settings-ref: default	Yes	The name of an existing OAuth provider that defines the required settings.	string



Property	Required	Description	Data type
<code>oauth-provider-settings-ref:url</code>	No	A URL to a document that contains serialized XML or JSON properties that defines OAuth token generate settings. URL reference takes precedence over any existing literal configuration or object reference.	string
<code>oauth-provider-settings-ref:literal</code>	No	A literal string that contains serialized XML or JSON properties that defines OAuth token generate settings. Literal configuration takes precedence over any existing object reference.	string
<code>supported-oauth-components:</code> - <code>  oauth_component_1</code> - <code>  oauth_component_2</code> . .	Yes	Specify the OAuth components that are supported by this policy, as follows:  - OAuthValidateRequest Validates the authorization request from the client. - OAuthGenerateAZCode Generates the authorization code for the client, which represents the resource owner's authorization that grants access to the requested resource. - OAuthVerifyAZCode Verifies the authorization code from the client. - OAuthVerifyRefreshToken Verifies the refresh token that is presented by the client. - OAuthCollectMetadata Collects metadata about a client for later user interaction - OAuthGenerateAccessToken Generates the access token to the client when the authorization code or refresh token is verified. - OAuthIntrospectToken Introspects the token to determine its state and, when active, its metadata.	string

## Overriding default OAuth provider settings

You can use either the `literal` property or the `url` property to dynamically override any OAuth provider configuration settings to dynamically override any OAuth provider configuration settings defined by the `default` property.

For example, to override the access token expiration time with a value of 200 seconds, include the following configuration in either the literal string or the document at the specified URL:

```
<OAuthProviderSettings><APICAccessTokenTTL>200</APICAccessTokenTTL></OAuthProviderSettings>
```

For a list of all OAuth provider settings, refer to the `OAuthProviderSettings` management schema, defined in the `xml-mgmt.xsd` file located in the store: directory on the DataPower API Gateway.

If you are using the API Manager user interface, the connection details are determined by the API Manager URL that you open, and the user ID with which you log in. If you are using the API Designer user interface, you provide the management server details and user ID in the login window that opens when you first launch API Designer; see [Logging into API Connect Designer](#).

## oauth policy example

```
- oauth:
  version: 2.0.0
  title: my-oauth-policy
  oauth-provider-settings-ref:
    default: my-oauth
  supported-oauth-components:
    - OAuthGenerateAZCode
    - OAuthGenerateAccessToken
    - OAuthIntrospectToken
    - OAuthVerifyAZCode
    - OAuthVerifyRefreshToken
```

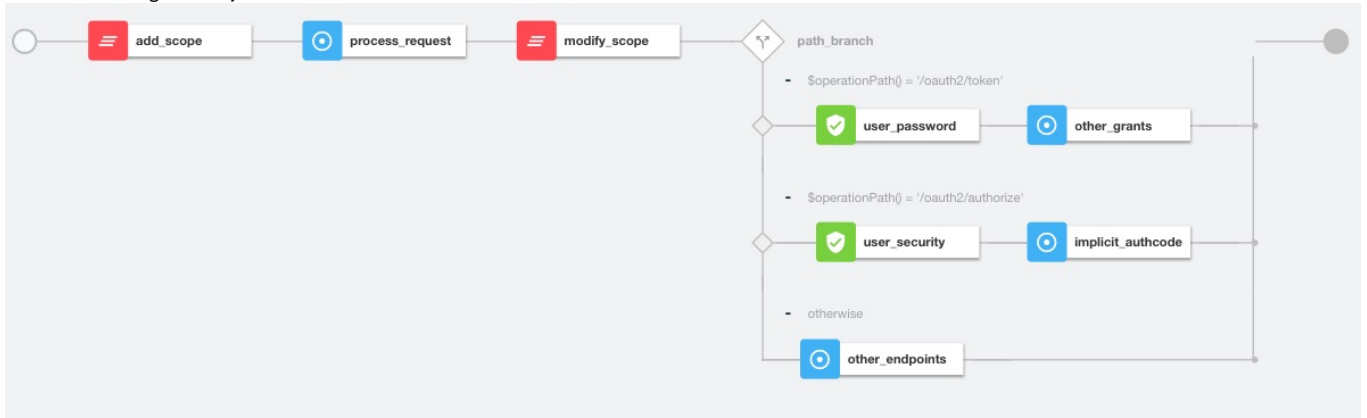
**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Example - using multiple OAuth policies in an OAuth provider assembly

This example demonstrates the use of multiple OAuth policies in the assembly flow for a native OAuth provider.

The example is based on the default assembly that is generated when you create a native OAuth provider, and is customized with the addition of `gatewayscript` policies that use [OAuth context variables](#) to manipulate the OAuth flow. For details on creating a native OAuth provider, see [Configuring a native OAuth provider](#).

It has the following assembly flow:



The following sections describe the OpenAPI source code that underlies each of the policies in the assembly; for the complete assembly code, download [multiple\\_oauth\\_policies.txt](#).

## Sample policy to add a custom scope

The `add_scope` policy is a `gatewayscript` policy that adds a custom scope to the request.

The underlying OpenAPI source YAML is as follows:

```
- gatewayscript:
  version: 2.0.0
  title: add_scope
  source: |-
    // Add another custom scope to the request
    let scope = context.get("request.parameters.scope.values[0]");
    if (scope)
      context.set("oauth.processing.scope", scope + " custom");
```

## Validate the initial OAuth request

The first `process_request` policy is an `oauth` policy that processes the initial request and verifies that the request is valid. The result of the processing is stored automatically in the `oauth.processing` context variables for use, as required, by the next OAuth policy in the assembly flow.

The underlying OpenAPI source YAML is as follows:

```
- oauth:
  title: process_request
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect protocol steps
    that are needed for OAuth Validation by default. The inputs and
    outputs of each of the steps are driven by documented context
    variables. Add or remove the Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthValidateRequest
```

## Sample policy to modify the scope

The `modify_scope` policy is a `gatewayscript` policy that modifies the scope depending on the calling application.

The underlying OpenAPI source YAML is as follows:

```
- gatewayscript:
  version: 2.0.0
  title: modify_scope
  source: |-
    let admin_id = '1f1a2aa4-db9f-4423-b2f1-e2572b12123a';

    // Check application and modify the scope
    let app = context.get("oauth.processing.client_id");
    let scope = context.get("oauth.processing.scope");
    if (app === admin_id) {
      context.set("oauth.processing.scope", scope + " admin");
    } else {
      context.set("oauth.processing.scope", scope + " customer");
    }
  }
```

## Branch conditionally according to the OAuth path

The `path_branch` policy is a `switch` policy that branches according to the different OAuth paths to process the resource owner.

The underlying OpenAPI source YAML is as follows:

```

- switch:
  version: 2.0.0
  title: path_branch
  case:
    - condition: ($operationPath() = '/oauth2/token')
      execute:
        .
        .
        .
        definition of the user_security and other_grants policies
        .
        .
    - condition: ($operationPath() = '/oauth2/authorize')
      execute:
        .
        .
        .
        definition of the user_password and implicit_authcode policies
        .
        .
    - otherwise:
        .
        .
        .
        definition of the other_endpoints policy
        .
        .

```

## Process the user name and password, and enable grant type component

The following two policies operate on the token endpoint.

- The `user_password` policy for password grant type processes the user name and password from the `x-www-form-urlencoded` body. The authentication method is derived from the User Security settings in the OAuth provider; see [Configuring user security for a native OAuth provider](#).
- The `other_grants` policy is an `oauth` policy that enables the `OAuthGenerateAccessToken`, `OAuthVerifyAZCode`, `OAuthVerifyRefreshToken`, and `OAuthCollectMetadata` components to perform the operations for client credentials, authorization code, refresh token, and password grant types.

The underlying OpenAPI source YAML is as follows:

```

- user-security:
  title: user_password
  version: 2.0.0
  description: ''
  factor-id: default
  extract-identity-method: context-var
  user-context-var: request.parameters.username.values
  pass-context-var: request.parameters.password.values
  ei-stop-on-error: false
  user-auth-method: auth-url
  au-stop-on-error: false
  auth-url: 'http://httpbin.org/basic-auth/user/pass'
  user-az-method: authenticated
  az-stop-on-error: true
  auth-response-headers-pattern: (?)*x-api*
  auth-response-header-credential: X-API-Authenticated-Credential
- oauth:
  title: other_grants
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect
    protocol steps that are needed for token path by default.
    The inputs and outputs of each of the steps are driven by
    documented context variables. Add or remove the Supported
    OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthGenerateAccessToken
    - OAuthVerifyAZCode
    - OAuthVerifyRefreshToken
    - OAuthCollectMetadata

```

## Perform authorization checks, and enable grant type components

These following two policies operate on the authorize endpoint.

- The `user_security` policy configuration is derived from the User Security settings in the OAuth provider; see [Configuring user security for a native OAuth provider](#).
- The `implicit_authcode` policy is an `oauth` policy that enables the `OAuthGenerateAZCode`, `OAuthGenerateAccessToken`, and `OAuthCollectMetadata` components to perform the operations for the implicit and authorization code grant types.

The underlying OpenAPI source YAML is as follows:

```

- user-security:
  title: user_security
  version: 2.0.0
  description: >-
    This user security policy performs EI(basic) and AU(auth

```

```

    url) check for oauth assembly. Change the security check
    method as required
    factor-id: default
    extract-identity-method: basic
    ei-stop-on-error: true
    user-auth-method: auth-url
    au-stop-on-error: true
    user-az-method: authenticated
    az-stop-on-error: true
    auth-response-headers-pattern: (?x-api*
    auth-response-header-credential: X-API-Authenticated-Credential
    auth-url: 'http://httpbin.org/basic-auth/user/pass'
- oauth:
  title: implicit_authcode
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect
    protocol steps that are needed for az code path by
    default. The inputs and outputs of each of the steps are
    driven by documented context variables. Add or remove the
    Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthGenerateAZCode
    - OAuthGenerateAccessToken
    - OAuthCollectMetadata

```

## Process all other endpoints

The **otherwise** condition catches all other endpoints such as the introspect and revoke endpoints. The **other\_endpoints** policy in the **otherwise** condition is an **oauth** policy that enables the **OAuthIntrospectToken**, and **OAuthRevokeTokencomponents** components to perform the operations for introspect and revoke.

The underlying OpenAPI source YAML is as follows:

```

- oauth:
  title: other_endpoints
  version: 2.0.0
  description: >-
    This oauth policy performs all OAuth/OpenID Connect
    protocol steps that are needed for all other paths by
    default. The inputs and outputs of each of the steps are
    driven by documented context variables. Add or remove the
    Supported OAuth Components as required.
  oauth-provider-settings-ref:
    default: custom-form
  supported-oauth-components:
    - OAuthIntrospectToken
    - OAuthRevokeToken

```

## Related concepts

- [API policies and logic constructs](#)

## Related reference

- [OAuth context variables](#)

## Related information

- [Configuring a native OAuth provider](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## parse

Use the parse policy to control the parsing of an input document.

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Parse](#).

## About

When the input document is a JSON string, the string is parsed instead of copied over. You can re-parse a JSON string as any of the allowed doc types (`detect`, `xml`, `json`, `binary`). For example, suppose that the JSON string "`<a></b></a>`" is parsed. You can re-parse it using doc type `xml` to extract the XML from that string.

The parse policy has the following format:

```
- parse:
  version: version
  title: title
  description: description
  use-content-type: request_header_usage_setting
  parse-settings-reference:
    .
    .
    .
    references_to_parse_settings
    .
    .
    .
  input: input_message
  output: output_message
```

## Properties

Table 2. parse policy properties

Property	Required	Description	Data type
version	Yes	The policy version number.	string
title	No	The title of the policy.	string
description	No	A description of the policy.	string
use-content-type	No	If this setting is enabled and the parse setting is configured to detect the document type, the parse action uses the <b>Content-Type</b> specified in the request headers. If this setting is enabled and the document type in the parse setting is configured for either JSON or XML, the parse action uses the <b>Content-Type</b> specified in the request headers and fails if the <b>Content-Type</b> in the request headers does not match the parse setting.  Enabling this setting is applicable only when the expected <b>Content-Type</b> is either JSON or XML.  If this setting is not enabled, the parse action uses either the document type specified in the parse setting, or the detected document type if the parse setting is configured to detect the document type.  The default value is <b>false</b> .	boolean
parse-settings-reference: default	No	An existing valid object from which to retrieve default property values for the dynamic object.	string
parse-settings-reference: literal	No	A literal string as serialized XML or JSON properties that are merged into the dynamic object. The literal property overrides any existing default properties.	string
parse-settings-reference: url	No	A URL that represents a named context from which to retrieve the serialized XML or JSON properties that are merged into the dynamic object. These properties override any existing literal or default properties.	string
input	No	The name of a variable in the API context. The content of the <b>body</b> field of the variable is the input to the policy. The default variable name is <b>message</b> .	string
output	No	The name of a variable in the API context. The content of the body field of the variable is the output of the parse operation. The parse metrics of the parsed document can be stored in a different part of the message. The default variable name is the same as the input name, so by default the input message is overwritten by the output message.	string

## parse policy example

```
- parse:
  version: 2.0.0
  title: my-parse-policy
  use-content-type: true
  parse-settings-reference:
    default: my-parse
  input: input-message
  output: output-message
```

## parse-settings-reference: literal example 1

In XML:

```
- parse:
  version: 2.0.0
  title: parse
  parse-settings-reference:
    default: apic-default-parsesettings
    literal: <ParseSettings><DocumentType>xml</DocumentType></ParseSettings>
```

In JSON:

```
- parse:
  version: 2.0.0
  title: parse
  parse-settings-reference:
  default: apic-default-parsesettings
  literal: { \"ParseSettings\" : { \"DocumentType\" : xml } }
```

## parse-settings-reference: literal example 2

You can use the literal to reduce the maximum allowed payload for a particular API by setting the DocumentSize value. By default, API Connect sets the maximum payload for an API to approximately 4.2 MB. The following example lowers the maximum to 2 MB (2000000 bytes):

In XML:

```
- parse:
  version: 2.0.0
  title: parse
  parse-settings-reference:
  default: apic-default-parsesettings
  literal: <ParseSettings><DocumentSize>2000000</DocumentSize></ParseSettings>
```

In JSON:

```
- parse:
  version: 2.0.0
  title: parse
  parse-settings-reference:
  default: apic-default-parsesettings
  literal: { \"ParseSettings\" : { \"DocumentSize\" : 2000000 } }
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## proxy

Apply the proxy policy to proxy another API within your operation, particularly if you need to call a large payload.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, functionality provided by <a href="#">invoke</a>	

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Proxy](#).

## About

Keep the following considerations in mind regarding the proxy policy.

- Only one proxy policy is permitted to be called per assembly.
- More than one proxy policy can be applied, if they are contained in mutually exclusive branches of the assembly.
- You can use the proxy policy to return multipart form data, that is, when the response is set to Content-Type: multipart/related. However, proxy must be the final policy in the assembly, otherwise the response that is received is manipulated causing the multipart form data to be lost.
- The proxy policy, if inside a conditional policy, must be the **final** policy to be executed in the API. If you need further processing afterward, use the [invoke](#) policy rather than the proxy policy.
- The Proxy policy does not currently attempt to rewrite a Location header that is returned from the back end.

The proxy policy has the following structure:

```
- proxy:
  version: version
  title: title
  description: description
  target-url: URL_of_target_API
  tls-profile: TLS_profile_to_be_used
  verb: method_type
  http-version: HTTP_version
  timeout: timeout_value_in_seconds
  compression: is_data_to_be_compressed
  username: username_if_authentication_required
  password: password_if_authentication_required
  output: location_of_the_proxy_result
  cache-key: unique_identifier_of_the_document_cache_entry
  cache-response: cache_behavior
  cache-putpost-response: response_caching_behavior
```

```
cache-ttl: cache_time_to_live
stop-on-error: errors_that_stop_the_flow
```

## Properties

The table describes the properties of the proxy policy.

Table 2. Proxy policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
target-url	Yes	The URL of the target API.	string
tls-profile	No	The TLS profile to be used.	string
verb	No	The operation method type.  Valid values: <ul style="list-style-type: none"> <li>• Keep</li> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• PATCH</li> <li>• HEAD</li> <li>• OPTIONS</li> </ul> The default value is <b>Keep</b> .	string
http-version	No	The HTTP version. The default value is <b>1.1</b> .	string
timeout	No	The timeout value in seconds. The default value is <b>60</b> .	integer
compression	No	Specifies whether data is to be compressed by using <b>gzip</b> before it is uploaded. The default value is <b>false</b> .	boolean
username	No	The user name, if authentication is required.	string
password	No	The password, if authentication is required.	string
output	No	Specifies the location of the proxy result. By default, the proxy result, that is the body, headers, statusCode and statusMessage, is saved in the variable <b>context.message</b> . The assembly developers can specify an additional location to store the proxy result with the output property.	string
cache-key	No	Specifies the unique identifier of the document cache entry.	string
cache-response	No	The cache response type. Valid values: <ul style="list-style-type: none"> <li>• protocol: The cache behavior is defined by the Cache-Control headers on the request and response.</li> <li>• no-cache: Specifies that there is no caching. However, if the document is already in the cache, the document is retrieved from the cache.</li> <li>• time-to-live: Specifies that the response stays in the cache for the specified time.</li> </ul> The default value is <b>protocol</b> .	string
cache-putpost-response	No	Specifies whether to cache the response from POST and PUT requests. Caching the response from POST and PUT requests can reduce server load and reduce latency in the response to the client request. The default value is <b>false</b> .	boolean
cache-ttl	No	Specifies the amount of time in seconds that the response stays in the cache. Applies only if the property cache-response is set to <b>time-to-live</b> . Enter a value in the range 5 - 31708800. The default value is <b>900</b> .	integer
stop-on-error	No	List the errors that, if thrown during the policy execution, cause the flow to stop. If there is a <b>catch</b> flow configured for the error, it is triggered to handle the error thrown. If an error is thrown and there are no errors specified for the Stop on error property, or if the error thrown is not one of the specified errors, the policy execution is allowed to complete, and the assembly flow continues.	string

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

API Gateway only

## ratelimit

Use the ratelimit policy to apply one or more rate or burst limits at any point in your API assembly flow. Rate and burst limits restrict the number of calls made to an API in a specified time period.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Rate Limit](#).

## About

The defined rate and burst limits are applied to whatever follows in the assembly flow. For example, if a ratelimit policy is placed before an invoke policy, and the call made by the invoke policy exceeds the limits defined by the ratelimit policy, the API call itself fails.

The ratelimit policy has the following format:

```
- ratelimit:  
  version: version  
  title: title  
  description: description  
  source: rate_and_burst_limit_location  
  rate-limit: rate_limits_to_apply  
  burst-limit: burst_limits_to_apply
```

## Properties

The following table describes the properties of the ratelimit policy.

Table 2. ratelimit policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
source	Yes	The location of all the rate limit and burst limit definitions that are included in this policy. Specify one of the following values: <ul style="list-style-type: none"><li><b>catalog-named:</b> the rate or burst limits to be applied are defined in the appropriate <b>api-collection</b> object on the DataPower API Gateway, which is the object that represents your API Connect Catalog in the gateway configuration. For details of how to configure a rate or burst limit in the <b>api-collection</b> object, see <a href="#">Configuring a rate or burst limit on the DataPower API Gateway</a>.</li><li><b>plan-default:</b> the rate and burst limits that are applied are the default ones configured in the Plan to which the calling application is subscribed. For details on how to configure default Plan rate and burst limits, see <a href="#">Editing a draft Product</a>.</li></ul>	string
rate-limit	Yes*	A list of rate limit names as defined in the DataPower API Gateway configuration. * You must provide at least one rate limit or at least one burst limit.	string
burst-limit	Yes*	A list of burst limit names as defined in the DataPower API Gateway configuration. * You must provide at least one rate limit or at least one burst limit.	string

## Example

```
- ratelimit:  
  version: 2.0.0  
  title: Apply rate and burst limits  
  source: catalog-named  
  rate-limit:  
    - 30perMinute  
    - 2000per3Hours  
  burst-limit:  
    - 5per10Seconds
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## redact - DataPower API Gateway

Use the redact policy to completely remove or to redact specified fields from the request body, the response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.

## Gateway support

Note: This page describes the redact policy implementation in the DataPower® API Gateway. If you are using the DataPower Gateway (v5 compatible), see [redact - DataPower Gateway \(v5 compatible\)](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version



Gateway	Policy version
DataPower API Gateway, policy available from V2018.4.1.7	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Redaction - DataPower API Gateway](#).

## About

The ratelimit policy has the following format:

```
- redact:
  version: version
  title: title
  description: description
  redactions:
    - action: remove_or_redact
      path: JSONata_expression_for_field_to_remove_or_redact
      .
      .
      .
  root: content_source
```

Note: With the DataPower API Gateway, the input to the redact policy must be parsed data. One way to produce parsed data is to use a [parse](#) policy before a redact policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table describes the properties of the redact policy.

Table 2. redact policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
root	No	Specifies the data source that contains the content to which the redact or remove action applies. If the root property is omitted, the action is applied to the entire API context. You can use any supported JSONata path expression.  If you want to apply the action to either request or response data, specify a value of <code>message.body</code> . The actual content to which the action is applied then depends on the positioning of the redact policy in the overall assembly flow; for example: <ul style="list-style-type: none"> <li>• If positioned at the beginning, the action is applied to the client request.</li> <li>• If positioned after an invoke policy, the action is applied to the response from the back end.</li> <li>• If positioned at the end, the action is applied to the response that is returned to the client.</li> </ul> If, in your assembly flow, the redact policy is used after a <a href="#">log</a> policy that specifies <code>gather-only</code> for the mode property, specify a root value of <code>log.request_body</code> for the logged request payload, or <code>log.response_body</code> for the logged response payload.	string
path	Yes	Specifies a JSONata path expression that identifies the fields to redact or remove from the source. For more information, see <a href="#">Constructing JSONata expressions to redact fields</a>	string
action	Yes	Specifies whether you want to remove or redact the content. Supply one of the following values: <ul style="list-style-type: none"> <li>• <code>remove</code>: Completely removes the specified fields.</li> <li>• <code>redact</code>: Redacts (obfuscates with "*"s) the fields to block out the data.</li> </ul> The default value is <code>redact</code> .  Note: If a numerical value is being redacted, the redacted value is depicted as <code>*****</code> and the type is changed to <code>string</code> .	string

## Example

# Specify separate remove and redact actions

```
- redact:
  version: 2.0.0
  title: remove price, redact author
  redactions:
    - action: remove
      path: xpath($, '//price')
    - action: redact
      path: $.**.author"
  root: message.body
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# redact - DataPower Gateway (v5 compatible)

Use the redact policy to completely remove or to redact specified fields from the request body, the response body, and the activity logs. You might find this policy useful for removing or blocking out sensitive data (for example, credit card details) for legal, security, or other reasons.

## Gateway support

Note: This page describes the redact policy implementation in the DataPower® Gateway (v5 compatible). If you are using the DataPower API Gateway, see [redact - DataPower API Gateway](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Redaction - DataPower Gateway \(v5 compatible\)](#).

## About

The redaction policy has the following format:

```
- redact:
  version: version
  title: title
  description: description
  actions:
    - action: remove_or_redact
      from:
        - where the redaction is to be applied
          path: XPath_expression_for_field_to_remove_or_redact
          .
          .
          further action/from/path combinations
          .
          .
```

You can specify as many **action/from/path** combinations as you want.

## Properties

The following table describes the policy properties:

Table 2. redact policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
action	No	Specifies whether you want to remove or redact the fields. Valid values: <ul style="list-style-type: none"><li><b>remove</b>: Completely removes the specified fields.</li><li><b>redact</b>: Redacts (obfuscates with "*"s) the fields to block out the data.</li></ul> The default value is <b>redact</b> . Note: If a numerical value is being redacted, the redacted value is depicted as <b>*****</b> and the type is changed to <b>string</b> .	string
from	No	Determines where the redaction is to be applied. Valid values: <ul style="list-style-type: none"><li><b>all</b>: Apply the redaction to the request body, the response body, and the activity logs.</li><li><b>request</b>: Apply the redaction to the request body only.</li><li><b>response</b>: Apply the redaction to the response body only.</li><li><b>logs</b>: Apply the redaction to the activity logs only.</li></ul> You can supply one or more values. The default value is <b>all</b> .	string
path	Yes	Specifies an XPath expression that defines the fields to remove or redact. You can construct an XPath expression that is based on JSON or XML depending on whether your API requests and responses use a JSON or an XML format. If the payload is JSON, use the DataPower XML representation of the JSON content (JSONx) to construct the expression. Note: Use a JSONx representation only to identify the XPath expressions for the fields to remove or redact. Do not change the format of any response bodies in API Manager. To learn more about constructing XPath expressions that are based on JSON or XML, see <a href="#">Constructing XPath expressions to redact fields</a> .	string

## Example

```
# Specify separate remove and redact actions

- redact:
  version: 1.0.0
  title: remove secret field, redact address
  actions:
    - action: remove
      from:
        - all
      path: /document/user/secret
    - action: redact
      from:
        - request
        - response
      path: /**[@name='secondaryAddress']/*[@name='streetAddress']
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## set-variable

Use the set-variable policy to set the value of a runtime variable, or to add or clear a runtime variable.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Set Variable](#).

## About

The set-variable policy has the following format:

```
- set-variable:
  version: version
  title: title
  description: description
  actions:
    - action_type: variable_name
      value: value
      type: data_type
```

## Properties

Table 2. set-variable policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
actions	Yes	Lists the actions to be performed by the set-variable policy.	array
set	Yes <sup>1</sup>	For setting a variable. specifies the name of the variable that you want to set. <sup>1</sup> One of the properties set, add, or clear is required.	string
add	Yes <sup>2</sup>	For adding a variable. specifies the name of the variable that you want to add. <sup>2</sup> One of the properties set, add, or clear is required.	string
clear	Yes <sup>3</sup>	For clearing a variable. specifies the name of the variable that you want to clear. <sup>3</sup> One of the properties set, add, or clear is required.	string
value	Yes <sup>4</sup>	Allocates this value to the specified variable. Can be a literal value, or another variable. <sup>4</sup> value is required only when set or add is specified as the action.	string

Property	Required	Description	Data type
<b>API Gateway only</b> type	Yes	Specifies the data type of the variable. Valid values: <ul style="list-style-type: none"> <li>• any</li> <li>• string</li> <li>• number</li> <li>• boolean</li> </ul> For all values other than <b>any</b> , the value is validated against the specified data type. If the data type is specified as <b>boolean</b> , the value property must be set to <b>true</b> or <b>false</b> .	string

## Example 1

```
# clear a variable

set-variable:
  version: 1.0.0
  title: clear_region
  actions:
    - clear: message.headers.region
```

## Example 2

```
# set a variable to the value of an API Gateway context variable

set-variable:
  version: 2.0.0
  title: set content type
  actions:
    - set: message.headers.contenttype
      value: $(message.headers.content-type)
      type: string
```

## Example 3

```
# add a variable

assembly:
  execute:
    - set-variable:
        version: 2.0.0
        title: set-variable
        actions:
          - value: testing add
            add: message.headers.jja
            type: string
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## switch

Use the switch component to execute one of a number of sections of the assembly based on which specified condition is fulfilled.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [switch](#).

## About

The switch policy has the following format:

```
- switch:
  version: version
  title: switch
  description: 'Description'
  case:
    - condition: Script_1
```

```



execute:
  Assembly_Section_1
- condition: Script_2
  execute:
  Assembly_Section_2
- otherwise:
  Assembly_Section_3

```

The **execute**: section can define any policy assembly, including further switch policies. For more information, see [execute](#).

## Properties

Table 2. switch policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
case	Yes	Contains the condition and execute pairs of the switch policy.	string
condition	Yes (one or more)	A script that returns <b>true</b> or <b>false</b> .  Use the JSONata expression language to define your condition. See <a href="#">Writing switch condition scripts - DataPower API Gateway</a> .  Use GatewayScript to define your condition. See <a href="#">Writing switch condition scripts - DataPower Gateway (v5 compatible)</a> .	string
execute	Yes (one per condition)	The policy assembly that you want to execute if the condition returns <b>true</b> . For more information, see <a href="#">execute</a> .	string
otherwise	No	The case you want to execute if no other cases are fulfilled. It functions in the same manner as an execute property. For more information, see <a href="#">execute</a> .	string

For examples, see one or other of the following topics, depending on which gateway type you are using:

- [Writing switch condition scripts - DataPower API Gateway](#)
- [Writing switch condition scripts - DataPower Gateway \(v5 compatible\)](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Writing switch condition scripts - DataPower API Gateway

You write condition scripts for the switch policy in the DataPower® API Gateway by using the JSONata expression language.

The following JSONata functions are supported:

- Aggregation functions:
  - `$average (array)`
  - `$max (array)`
  - `$min (array)`
  - `$sum (array)`
- Array functions:
  - `$append (array1, array2)`
  - `$count (array)`
  - `$reverse (array)`
  - `$sort (array [, function])`
  - `$zip (array1, ...)`
- Boolean functions:
  - `$boolean (arg)`
  - `$exists (arg)`
  - `$not (arg)`
- Numeric functions:
  - `$abs (number)`
  - `$ceil (number)`
  - `$floor (number)`
  - `$formatBase (number [, radix])`
  - `$number (arg)`
  - `$power (base, exponent)`
  - `$round (number [, precision])`
  - `$sqrt (number)`
- Object functions:
  - `$keys (object)`
  - `$lookup (object, key)`
  - `$merge (array<object>)`
  - `$spread (object)`
- String functions:

- `$contains(str, pattern)`
- `$join(array[, separator])`
- `$length(str)`
- `$lowercase(str)`
- `$match(str, pattern [, limit])`
- `$pad(str, width [, char])`
- `$replace(str, pattern, replacement [, limit])`
- `$split(str, separator [, limit])`
- `$substring(str, start[, length])`
- `$substringAfter(str, chars)`
- `$substringBefore(str, chars)`
- `$trim(str)`
- `$uppercase(str)`

You can use the following JSONata numeric operators:

- + (addition)
- - (subtraction)
- \* (multiplication)
- / (division)
- % (modulo)

You can use the following JSONata comparison operators for number values or strings:

- =
- !=
- <
- >
- <=
- >=

You can also use the following functional extensions to standard JSONata notation. Each extension corresponds to a part of the API context.

Extension	Variable	Description
<code>\$header(name)</code>	<code>message.headers.name</code>	Message header
<code>\$httpVerb()</code>	<code>request.verb</code>	HTTP method of the request
<code>\$operationID()</code>	<code>api.operation.id</code>	ID of the operation
<code>\$operationPath()</code>	<code>api.operation.path</code>	Path of the operation
<code>\$queryParameter('name')</code>	<ul style="list-style-type: none"> <li>• <code>request.parameters.name.locations</code></li> <li>• <code>request.parameters.name.values</code></li> </ul>	Returns the value of a request query parameter given the parameter name. The function searches the <code>request.parameters.name.locations</code> array for the index position of the value <code>query</code> , and returns <code>request.parameters.name.values[index]</code> , where <code>index</code> is the index position. Parameter values are not URL decoded.
<code>\$statusCode()</code>	<code>message.status.code</code>	Status code
<code>\$storageType([arg])</code>	<p><code>variable.body</code></p> <p>You can specify any variable in the API context. When no variable is specified, the default variable <code>message.body</code> is used.</p>	Storage type of the message. The supported values are: <ul style="list-style-type: none"> <li>• <code>binary</code></li> <li>• <code>json</code></li> <li>• <code>stream</code></li> <li>• <code>xml</code></li> </ul>
<code>\$urlParameter('name')</code>	<ul style="list-style-type: none"> <li>• <code>request.parameters.name.locations</code></li> <li>• <code>request.parameters.name.values</code></li> </ul>	<p>Given a request parameter name, returns the parameter values for all occurrences of that parameter as a path parameter or query parameter.</p> <p>The function searches the <code>request.parameters.name.locations</code> array for the index positions of the values <code>path</code> and <code>query</code>, and returns, in a single array, the <code>request.parameters.name.values[index]</code> values for all identified index positions. When the URL contains both path and query parameter values, the array includes the path values first followed by the query values. The values of each parameter type are added in the order that they are received. Parameter values are URL decoded.</p> <p>For example, the following URL contains both path and query parameter values:</p> <pre>http://example.com/petstore/cats/adopt?breed=Sphynx&amp;breed=Siamese</pre> <p>The function call <code>\$urlParameter('breed')</code> returns the following array of values:</p> <pre>[cats, adopt, Sphynx, Siamese]</pre> <p>In this example, the URL includes an API path that is configured as <code>/petstore/{breed}/{breed}</code>, where <code>breed</code> is configured to be a path parameter of the API path. As a result, <code>cats</code> and <code>adopt</code> are included in the output.</p>
<code>\$xpath(path, xpathExpression)</code>	You can specify any writable variable in the API context. The <code>xpathExpression</code> must be a literal string.	Allows use of XPath expressions

## Simple condition statements

The following examples show conditions that use a single function.

This example uses the `$httpVerb()` extension to specify the HTTP method of the request.

```
$httpVerb()="GET"
```

This example uses the `$operationPath()` extension to specify the path of the operation.

```
$operationPath()="/base/path-2"
```

This example uses the `$operationID()` extension to specify the operation ID.

```
$operationID()="test-gatewayscript-GET"
```

This example uses the `$statusCode()` extension to specify the message status code.

```
$statusCode()=200
```

This example uses the `$header(name)` extension to specify the content type of the message header.

```
$header("Content-Type")="application/json"
```

## Combining conditions with logical operators

You can use `and` and `or` operators to combine multiple functions in a single condition.

This example specifies an HTTP GET request and an API operation path equal to `test-gatewayscript-GET`.

```
$httpVerb()="GET" and $operationPath()="test-gatewayscript-GET"
```

This example specifies either a `POST` or `PUT` request.

```
$httpVerb()="POST" or $httpVerb()="PUT"
```

This example specifies an API operation ID equal to `test-gatewayscript_POST` and a message status code equal to `200`, or an API operation ID equal to `test-gatewayscript-GET` and a message status code equal to `500`.

```
($operationID()="test-gatewayscript-POST" and $statusCode()=200) or ($operationID()="test-gatewayscript-GET" and $statusCode()=500)
```

This example specifies an API operation ID equal to `test-gatewayscript-POST` and a message status code equal to `200` with an API operation path equal to `/base/path-2`.

```
($operationID()="test-gatewayscript-POST" and $statusCode()=200) and $operationPath()="/base/path-2"
```

This example specifies a message header content type equal to `text/plain` and a message header length equal to `300`, or a message status code equal to `200`.

```
($header("Content-Type")="text/plain" and $header("Content-Length")=300) or $statusCode()= 200
```

## Example switch policy

```
- switch:
  version: 2.0.0
  title: switch
  case:
    - condition: ($statusCode() = 200)
      execute:
        - invoke:
            title: invoke
            timeout: 60
            verb: GET
            cache-response: protocol
            cache-ttl: 900
            target-url: 'https://example.com/1'
    - condition: ($statusCode() != 200 and ($httpVerb() = 'GET' or $httpVerb() = 'PUT'))
      execute:
        - invoke:
            title: invoke
            timeout: 60
            verb: GET
            cache-response: protocol
            cache-ttl: 900
            target-url: 'https://example.com/2'
    - otherwise:
      - set-variable:
          title: set-variable
          actions:
            - set: message.body
              value: Default result
              description: 'Set the default result for the otherwise case'
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Writing switch condition scripts - DataPower Gateway (v5 compatible)

You write condition scripts for the **switch** policy in the DataPower® Gateway (v5 compatible) by using GatewayScript.

To reference variables in your **switch** conditions, use the form `apim.getvariable('context.location.variable')`, where *context* is the context that you want to reference, *location* is the location of the variable within that context, and *variable* is the name of the variable.

to define a **switch** condition based on the operation called, use one of the following forms:

```
- condition: "((request.verb==='GET') && (api.operation.path===' /path-1'))"  
- condition: "((api.operation.id==='Operation_ID'))"
```

where the context variables `request.verb` and `api.operation.path` retrieve the HTTP verb and the path segment of the API and operation that you want your case to apply to, while `api.operation.id` retrieves the operation ID of your operation, if one has been specified.

For information about the context variables you can use and how to reference them in your script, see [API Connect context variables](#).

## Example switch policy

```
- switch:  
  version: 1.0.0  
  title: switch  
  case:  
    - condition: message.status.code===200  
      execute:  
        - invoke:  
          title: invoke  
          timeout: 60  
          verb: GET  
          cache-response: protocol  
          cache-ttl: 900  
          target-url: 'https://example.com/1'  
    - condition: ((request.verb==='GET') && (api.operation.path===' /path-2')) || ((request.verb==='GET') &&  
(api.operation.path===' /path-1'))  
      execute:  
        - invoke:  
          title: invoke  
          timeout: 60  
          verb: GET  
          cache-response: protocol  
          cache-ttl: 900  
          target-url: 'https://example.com/2'  
    - otherwise:  
      - set-variable:  
        title: set-variable  
        actions:  
          - set: message.body  
            value: Default result  
            description: Set the default result for the otherwise case
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## throw

Use the throw construct to throw an error when it is reached during an assembly flow, usually as a result of a condition being reached.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [throw](#).

## About

The throw policy has the following format:

```
- throw:  
  title: title  
  name: 'error_name'  
  message: error_message
```

When the throw policy is encountered, the specified error and error message is returned. If a catch has been configured that the error produced by the throw policy fulfills, the catch will be triggered.

## Properties



Table 2. operation-switch policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
name	Yes	The name of the error to be thrown.	string
message	Yes	The message to accompany the error.	string

## Example

```
- throw:
  version: 1.0.0
  title: throw
  name: '404'
  message: Not found
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## user-security

Use the user-security policy to extract a user's credentials, authenticate those credentials, and obtain authorization from the user.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [User Security](#).

## About

The user-security policy has the following format:

```
- user-security:
  version: version
  title: title
  description: description
  factor-id: factor ID
  extract-identity-method: method_used_to_extract_credentials
  .
  .
  .
  properties_specific_to_the_specified_identity_extraction_method
  .
  .
  .
  user-auth-method: authentication_method
  .
  .
  .
  properties_specific_to_the_specified_authentication_method
  .
  .
  .
  user-az-method: authorization_method
  .
  .
  .
  properties_specific_to_the_specified_authorization_method
  .
  .
  .
```

## Properties

Table 2. user-security policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	The title of the policy.	string
description	No	A description of the policy.	string
factor-id	No	The identity that identifies the results of factor-authentication in the API context.	string

Property	Required	Description	Data type
extract-identity-method	Yes	<p>Select the method that is used to extract the user credentials. The following options are available:</p> <p><b>basic</b> Use basic authentication; no additional configuration is required.</p> <p><b>context-var</b> The credentials are provided by API Connect context variables; specify the following properties:</p> <ul style="list-style-type: none"> <li>• user-context-var: the context variable that is used to obtain the user name.</li> <li>• pass-context-var: the context variable that is used to obtain the password.</li> </ul> <p><b>html-form</b> Use forms based identity-extraction. You can use the default form or a custom form. To use the default form, specify <b>ei-default-form: true</b>. To use a custom form, specify <b>ei-default-form: false</b> and supply the following properties:</p> <ul style="list-style-type: none"> <li>• ei-custom-form: the location of the form.</li> <li>• ei-custom-form-tls-client-profile: the TLS client profile that is used to secure the connection to the remote server.</li> </ul> <p>Use the ei-form-time-limit property to specify the time allowed to submit the form.</p> <p><b>redirect</b> Use a redirect for identity-extraction; specify the following properties:</p> <ul style="list-style-type: none"> <li>• redirect-url: the URL fragment to which to redirect the request to obtain user credentials.</li> <li>• redirect-time-limit: the time allowed for the transaction to complete.</li> </ul> <p><b>disabled</b> Identity-extraction is disabled; this aspect of processing is skipped.</p> <p>Specify <b>ei-stop-on-error: true</b> to halt assembly processing in the event of identity-extraction failure. This field is required if identity extraction is not disabled.</p>	string
user-auth-method	Yes	<p>Select the authentication method. The following options are available:</p> <p><b>auth-url</b> The credentials are authenticated by an external endpoint; specify the following properties:</p> <ul style="list-style-type: none"> <li>• auth-url: the URL of the authentication endpoint.</li> <li>• auth-tls-client-profile: the TLS client profile that is used to secure the connection to the authentication endpoint.</li> <li>• auth-response-headers-pattern: the pattern that is used to select which response headers to add to the API context.</li> <li>• auth-response-header-credential: the response header that contains the authenticated user credentials.</li> </ul> <p><b>ldap</b> The credentials are authenticated by an LDAP user registry; use the ldap-registry property to specify the required registry.</p> <p><b>disabled</b> Authentication is disabled; this aspect of processing is skipped.</p> <p>Specify <b>au-stop-on-error: true</b> to halt assembly processing in the event of user authentication failure. This field is required if authentication is not disabled.</p>	string

Property	Required	Description	Data type
user-az-method	Yes	<p>Select the authorization method. The following options are available:</p> <p>authenticated Implicitly accept any previously authenticated users; no additional configuration is required.</p> <p>html-form The user provides authorization through an HTML form. You can use the default form or a custom form. To use the default form, specify <b>az-default-form: true</b>. To use a custom form, specify <b>az-default-form: false</b>.</p> <p>Supply the following properties:</p> <ul style="list-style-type: none"> <li>az-table-dynamic-entries: the name of a context variable that specifies the scopes that are to be added automatically to the authorization consent form.</li> <li>az-form-time-limit: the time allowed to submit the form.</li> </ul> <p>For a custom form, supply the following properties:</p> <ul style="list-style-type: none"> <li>az-custom-form: the location of the form.</li> <li>az-custom-form-tls-client-profile: the TLS client profile that is used to secure the connection to the remote server.</li> </ul> <p>disabled Authorization is disabled; this aspect of processing is skipped.</p> <p>Specify <b>az-stop-on-error: true</b> to halt assembly processing in the event of user authorization failure. This field is required if authorization is not disabled.</p>	string

## Example

```
# basic authentication with LDAP registry
```

```
- user-security:
  version: 2.0.0
  title: user-security
  factor-id: default
  extract-identity-method: basic
  ei-stop-on-error: true
  user-auth-method: ldap
  ldap-registry: corporate-ldap
  au-stop-on-error: true
  user-az-method: authenticated
  az-stop-on-error: true
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## validate - DataPower API Gateway

Use the validate policy to validate the payload in an assembly flow against a schema.

### Gateway support

Note: This page describes the validate policy implementation in the DataPower® API Gateway. If you are using the DataPower Gateway (v5 compatible), see [validate - DataPower Gateway \(v5 compatible\)](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Validate - DataPower API Gateway](#).

### About

The validate policy has the following format:

```
- validate:
  version: version
  title: title
  description: description
  validate-against: validation_mechanism
```

```
properties_specific_to_the_specified_validation_mechanism
```

Apply this policy by adding an assembly extension with an execute field to your OpenAPI definition file.

Note: With the DataPower API Gateway, the input to the validate policy must be parsed data. One way to produce parsed data is to use a [parse](#) policy before a validate policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table describes the policy properties:

Table 2. validate policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
input	No	Identifies a variable in the API context. The content of the <b>body</b> field of the variable, which is represented by <b>variable_name.body</b> , is the input data to validate. By default, the variable name is <b>message</b> .	string
output	No	Specifies the name of a variable in the API context. <ul style="list-style-type: none"><li>If the validation passes, the body field of the output variable, which is represented by <b>variable_name.body</b>, stores the output of the assembly validate action.</li><li>If the schema to validate is a JSON schema, the validation also adds any default values that are missing from the payload.</li><li>If the validation fails, no output is stored.</li><li>If an output variable is not specified, the results of the assembly validate action are not saved.</li></ul>	string
validate-against	Yes	Specifies the schema to be used for validating the payload. Valid values: <ul style="list-style-type: none"><li><b>definition</b>: A previously defined schema will be used to validate the payload that is returned from other invoke actions or tasks in the assembly flow. In addition, supply a <b>definition</b> property to specify the required schema.</li><li><b>url</b>: the schema is identified by a URL location. In addition, supply the following properties:<ul style="list-style-type: none"><li><b>json-schema</b> field: the URL of the JSON schema to be used for validating a JSON payload.</li><li><b>xml-validation-mode</b>, specify one of the following values to define how an XML payload is validated:<ul style="list-style-type: none"><li><b>xsd</b>: use an XML schema; in addition, supply an <b>xml-schema</b> property that specifies the URL of the XML schema.</li><li><b>wSDL</b>: use a WSDL schema; in addition, supply an <b>xml-schema</b> property that specifies the URL of the WSDL schema to be used for validating a SOAP payload..</li><li><b>soap-body</b>: validate the body of a SOAP message against the XML schema only.</li></ul></li></ul></li><li><b>wSDL</b>: use the XML schema in the WSDL file associated with the API operation or the API path.</li><li><b>body-param</b>: validate the request input against the schema definition that is specified in the <b>schema</b> property for the request parameter for this operation.</li><li><b>response-param</b>: validate the response to be returned to the client application, against the schema definition that is specified in the <b>schema</b> property for the response parameter for this operation.</li></ul>	string

You can also apply a validate policy by using the API Manager assembly editor to add a built-in policy to the API. For more information, see [Validate - DataPower API Gateway](#) in the built-in policies section.

## Example 1

```
- validate:
  version: 2.0.0
  title: 'validate, response parameter schema'
  validate-against: response-param
```

## Example 2

```
- validate:
  version: 2.0.0
  title: 'validate, predefined schema'
  validate-against: definition
  definition: '#/definitions/RouteOutput'
```

## Example 3

```
- validate:
  version: 2.0.0
  title: 'validate, JSON and XML schema URLs'
  validate-against: url
  json-schema: 'https://my.json.schema'
```

```
xml-validation-mode: xsd
xml-schema: 'https://my.xml.schema'
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible)

## validate - DataPower Gateway (v5 compatible)

Use the validate policy to validate the payload in an assembly flow against a schema.

### Gateway support

Note: This page describes the validate policy implementation in the DataPower® Gateway (v5 compatible). If you are using the DataPower API Gateway, see [validate - DataPower API Gateway](#).

For information on the different types of gateway, see [API Connect gateway types](#).

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower Gateway (v5 compatible)	1.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Validate - DataPower Gateway \(v5 compatible\)](#).

Restriction:

- The schema that represents the XML can reference only one XML namespace.
- The schema cannot reference polymorphic XML elements.
- The validation works on the `message.body` variable and not any other output/context variable. If the invoke policy contains a configured response object variable, then `message.body` is not set, and validate is not able to act.
- If you use the `multipleOf` keyword in a schema definition for the API then, due to rounding behavior, the specified value must satisfy the following conditions, otherwise the validation fails when the API is called:
  - The value must not be less than `0.0000009999999999999999848869`.
  - If the value is greater than 1, the amount before the decimal point must not be greater than `999999999999999934463`.

### About

The validate policy has the following format:

```
- validate:
  version: version
  title: title
  description: description
  definition: swagger_schema_definition_to_be_used
```

Apply this policy by adding an assembly extension with an execute field to your OpenAPI definition file.

### Properties

The following table describes the policy properties:

Table 2. validate policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	No	A title for the policy.	string
description	No	A policy description.	string
definition	Yes	The schema to be used to validate the payload. Valid values: <ul style="list-style-type: none"><li>• <b>request:</b> Select this value to validate the request input against the schema definition that is specified in the Type field for the request parameter for this operation. For information about how to create a request parameter, see <a href="#">Configuring an operation</a>.</li><li>• <b>response:</b> Select this value to validate the response to be returned to the client application, against the schema definition that is specified in the Schema field for the response parameter for this operation. For information about how to create a response parameter, see <a href="#">Configuring an operation</a>.</li><li>• The name of a schema definition, in the following format: <pre>#/definitions/schema_name</pre> The schema must be defined in the <code>definitions:</code> section of your OpenAPI file.</li></ul>	string

You can also apply an validate policy by using the API Manager assembly editor to add a built-in policy to the API. For more information, see [Validate - DataPower Gateway \(v5 compatible\)](#) in the built-in policies section.

## Example 1

```
validate:
  version: 1.0.0
  title: validate the response
  definition: #/definitions/RouteOutput
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## validate-username-token

Use the Validate Username Token policy to validate a Web Services Security (WS-Security) UsernameToken in a SOAP payload before allowing access to the protected resource.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0 1.1.0 (IBM API Connect® Version 2018.4.1.4 or later)

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [Validate Username Token](#).

## About

The validate-username-token policy has the following format:

```
- validate-username-token:
  version: version
  title: title
  description: description
  auth-type: Authentication URL or LDAP Registry (policy version 1.0.0 only)
  auth-url: authentication_url_to_use (policy version 1.0.0 only)
  tls-profile: tls_profile_to_use (policy version 1.0.0 only)
  ldap-registry: name_of_the_ldap_user_registry (policy version 1.0.0 only)
  registry: name_of_the_ldap_or_authurl_user_registry (policy version 1.1.0 and later)
  ldap-search-attribute: name_of_the_ldap_user_password_attribute
```

## Properties

The following table describes the policy properties:

Table 2. Validate Username Token policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	The title of the policy.	string
description	No	A description of the policy.	string
auth-type (policy version 1.0.0 only)	Yes	The authentication type to use to validate the UsernameToken. Valid values: <ul style="list-style-type: none"><li>• <b>Authentication URL</b>: Specify this value to validate the user credentials against an authentication URL.</li><li>• <b>LDAP registry</b>: Specify this value to validate the user credentials against an LDAP user registry.</li></ul> The default value is: <b>Authentication URL</b> .	string
auth-url (policy version 1.0.0 only)	Yes	The authentication URL to use to validate the UsernameToken user credentials against. Note: This property is required only if Authentication type is set to <b>Authentication URL</b> .	string
tls-profile (policy version 1.0.0 only)	No	The TLS profile to use for the secure transmission of data to the authentication URL. Note: This property is available only if Authentication type is set to <b>Authentication URL</b> .	string
ldap-registry (policy version 1.0.0 only)	Yes	The name of the LDAP user registry to validate the UsernameToken user credentials against. Note: This property is required only if Authentication type is set to <b>LDAP registry</b> .	string
registry (policy version 1.1.0 and later)	Yes	The name of the LDAP or Authentication URL registry to use to validate the UsernameToken.	string
ldap-search-attribute	Yes	The name of the LDAP user password attribute. Note: This property is required only for an LDAP user registry.	string

## Examples

The following example shows an LDAP user registry authentication:

```
- validate-usnametoken:
  version: 1.0.0
  title: "validate-usnametoken"
  auth-type: "LDAP Registry"
  ldap-registry: "wstest"
  ldap-search-attribute: "userPassword"
```

The following example shows an Authentication URL definition:

```
- validate-usnametoken:
  version: 1.0.0
  title: "validate-usnametoken"
  auth-type: "Authentication URL"
  auth-url: "https://www.google.com"
  tls-profile: "default-ssl-profile"
```

For more information about how to use a validate-usnametoken security policy, see [Validate Username Token](#) in the built-in policies section.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## xml-to-json

Use the xml-to-json policy to convert the context payload of your API from the extensible markup language (XML) format to JavaScript Object Notation (JSON).

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [XML to JSON](#).

## About

The xml-to-json policy has the following structure:




```
- xml-to-json:
  version: version
  title: Title
  description: Description
```

Note: If you are using the DataPower API Gateway, the input to the xml-to-json policy must be parsed data. One way to produce parsed data is to use a [parse](#) policy before an xml-to-json policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table describes the policy properties:

Table 2. Policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	The title of the policy.	string
description	No	A description of the policy.	string
 input	No	The input message to convert. Specify the name of a variable in the API context. <b>variableName.body</b> , the message payload, represents the JSON input to convert. The default value of the variable is <b>message</b> and <b>message.body</b> is the default input.	string
 output	No	The output message to store the conversion result. Specify the name of a variable in the API context. <b>variableName.body</b> represents the result of conversion from JSON format to XML format. When the specified input message is the default message, the default output is <b>message.body</b> . Otherwise, when the input message is the variable <b>my-message-variable</b> , for example, the default output is <b>my-message-variable.body</b> . The variable cannot be any read-only in the API context.	string
 conversionType	No	The conversion type that determines the target format of the output. The following options are available: <ul style="list-style-type: none"> <li>badgerFish: BadgerFish convention is used to determine the target conversion format of the output.</li> <li>apicv5: apicv5 convention is used to determine the target conversion format of the output.</li> </ul>	string

## Example

The following is an example of a xml-to-json policy:

```
- xml-to-json:
  version: 1.0.0
```

**title:** XML to JSON transform  
**description:** Transforms XML message body to JSON format

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## xslt

Use the xslt policy to apply an XSLT transform to the payload of the API definition.

## Gateway support

Table 1. Table showing which gateways support this policy, and the corresponding policy version

Gateway	Policy version
DataPower® Gateway (v5 compatible)	1.0.0
DataPower API Gateway, policy available from V2018.4.1.0	2.0.0

This topic describes how to configure the policy in your OpenAPI source; for details on how to configure the policy in the assembly user interface, see [XSLT](#).

## About

The xslt policy has the following structure:

```
- xslt:  
  version: version  
  title: Title  
  description: Description  
  input: Input True False  
  source: Transform
```

**Note:** If you are using the DataPower API Gateway, the input to the xslt policy must be parsed data. One way to produce parsed data is to use a [parse](#) policy before an xslt policy in your assembly flow, which provides explicit control of the parse action.

## Properties

The following table describes the policy properties:

Table 2. xslt policy properties

Property	Required	Description	Data type
version	Yes	The policy version number	string
title	Yes	The title of the policy.	string
description	No	A description of the policy.	string
input	No	Indicates whether this XSLT input document uses the context current payload, or if there is no input. The default value is <b>false</b> .	boolean
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> serialize-output	No	If set to <b>true</b> , the output tree that is generated by the XSLT policy is serialized. The content of <b>message.body</b> is updated with the serialized binary data rather than the XML tree. The default value is <b>false</b> .	boolean
source	Yes	The XSLT transform source to execute.	string

You can also apply an xslt policy by using the API Manager assembly editor to add a built-in policy to the API. For more information, see [XSLT](#) in the built-in policies section.

For examples of the OpenAPI definitions of xslt policies, see [XSLT policy examples](#) in the built-in policies section.

For more examples of how to use XSLT to access and modify properties and context, see [Implementation code examples](#) and, if you are using the DataPower API Gateway, [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

## Related concepts

- [Variable references in API Connect](#)

## Related reference

- [XSLT](#)
- [API Connect context variables](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## catch

Use the `catch` extension to catch errors that occur during an API call.

The `catch` extension takes the following form:

```
catch:
  - errors:
    - Error_1
    - Error_2
  - execute:
    assembly_1
  - errors:
    - Error_3
  - execute:
    assembly_2
  - default:
    assembly_3
```

The following table shows the properties of the `catch` extension:

Table 1. The properties of the `catch` extension

Property	Required	Description	Type
<code>errors</code>	Yes	The errors for which the catch will activate. For a list of errors, see <a href="#">Error cases supported by assembly catches</a> .	Array (String)
<code>execute</code>	Yes	The section of the assembly that will execute when the catch is activated.	Array (Object)
<code>default</code>	No	The section of the assembly that will execute when an error does not trigger other catches. It behaves in the same manner and has the same structure as <code>execute</code>	Array (Object)

The following is an example of a `catch` extension:

```
catch:
  - errors:
    - ConnectionError
    - JavaScriptError
  execute:
    - activity-log:
      title: activity-log
      content: activity
      error-content: payload
  - default:
    - activity-log:
      title: activity-log
      content: activity
      error-content: payload
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## properties

Use the `properties` extension to define properties for referencing in an API.

The `properties` extension has the following structure:

```
properties:
  property_1:
    value: default_value_1
    description: description_1
    encoded: encoded_boolean_1
  property_2:
    value: default_value_2
    description: description_2
    encoded: encoded_boolean_2
```

The following table lists the fields found in the `properties` extension:

Table 1. The properties extension

Property	Required	Description	Data type
<code>property</code>	Yes	The name of the property. It is used when referencing the property. Note that this is the field name, not the contents of the field.	Object
<code>value</code>	No	The default value that the property takes. It can be empty.	String
<code>description</code>	No	The description of the property. It can be empty.	String
<code>encoded</code>	No	Specifies whether to encode the value of this property.	Boolean

The following example shows a sample `properties` field:

```
properties:
  ID:
    value: 1234
    description: An ID to be used for validating.
    encoded: false
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## catalogs

Use the `catalogs` extension to assign catalog-specific values to properties defined in the `properties` section.

The `catalogs` section has the following structure:

```
catalogs:
  catalog_name_1:
    properties:
      property_name_1: value_1
      property_name_2: value_2
  catalog_name_2:
    properties:
      property_name_1: value_3
```

The following table lists the fields found in the `catalogs` extension:

Table 1. Catalogs properties

Property	Required	Description	Data type
<code>catalog_name</code>	Yes	The name of the catalog. It must match a catalog in API Manager. Note that this is the field name, not the contents of the field.	Object
<code>properties</code>	Yes	This field contains any properties that are to be set.	Object
<code>property_name</code>	Yes	The field name is the property to be set and must match a property defined in the <code>properties</code> extension. It contains the value of the property for the catalog to which the field belongs. If, in the <code>properties</code> extension, it has been specified as encoded, this will be encoded when saved or staged by IBM® API Connect.	String

The following is an example of a `catalogs` extension:

```
catalogs:
  Sandbox:
    properties:
      ID: 5678
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## activity-log

If you're using the DataPower® API Gateway, you can use the `activity-log` extension to configure your logging preferences for the API activity that is stored in Analytics. The preferences that you specify will override the default settings for collecting and storing details of the API activity.

An API event record exists for each API execution event in the Gateway server. By default, the content type that is collected and stored in API event records is `activity` for when API execution completes successfully, and `payload` for when API execution completes with an error code. When you compose your API definition, you can change the type of content to log in these API event records. For more information about API event records, see [API event record fields](#).

Note that if you're using the DataPower Gateway (v5 compatible), you can configure your logging preferences by using an `activity-log` policy in your API assembly. For more information, see [activity-log](#).

The `activity-log` extension takes the following form:

```
activity-log:
  success-content: activity_to_log_if_call_successful
  error-content: activity_to_log_if_call_unsuccessful
  enabled: is_activity_logging_enabled
```

The following table shows the properties of the `activity-log` extension:

Table 1. The properties of the activity-log extension

Property	Required	Description	Type
<code>success-content</code>	No	Defines the type of content to be logged when the operation is successful. Valid values: <ul style="list-style-type: none"> <li><code>none</code>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li><code>activity</code>: Logs invocation only (only the resource URI is recorded).</li> <li><code>header</code>: Logs activity and header.</li> <li><code>payload</code>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <code>activity</code> .	String

Property	Required	Description	Type
<code>error-content</code>	No	Indicates what content to log when an error occurs. Valid values: <ul style="list-style-type: none"> <li><b>none</b>: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal.</li> <li><b>activity</b>: Logs invocation only (only the resource URI is recorded).</li> <li><b>header</b>: Logs activity and header.</li> <li><b>payload</b>: Logs activity, header, and payload (the original request, if any, and the final response).</li> </ul> The default value is <b>payload</b> .	String
<code>enabled</code>	No	Indicates whether activity logging is enabled or disabled.	Boolean

## Example 1

Override the default activity logging settings:

```
activity-log:
  success-content: none
  error-content: header
  enabled: true
```

## Example 2

Turn off activity logging:

```
activity-log:
  enabled: false
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Specifying a gateway type for an API definition

An API definition is specific to one or other of the gateway types, DataPower® API Gateway or DataPower Gateway (v5 compatible). API definitions must specify which type of gateway the API uses.

### Before you begin

IBM® API Connect supports two gateway types: DataPower Gateway (v5 compatible) and DataPower API Gateway.

DataPower Gateway (v5 compatible) has been available with IBM API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

For more information on how to choose which gateway type to use, see [API Connect gateway types](#).

### About this task

You must specify which type of gateway each API uses. APIs can use only one type of gateway.

When you modify your API definitions to use a specific gateway type, you must ensure that each policy and policy version in the API are supported by the gateway type. DataPower Gateway (v5 compatible) and DataPower API Gateway each support policies that the other gateway type does not. In some cases, the same policy is supported by both gateway types, but with a different version number.

For example, DataPower Gateway (v5 compatible) supports version **1.0.0** of the **invoke** policy, but DataPower API Gateway requires version **2.0.0**.

For information on policies, see [execute](#).

For information on policy versions, see the documentation for each individual policy. For example, to review the **invoke** policy, see [invoke](#).

### Procedure

- Edit the API definition YAML file and add the gateway type. For example:
  - DataPower Gateway (v5 compatible):

```
gateway: datapower-gateway
```
  - DataPower API Gateway

```
gateway: datapower-api-gateway
```
- Ensure that the policies, including policy versions, in your APIs are supported by the gateway type. Review the list of policies on the [execute](#) page. If necessary, review the individual policy pages to determine the supported version number.
- Ensure that any Product that uses this API is configured to use the same gateway type. See [Specifying a gateway type for your Product](#).

## Related information

- [DataPower API Gateway porting notes](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Referring to an extension in an API definition

To refer to an OpenAPI extension in your API definition YAML (or JSON) file, add an `extensions` key under `x-ibm-configuration`.

## Before you begin

Add the OpenAPI extension to any Catalogs that the API is to be published to, by importing its definition file; for details, see [Working with OpenAPI extensions](#). If the target Catalog does not contain the OpenAPI extension then attempts to publish the API to it will fail.

## About this task

The following example shows an extension definition file that defines a bank branch, in YAML format:

```
extension: '1.0.0'

info:
  title: Banking services
  name: banking
  version: 1.0.0
  description: Banking extensions
  contact:
    name: IBM API Connect
    url: https://apiconnect.ibm.com/
    email: myname@ibm.com

portal-visible: true

properties:
  title: "Branch"
  type: "object"
  properties:
    Branch type:
      type: "string"
      enum:
        - "ATM"
        - "Walk in"
    location:
      type: "object"
      title: "Location"
      properties:
        city:
          type: "string"
          default: "San Francisco"
        state:
          type: "string"
          default: "CA"
        citystate:
          type: "string"
          description: "This is generated automatically from the previous two fields"
          template: "{{city}}, {{state}}"
          watch:
            city: "location.city"
            state: "location.state"
      required:
        - Branch type
```

## Procedure

1. Edit the API definition YAML file and add the reference. For example:

```
x-ibm-configuration:
  .
  .
  .
  extensions:
    banking: 1.0.0
```

2. In your API definition file, add a property whose name must begin with `x-` and references the name of the extension you created. Specify the values of the properties that were defined in the OpenAPI extension YAML file.  
The following example uses the banking extension:

```
.
.
```

```
x-banking:
  Branch type: ATM
  location:
    city: San Francisco
    state: CA
    citystate: 'San Francisco, CA'
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using \$ref to reuse code fragments in your OpenAPI files

If you deploy an API to an IBM® API Connect Management server by using the developer toolkit command line, you can use the `$ref` field in your OpenAPI YAML and JSON API definition files to reference a fragment of OpenAPI code that is defined in a separate file. When API Connect processes the source API definition file, the `$ref` field is replaced with the contents of the target file.

Use the following syntax in your source YAML file:

```
$ref: path_to_file_containing_code_fragment
```

Use the following syntax in your source JSON file:

```
{
  $ref: path_to_file_containing_code_fragment
}
```

For example:

```
$ref: ./code_fragments/my_fragment.yaml

{
  "$ref": "./code_fragments/my_fragment.json"
}
```

The replacement of the `$ref` field with the target code fragment occurs when you perform any of the following actions on the API defined by the source API definition file:

- Stage or publish an API to an API Connect Management server by using the `apic publish` command. For more information, see [Publishing APIs and applications](#).
- Validate the API definition YAML file by using the `apic validate` command. For more information, see [Validating the YAML or JSON definition of an API or Product](#).

Important: You cannot insert a `$ref` field at the root level of your OpenAPI file.

---

## Example

A source YAML file contains the following OpenAPI code:

```
swagger: '2.0'
info:
  version: 1.0.0
  title: Branches
  x-ibm-name: Branches
  description: Provides operations relating to BankA branch information.
basePath: /branches
paths:
  $ref: ./code_fragments/paths.yaml
  .
  .
  .
```

The file `paths.yaml` contains the following OpenAPI code fragment:

```
/details:
  get:
    responses:
      '200':
        description: 200 OK defined in $ref file
        schema:
          $ref: '#/definitions/branch'
    summary: Branch details
    description: Retrieve details of the current branches of BankA.
```

When API Connect processes the source YAML file, the `$ref` field is replaced with the target code fragment, yielding the following OpenAPI code:

```
swagger: '2.0'
info:
  version: 1.0.0
  title: Branches
  x-ibm-name: Branches
  description: Provides operations relating to BankA branch information.
basePath: /branches
paths:
  /details:
    get:
      responses:
        '200':
```

```
description: 200 OK
schema:
  $ref: '#/definitions/branch'
summary: Branch details
description: Retrieve details of the current branches of BankA.
.
.
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Product definition file

You define a Product by creating a Product definition file.

### About this task

---

You can create a Product definition file in either of the following ways:

- Create the file in an editor of your choice.
- Create a Product definition file by using the `apic create:product` command and then modify it. You can base your Product definition file on the default Product template or on your own custom template.

This topic describes the syntax that is needed for both options and also the structure needed when the file is created in the editor. For details on what you can include in the product definition file, see [Product definition schema](#).

You can create a Product in the CLI by running `apic create:product` and supplying additional arguments on the command line. For example, you can enter the following text in a single command:

```
apic create:product --name product_name --title product_title --filename product_file_name.yaml
--apis "filename_of_api1.yaml filename_of_api2.yaml"
```

where:

- `product_name` is the name for your new Product.
- `product_title` is the title of your new Product.
- `product_file_name.yaml` is the name of the yaml file that is created for your new Product.
- `filename_of_api1.yaml` is the file name of one of the APIs that is used within the new Product.
- `filename_of_api2.yaml` is the file name of one of the APIs that is used within the new Product.

Another option is to create a Product interactively in the command line by running `apic create:product` and following the prompts.

You can see further details and available options for the `apic create:product` command by entering the command:

```
apic create:product --help
```

You can also create a Product from a custom Handlebars template by using the following command:

```
apic create:product --template template_filename --title product_title
```

where `template_filename` is the name of the Handlebars template to use, and `product_title` is the title of your Product. A Product template file must have a `.hbs` file name extension. You can create a template from scratch, or start with the example (default) Product template provided in [API and Product definition template examples](#).

A Product definition file contains the following sections:

- The specification version
- An information section
- A visibility section
- An APIs section
- A Plans section

In this topic, YAML is used, but the instructions can be adapted to use JSON. Both `.yaml` and `.yml` file extensions are supported, but the use of the `.yaml` file extension is recommended by [yaml.org](#).

**Note:** All keys and enumeration values that are specified in this topic are case-sensitive.

## Procedure

---

Structure your Product definition file by completing the following steps:

1. Set the specification version by adding the following line to the beginning of the file: `product: 1.0.0`.  
This line specifies the template version, and is always `product: 1.0.0`.  
Note: The specification version is distinct from your Product version. The specification version refers to this YAML file, while the Product version is by your discretion.
2. Include an information section with details about the Product, as described in [Completing the information section of your Product description](#).
3. Include a visibility section that specifies who can view and subscribe to the Product, as described in [Specifying the visibility of your Product](#).
4. Include an APIs section that references the APIs to be included in the Product, as described in [Referencing the APIs for your Product](#).
5. Include a Plans section that described the Plans you want include in your Product, as described in [Describing Plans in your Product](#).

## Results

---

You have completed a YAML representation of your Product. A complete example with full indentation can be found in [An example YAML representation of a Product](#).

You can create multilingual API and Product documentation by using an `x-ibm-languages` extension directly in the OpenAPI definition. For more information, see [Using x-ibm-languages to create multilingual API and Product documentation](#).

- [Product definition schema](#)  
The product definition schema defines the properties (and their data types) that you can include in a product definition file.
- [Converting a Product YAML file to use DataPower API Gateway](#)  
IBM API Connect now supports DataPower® API Gateway. You can optionally specify this gateway type in your Product YAML file.

## Related information

---

- [Changing the availability of a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Product definition schema

The product definition schema defines the properties (and their data types) that you can include in a product definition file.

```
{
  "id": "https://api.ibm.com/product/schema.json#",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "title": "A schema to represent the IBM specific API configuration",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "product",
    "info"
  ],
  "patternProperties": {
    "^x-": {}
  },
  "properties": {
    "product": {
      "type": "string",
      "enum": [
        "1.0.0"
      ]
    },
    "info": {
      "$ref": "#/definitions/InfoSection"
    },
    "visibility": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "view",
        "subscribe"
      ]
    },
    "patternProperties": {
      "^x-": {}
    },
    "properties": {
      "view": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "type"
        ]
      },
      "properties": {
        "type": {
          "enum": [
            "public",
            "authenticated",
            "custom"
          ]
        }
      },
      "orgs": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "tags": {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    },
    "enabled": {
```

```

        "type": "boolean"
      }
    },
    "subscribe": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "type"
      ],
      "properties": {
        "type": {
          "enum": [
            "authenticated",
            "custom"
          ]
        },
        "orgs": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "tags": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "enabled": {
          "type": "boolean"
        }
      }
    }
  },
  "properties": {
    "type": "object",
    "propertyNames": {
      "pattern": "^[^ ]|([^ ].*[^ ])$"
    },
    "additionalProperties": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "value": {
          "type": "string"
        },
        "description": {
          "type": "string"
        },
        "encoded": {
          "type": "boolean"
        }
      }
    }
  },
  "catalogs": {
    "type": "object",
    "propertyNames": {
      "pattern": "^[^ ]|([^ ].*[^ ])$"
    },
    "additionalProperties": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "properties": {
          "type": "object",
          "additionalProperties": {
            "type": "string"
          }
        }
      }
    }
  },
  "gateways": {
    "type": "array",
    "minItems": 1,
    "items": {
      "enum": [
        "micro-gateway",
        "datapower-gateway",
        "datapower-api-gateway",
        "event-gateway"
      ]
    }
  },
  "apis": {
    "type": "object",
    "propertyNames": {
      "pattern": "^[^ ]|([^ ].*[^ ])$"
    },
    "additionalProperties": {
      "type": "object",
      "additionalProperties": false,
      "properties": {

```



```

    "$ref": {
      "type": "string"
    },
    "name": {
      "type": "string"
    }
  }
},
"oauthProviders": {
  "type": "array",
  "propertyNames": {
    "pattern": "^[^ ]|([^ ].*[^ ])$"
  },
  "items": {
    "type": "string"
  }
},
"billings": {
  "type": "object",
  "propertyNames": {
    "pattern": "^[^ ]|([^ ].*[^ ])$"
  },
  "additionalProperties": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "name": {
        "type": "string"
      }
    }
  }
},
"plans": {
  "type": "object",
  "propertyNames": {
    "pattern": "^[^ ]|([^ ].*[^ ])$"
  },
  "additionalProperties": {
    "type": "object",
    "additionalProperties": false,
    "patternProperties": {
      "^x-": {}
    },
    "required": [
      "title"
    ],
    "properties": {
      "title": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "approval": {
        "type": "boolean"
      },
      "rate-limit": {
        "$ref": "#/definitions/RateLimitSection"
      },
      "rate-limits": {
        "$ref": "#/definitions/RateLimitsSection"
      },
      "burst-limits": {
        "$ref": "#/definitions/BurstLimitsSection"
      },
      "assembly-rate-limits": {
        "$ref": "#/definitions/AssemblyRateLimitsSection"
      },
      "assembly-burst-limits": {
        "$ref": "#/definitions/AssemblyBurstLimitsSection"
      },
      "assembly-count-limits": {
        "$ref": "#/definitions/AssemblyCountLimitsSection"
      },
      "billing": {
        "$ref": "#/definitions/BillingSection"
      },
      "graphql": {
        "$ref": "#/definitions/GraphQLSection"
      },
      "apis": {
        "type": "object",
        "propertyNames": {
          "pattern": "^[^ ]|([^ ].*[^ ])$"
        },
        "additionalProperties": {
          "type": "object",
          "additionalProperties": false,
          "properties": {
            "operations": {
              "type": "array",
              "items": {
                "type": "object",
                "additionalProperties": false,
                "properties": {

```

```

        "operationId": {
          "type": "string"
        },
        "path": {
          "type": "string"
        },
        "operation": {
          "type": "string"
        },
        "rate-limit": {
          "$ref": "#/definitions/RateLimitSection"
        },
        "rate-limits": {
          "$ref": "#/definitions/RateLimitsSection"
        }
      }
    },
    "graphql-schema-options": {
      "$ref": "#/definitions/GraphQLSchemaOptions"
    }
  }
}
},
"definitions": {
  "GraphQLSchemaOptions": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "visibility-list"
    ],
    "properties": {
      "visibility-list": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "type",
          "values"
        ],
        "properties": {
          "type": {
            "type": "string",
            "enum": [
              "remove"
            ]
          },
          "values": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/NonEmptyString"
            }
          }
        }
      }
    }
  }
},
"InfoSection": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "title",
    "version"
  ],
  "patternProperties": {
    "^x-": {}
  },
  "properties": {
    "name": {
      "type": "string",
      "pattern": "^[a-zA-Z0-9._]+(-)*([a-zA-Z0-9])$"
    },
    "version": {
      "type": "string"
    },
    "title": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "termsOfService": {
      "type": "string"
    },
    "contact": {
      "type": "object",
      "additionalProperties": false,
      "patternProperties": {
        "^x-": {}
      },
      "properties": {
        "name": {
          "type": "string"
        }
      }
    }
  }
}

```

```

    },
    "url": {
      "type": "string",
      "format": "uri"
    },
    "email": {
      "type": "string",
      "format": "email"
    }
  }
},
"license": {
  "type": "object",
  "additionalProperties": false,
  "patternProperties": {
    "^x-": {}
  },
  "required": [
    "name"
  ],
  "properties": {
    "name": {
      "type": "string"
    },
    "url": {
      "type": "string",
      "format": "uri"
    }
  }
},
"summary": {
  "type": "string",
  "description": "Short description of the API."
},
"categories": {
  "type": "array",
  "items": {
    "type": "string"
  }
}
},
"RateLimitSection": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "value"
  ],
  "patternProperties": {
    "^x-": {}
  },
  "properties": {
    "value": {
      "$ref": "#/definitions/RateLimitValue"
    },
    "hard-limit": {
      "$ref": "#/definitions/RateLimitHardLimit"
    }
  }
},
"AssemblyRateLimitSection": {
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "value",
      "hard-limit",
      "cache-only",
      "is-client",
      "use-api-name",
      "use-app-id",
      "use-client-id"
    ],
    "patternProperties": {
      "^x-": {}
    },
    "properties": {
      "value": {
        "$ref": "#/definitions/RateLimitValue"
      },
      "hard-limit": {
        "$ref": "#/definitions/RateLimitHardLimit"
      },
      "cache-only": {
        "type": "boolean"
      },
      "is-client": {
        "type": "boolean"
      },
      "use-api-name": {
        "type": "boolean"
      },
      "use-app-id": {
        "type": "boolean"
      }
    }
  }
}

```

```

    },
    "use-client-id": {
      "type": "boolean"
    },
    "dynamic-value": {
      "type": "string"
    },
    "weight": {
      "$ref": "#/definitions/NonEmptyString"
    }
  }
},
"AssemblyBurstLimitSection": {
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "value",
      "cache-only",
      "is-client",
      "use-api-name",
      "use-app-id",
      "use-client-id"
    ],
    "patternProperties": {
      "^x-": {}
    },
    "properties": {
      "value": {
        "$ref": "#/definitions/RateLimitValue"
      },
      "cache-only": {
        "type": "boolean"
      },
      "is-client": {
        "type": "boolean"
      },
      "use-api-name": {
        "type": "boolean"
      },
      "use-app-id": {
        "type": "boolean"
      },
      "use-client-id": {
        "type": "boolean"
      },
      "dynamic-value": {
        "type": "string"
      },
      "weight": {
        "$ref": "#/definitions/NonEmptyString"
      }
    }
  }
},
"NonEmptyString": {
  "type": "string",
  "pattern": ".+"
},
"AssemblyCountLimitSection": {
  "type": "array",
  "minItems": 1,
  "maxItems": 1,
  "items": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "value",
      "hard-limit",
      "cache-only",
      "is-client",
      "use-api-name",
      "use-app-id",
      "use-client-id"
    ],
    "patternProperties": {
      "^x-": {}
    },
    "properties": {
      "value": {
        "oneOf": [
          {
            "type": "integer"
          },
          {
            "type": "string",
            "enum": [
              "unlimited"
            ]
          }
        ]
      },
      "hard-limit": {

```

```

    "$ref": "#/definitions/RateLimitHardLimit"
  },
  "cache-only": {
    "type": "boolean"
  },
  "is-client": {
    "type": "boolean"
  },
  "use-api-name": {
    "type": "boolean"
  },
  "use-app-id": {
    "type": "boolean"
  },
  "use-client-id": {
    "type": "boolean"
  },
  "dynamic-value": {
    "type": "string"
  },
  "weight": {
    "$ref": "#/definitions/NonEmptyString"
  },
  "auto-decrement": {
    "type": "boolean"
  }
}
},
"RateLimitValue": {
  "type": "string",
  "pattern": "^[0-9]|[1-9][0-9]{1,8}|[1-3][0-9]{9}|4[01][0-9]{8}|42[0-8][0-9]{7}|429[0-3][0-9]{6}|4294[0-8][0-9]{5}|42949[0-5][0-9]{4}|429496[0-6][0-9]{3}|4294967[01][0-9]{2}|42949672[0-8][0-9]|429496729[0-5]\\.\\.\\/([0-9]|[1-9][0-9]{1,3}|[1-5][0-9]{4}|6[0-4][0-9]{3}|65[0-4][0-9]{2}|655[0-2][0-9]|6553[0-5])?(second|minute|hour|day|week)|unlimited)$"
},
"RateLimitHardLimit": {
  "type": "boolean",
  "default": false
},
"RateLimitsSection": {
  "type": "object",
  "patternProperties": {
    "^x-": {}
  },
  "additionalProperties": {
    "type": "object",
    "oneOf": [
      {
        "$ref": "#/definitions/RateLimitSection"
      }
    ]
  }
},
"AssemblyRateLimitsSection": {
  "type": "object",
  "patternProperties": {
    "^x-": {}
  },
  "additionalProperties": {
    "type": "object",
    "$ref": "#/definitions/AssemblyRateLimitSection"
  }
},
"AssemblyBurstLimitsSection": {
  "type": "object",
  "patternProperties": {
    "^x-": {}
  },
  "additionalProperties": {
    "type": "object",
    "$ref": "#/definitions/AssemblyBurstLimitSection"
  }
},
"AssemblyCountLimitsSection": {
  "type": "object",
  "patternProperties": {
    "^x-": {}
  },
  "additionalProperties": {
    "type": "object",
    "$ref": "#/definitions/AssemblyCountLimitSection"
  }
},
"BurstLimitsSection": {
  "type": "object",
  "additionalProperties": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "value"
    ]
  },
  "patternProperties": {
    "^x-": {}
  },
  "properties": {
    "value": {

```



---

## Completing the information section of your Product description

Provide users with information about your Product.

### About this task

---

The information section of a YAML representation of a Product contains the Product's name and version. It can also include contact and license details.

Note: All keys and enumeration values that are specified in this topic are case-sensitive.

An example information section for a YAML representation of a Product can be found at the end of this topic. An example of a complete YAML representation of a Product can be found in [An example YAML representation of a Product](#).

### Procedure

---

To complete the information section of your Product description, complete the following steps.

1. Begin the section and provide a name, title, description, and version number; use the following syntax:

```
info:
  version: Product_Version
  title: Product_Title
  name: Short_Name
  summary: Product_Description
```

where:

- *version* is the version number of the policy.  
Tip: The `version.release.modification` version numbering scheme is recommended, for example `1.0.0`.
- *Product\_Title* is the title of the Product. Any string can be used, but the title should be kept short so that it can be displayed in the API Manager user interface.
- *Short\_Name* is the short name for the policy. It must be a single word and contain only alphanumeric characters, and the - (dash) and \_ (underscore) characters. The name is case-sensitive, and should be 20 characters or fewer so that it can be displayed in the API Manager user interface of IBM API Connect.
- *Product\_Description* is a short description of the policy. Any string can be used.

2. Optional: Under info, supply contact information; use the following syntax:

```
contact:
  name: Contact_Name
  url: 'Contact_URL'
  email: Contact_email
```

where:

- *Contact\_Name* is the name of the contact for this Product.
- *Contact\_URL* is the URL for contact your organization.
- *Contact\_email* is an email for contacting your organization.

3. Optional: Under info, add license information for your Product; use the following syntax:

```
license:
  name: License_Name
  url: 'License_URL'
```

where:

- *License\_Name* is the name of the license that applied to your Product.
- *License\_URL* is the URL at which information about the license can be accessed.

4. Optional: Under info, add your terms of service; use the following syntax:

```
termsOfService: Terms
```

where *Terms* is a string that details the terms of service for using your Product.

### Results

---

You have completed the information section of your Product's YAML representation. It should have the following form:

```
info:
  version: Product_Version
  title: Product_Title
  name: Short_Name
  description: Product_Description
  contact:
    name: Contact_Name
    url: 'Contact_URL'
    email: Contact_email
  license:
    name: License_Name
    url: 'License_URL'
  termsOfService: Service_Terms
```

where all variables are as described in previously in this topic. The indentation must be as in the example.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Specifying a gateway type for your Product

Product definitions must specify which type of gateway the Product uses.

### Before you begin

---

IBM® API Connect supports two gateway types: DataPower® Gateway (v5 compatible) and DataPower API Gateway.

DataPower Gateway (v5 compatible) has been available with IBM API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

For more information on how to choose which gateway type to use, see [API Connect gateway types](#).

### About this task

---

You can specify the gateway type for a Product.

Products can use only one type of gateway, and the APIs in the Product must use the same type of gateway. See [Specifying a gateway type for an API definition](#).

### Procedure

---

1. Determine the gateway type that is configured for the APIs in your Product. The Product gateway type must match the APIs gateway type.  
To review the APIs for your Product, see [Referencing the APIs for your Product](#).

2. Edit the Product definition file (YAML source) to add the gateway type you want to use:

- DataPower Gateway (v5 compatible):

```
gateways:  
  datapower-gateway
```

- DataPower API Gateway

```
gateways:  
  datapower-api-gateway
```

3. Ensure that the policies, including policy versions, in the APIs in your Product are supported by the gateway type. Review the list of policies on the [execute](#) page. If necessary, review the individual policy pages to determine the supported version number.

### Related information

---

- [DataPower API Gateway porting notes](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Converting a Product YAML file to use DataPower API Gateway

IBM® API Connect now supports DataPower® API Gateway. You can optionally specify this gateway type in your Product YAML file.

### Before you begin

---

IBM API Connect provides two gateway types, DataPower Gateway (v5 compatible) and DataPower API Gateway.

DataPower Gateway (v5 compatible) has been available with IBM API Connect for a number of years. The DataPower API Gateway is a new gateway that has been designed with APIs in mind, and with the same security focus as DataPower Gateway (v5 compatible).

For more information on how to choose which gateway type to use, see [API Connect gateway types](#).

### About this task

---

Product definitions can specify only one type of gateway. The type of gateway must match the gateway type that is used by the APIs that are included in the product. If you want to take advantage of the capabilities of DataPower API Gateway, you can easily modify your existing Product definitions to specify it as the gateway type.

### Procedure

---



1. Verify that the APIs in the product specify DataPower API Gateway as the gateway type, and that the APIs have been converted for use with the DataPower API Gateway.
2. Open the Product YAML file for editing.  
For example:

```
product: '1.0.0'
info:
  name: testprod
  title: testprod
  version: 1.0.0
  .
  .
```

3. Add a `gateways` setting, and specify `datapower-api-gateway`.  
For example:

```
product: '1.0.0'
info:
  name: testprod
  title: testprod
  version: 1.0.0
gateways:
  - datapower-api-gateway
  .
  .
```

4. Save the modified YAML file.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Specifying the visibility of your Product

Detail who can view and subscribe to your Product.

### About this task

The visibility section of your Product's YAML representation determines who can view and subscribe to your Product in IBM® API Connect.

Note: All keys and enumeration values that are specified in this topic are case-sensitive.

An example visibility section of a YAML representation of a Product can be found at the end of this topic. An example of a complete YAML representation of a Product can be found in [An example YAML representation of a Product](#).

### Procedure

To complete the visibility section of your Product description, complete the following steps:

1. Title the section by adding `visibility`:
2. Under visibility, specify who can view the Product; use the following syntax:

```
view:
  enabled: View_Toggle
  type: View_Audience
  tags:
    - View_Tag_1
    - View_Tag_2
  orgs: View_Organizations
```

where

- `View_Toggle` determines whether the Product is visible to anybody or not. It must be `true` or `false`. If `false`, the Product will not be visible in the Developer Portal.
- `View_Audience` must be `public`, in which case the Product is visible to anybody who uses the Developer Portal, `authenticated`, in which case the Product is visible to anybody registered through the Developer Portal, or `custom`, in which case the Product is visible to a specified group of users.
- The `View_Tag` variables are strings that contains any tags that you want to attach to your Product's view status. Each tag must be on a new line and preceded with a dash. If you do not want to use any tags, do not include `tags:`.
- `View_Organizations` specifies the organizations that can view the Product if `View_Audience` is set to `custom`. If `View_Audience` is not set to `custom`, do not include `orgs:`. If `View_Audience` is `custom`, specify organizations in the following manner:

```
orgs:
  - Organization_1
  - Organization_2
```

where the `Organization` variables are the names of the organizations that are to be allowed to view the Product.

3. Under visibility, specify who can subscribe to your Product's Plans; use the following syntax:

```
subscribe:
  enabled: Subscribe_Toggle
  type: Subscribe_Audience
  tags:
    - Subscribe_Tag_1
    - Subscribe_Tag_2
  orgs: Subscribe_Organizations
```

where:

- *Subscribe\_Toggle* must be `true` or `false`. If `false`, no developers will be able to subscribe to the Product.
- *Subscribe\_Audience* must be `authenticated`, in which case the Product's Plans can be subscribed to by anybody who is registered through the Developer Portal, or `custom`, in which case the Product's Plans can be subscribed to by a specified group of users.
- The *Subscribe\_Tag* variables are strings that contains any tags you want to attach to your Product. Each tag must be on a new line and preceded by a dash.
- *Subscribe\_Organizations* specifies the organizations that can subscribe to the Product's Plans if *Subscribe\_Audience* is set to `custom`. If *Subscribe\_Audience* is not set to `custom`, omit it or set as `[]`. If *Subscribe\_Audience* is `custom`, specify organizations in the following manner:

```
orgs:
  - Organization 1
  - Organization 2
```

where the *Organization* variables are the names of the organizations that are to be allowed to subscribe to the Product.

Note: If a Product is to be available for subscription, it must already be visible. As a result, for the subscription audience to be `authenticated`, the view audience cannot be `custom`.

## Results

You have completed the visibility section of your Product's YAML representation. It should have the following form:

```
visibility:
  view:
    enabled: View_Toggle
    type: View_Audience
    tags:
      - View_Tag_1
      - View_Tag_2
    orgs:
      - Organization_1
      - Organization_2

  subscribe:
    enabled: Subscribe_Toggle
    type: Subscribe_Audience
    tags:
      - Subscribe_Tag_1
      - Subscribe_Tag_2
    orgs:
      - Organization_1
      - Organization_2
```

where all variables are as described in previously in this topic. The indentation must be as in the example.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Referencing the APIs for your Product

Detail the file paths for the APIs you want to include in your Product in IBM® API Connect.

### About this task

Before an API can be included in a Plan, it must first be referenced in the APIs section of your Product description.

### Procedure

Begin the APIs section and reference the APIs you want to include in your Product. The syntax that you use depends on which user interface you are using, as follows:

- API Designer:

```
apis:
  API_1_NameAPI_1_Version
  $ref: API_1_File_Path
  API_2_NameAPI_2_Version
  $ref: API_2_File_Path
```

- API Manager:

```
apis:
  API_1_NameAPI_1_Version
  name: 'API_1_Name:API_1_Version'
  API_2_NameAPI_2_Version
  name: 'API_2_Name:API_2_Version'
```

where

- the *API\_n\_Name* variables are the case-sensitive names of your APIs.
- the *API\_n\_Version* variables are the versions of your APIs.
- the *API\_n\_File\_Path* variables are the file paths for the YAML files containing OpenAPI representations of your APIs.

The following example shows a complete sample APIs section:

```
apis:
  api1.0.0:
    name: 'api1:1.0.0'
  api2.0.0:
    name: 'api2:1.0.0'
  api3.0.0:
    name: 'api3:1.0.0'
```

For more information about creating OpenAPI definitions, see [Creating an OpenAPI definition file](#).

The indentation must be as in the examples and the APIs must have been created before they can be referenced.

## Results

You have referenced the APIs that are to be included in your Product. An example of a complete YAML representation of a Product can be found in [An example YAML representation of a Product](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Describing Plans in your Product

Describe the Plans that you want to include in your IBM® API Connect Product and which APIs they will contain, as well as any rate limits that apply.

### About this task

A Plan contains APIs and their operations. It can be used to implement rate limits and tailor visibility.

Note: If you are using more than one DataPower® server in a Gateway service, then to properly calculate API calls for rate limits the servers must be able to communicate with each other by using SLM peer groups, using either SLM unicast peering or SLM multicast peering depending on your network configuration. For more information, see [SLM peering](#).

In addition, you can apply burst limits to your Plans, to prevent usage spikes that might damage infrastructure. Multiple burst limits can be set per Plan, at second and minute time intervals. You can also set multiple rate limits per Plan and per operation, at second, minute, hour, day, and week time intervals.

Note: All keys and enumeration values that are specified in this topic are case-sensitive.

Two example Plan descriptions from a YAML representation of a Product can be found at the end of this topic. An example of a complete YAML representation of a Product can be found in [An example YAML representation of a Product](#).

## Procedure

To describe your Plans, include APIs in them, and set rate limits for Plans or specific operations, complete the following instructions:

1. Begin the Plans section with `plans`:
2. Under Plans, begin the description of your first Plan by providing a name, description, and specifying whether approval is required for subscription requests; use the following syntax:

```
plans:
  Plan_Name:
    title: Plan_Title
    description: Plan_Description
    approval: Approval_Toggle
```

where:

- *Plan\_Name* is the name of the Plan. It must be a single word and contain only alphanumeric characters, and the - (dash) and \_ (underscore) characters. The name is case-sensitive, and should be 20 characters or fewer so that it can be displayed in the API Manager user interface.
- *Plan\_Title* is the title of the Plan. Any string can be used, but the title should be kept short so that it can be displayed in the API Manager user interface.
- *Plan\_Description* is a description of your Plan.
- *Approval\_Toggle* must be either `true`, in which case approval is needed for subscription requests, or `false`, in which case subscriptions are automatically approved.

3. Optional: Under your Plan, add multiple burst and rate limits that will be shared across all the operations in the Plan; use the following syntax:

Note: For information about rate limits and burst limits in API Connect, see [Understanding rate limits for APIs and Plans](#).

```
rate-limits:
  Name:
    value: Rate_Limit
    hard-limit: Limit_Toggle
  Name:
    value: Rate_Limit
    hard-limit: Limit_Toggle
burst-limits:
  Name:
    value: Burst_Limit
```

where:

- *Name* is the name of the limit.
- *Rate\_Limit* is your rate limit. It can be for multiples of seconds, minutes, hours, days, or weeks, written as *second*, *minute*, *hour*, *day*, and *week* respectively; do not use the plural forms of the words. Use the syntax: **1/2minute**. If the time unit is singular, then it is not necessary to precede it with a number, for example, **1/minute**. If you do not want to apply a rate limit, set *Rate\_Limit* as *unlimited*.
- *Limit\_Toggle* must be *true* or *false*. If *true*, API calls by a developer will fail if the rate limit is exceeded. This step is not necessary if you have set *Rate\_Limit* to *unlimited*.
- *Burst\_Limit* is your burst limit. It can be for multiples of seconds or minutes, written as *second* or *minute*; do not use the plural forms of the words.

Note:

- Applying a rate limit at the Plan level creates a default rate limit that is shared across all the operations within the Plan. If you need to set specific rate limits for specific operations, you must set these within the operations themselves and these settings will override the setting at the Plan level.
- The test application that is used by the API Manager test tool is not subject to rate limits if you have enabled automatic subscriptions for the Catalog in which you are testing. For more information, see [Working with Catalogs](#)

4. Under your Plan, specify which APIs are to be included.

Reference APIs by name and version, as follows:

```
apis:
  API_1_nameAPI_1_version: {}
  API_2_nameAPI_2_version: {}
```

where

- the *API\_n\_name* variables are the case-sensitive names of your APIs.
- the *API\_n\_version* variables are the versions of your APIs.

5. Optional: If you want to include only a subset of an API's operations, list the operations that are to be included; use the following syntax:

```
apis:
  API_nameAPI_version:
    operations:
      - operation: Operation_1_Verb
        path: Operation_1_Path
      - operation: Operation_2_Verb
        path: Operation_2_Path
```

where

- the *Operation\_n\_Path* variables are the paths of operations that is to be included. The dash is required for each new operation.
- the *Operation\_n\_Verb* variables are the appropriate REST verbs for the operations.

6. Optional: If you want to specify multiple rate limits for a single operation; use the following syntax:

```
apis:
  API_nameAPI_version:
    operations:
      - operation: Operation_Verb
        path: Operation_Path
        rate-limits:
          Name:
            value: Operation_Limit
            hard-limit: Operation_Limit_Toggle
          Name:
            value: Operation_Limit
            hard-limit: Operation_Limit_Toggle
```

where:

- *Operation\_Limit* is the rate limit that you want to apply to your operation. It can be for multiples of seconds, minutes, hours, days, or weeks, written as *second*, *minute*, *hour*, *day*, and *week* respectively; do not use the plural forms of the words. Use the syntax: **1/2minute**. If the time unit is singular, then it is not necessary to precede it with a number, for example, **1/minute**. If you do not want to apply a rate limit, set *Operation\_Limit* as *unlimited*.
- *Operation\_Limit\_Toggle* must be *true* or *false*. If *true*, API calls by a developer will fail if the rate limit is exceeded. This step is not necessary if you have set *Operation\_Limit* to *unlimited*.

## Results

You have described the Plans that are to be included in your Product. Your Plans section should be similar to the following scenarios.

If you have enforced a rate limit for your Plan and not specified individual operations, your Plans section should have the following form:

```
plans:
  Plan_Name:
    title: Plan_Title
    description: Plan_Description
    approval: Approval_Toggle
    rate-limits:
      Name:
        value: Limit
        hard-limit: Limit_Toggle
  apis:
    API_1_nameAPI_1_version: {}
    API_2_nameAPI_2_version: {}
```

If you have enforced rate limits at the Plan level and want to include two operations, with a separate rate limit one for one of the operations, your Plans section should have the following form:

```
plans:
  Plan_Name:
    title: Plan_Title
    description: Plan_Description
    approval: Approval_Toggle
    rate-limits:
      Name:
        value: API_Limit
        hard-limit: API_Limit_Toggle
```

```

burst-limits:
  Name:
    value: Burst_Limit
apis:
  API_nameAPI_version:
    operations:
      - operation: Operation_1_Verb
        path: Operation_1_Path
        rate-limits:
          Name:
            value: Operation_Limit
            hard-limit: Operation_Limit_Toggle
      - operation: Operation_2_Verb
        path: Operation_2_Path

```

In both examples, variables are as in the previous steps and the indentation must be as presented.

The following example shows a complete sample Plans section:

```

plans:
  default:
    title: Default Plan
    description: Default Plan
    approval: false
    rate-limits:
      default:
        value: 100/hour
        hard-limit: false
  my-plan:
    title: my_plan
    rate-limits:
      Default rate-limit:
        value: 50/1hour
        hard-limit: false
    burst-limits:
      Default burst-limit:
        value: 10/1second
  apis:
    api21.0.0:
      operations:
        - operation: get
          path: /path2
    api31.0.0:
      operations:
        - operation: get
          path: /path3
      rate-limits:
        limit3:
          value: 10/1second
          hard-limit: true
        limit4:
          value: 100/1hour
          hard-limit: true
    api11.0.0: {}

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## An example YAML representation of a Product

Products in IBM® API Connect can be represented by using a YAML file in a similar fashion to how APIs can be represented by using OpenAPI.

The following code describes a complete Product, with sample values.

```

info:
  version: 1.0.0
  title: SampleProduct
  name: SampleProduct
  summary: This is a sample product
gateways:
  - datapower-gateway
product: 1.0.0
visibility:
  view:
    enabled: true
    type: public
    tags: []
    orgs: []
  subscribe:
    enabled: true
    type: authenticated
    tags: []
    orgs: []
plans:
  default:
    title: Default Plan
    description: Default Plan
    approval: false

```

```

rate-limits:
  default:
    value: 100/hour
    hard-limit: false
newplan:
  title: NewPlan
  rate-limits:
    low:
      value: 10/1hour
      hard-limit: true
  burst-limits:
    Default burst-limit:
      value: 10/1second
apis:
  SampleAPI:
    name: 'Sampleapi:1.0.0'

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using `x-ibm-languages` to create multilingual API and Product documentation

Create multilingual API and Product documentation by using the `x-ibm-languages` extension in your API and Product OpenAPI definitions.

You can scale your API initiatives to a global user base, while still maintaining a single API definition. Translations are custom configured and controlled, so that they can be tailored to both your API documentation and the API consumer. The extensions that are used in API Connect are added directly to the YAML file as `x-ibm-languages`. These extensions are then used in the Developer Portal, so that whatever language the Developer Portal is set to reflects the translations that are contained in the API and Product definitions.

Add the following syntax into your API or Product YAML file at every point that you require some translated text:

```

x-ibm-languages:
  item_name:
    language_code: translated_text

```

Where:

- `item_name` is the name of the item that you want to translate, for example `summary`.
- `language_code` is the ISO code for the translation language. Supported languages with their ISO codes are listed in the following table.

Table 1. List of supported language codes

Language	Code
Chinese, simplified	zh_cn
Chinese, traditional	zh_tw
Dutch	nl
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Portuguese	pt
Spanish	es
Turkish	tr

- `translated_text` is the translated text that you want to be displayed for the item in the Developer Portal.

Note: The Developer Portal automatically assumes that the default language in the OpenAPI definition is English. If you want to use a different default language in the definition, you must provide both the English translation and the default language text in the `x-ibm-languages` extension sections of your OpenAPI definition, by using the following syntax:

```

x-ibm-languages:
  item_name:
    en: English translated_text
    default_language_code: default_language_text

```

For an example of how to write an OpenAPI YAML file with French as the default language, see the [Examples](#) section.

## Examples

An example API OpenAPI YAML file that contains the `fr` language extension, where English is the default language:

```

swagger: '2.0'
info:
  x-ibm-name: climbing-weather-api
  title: Climbing Weather API
  version: 1.0.0
  x-ibm-languages:
    title:

```

```

    fr: Climat d'escalade
schemes:
  - https
host: ${catalog.host}
consumes:
  - application/json
produces:
  - application/json
x-ibm-configuration:
  assembly:
    execute:
      - invoke:
          target-url: 'https://1234.com'
          title: 3 day forecast invocation
          cache-response: time-to-live
          cache-key: ${request.search}
gateway: datapower-gateway
enforced: true
testable: true
phase: realized
cors:
  enabled: true
paths:
  /weather/forecast:
    get:
      summary: Retrieve the 3 day forecast for a location
      description: Retrieve the locations weather forecast descriptions for the next 3 days and nights
      operationId: getWeather
      x-ibm-languages:
        summary:
          fr: Récupérer la prévision de 3 jours pour un emplacement
        description:
          fr: Récupérer les descriptions des prévisions météo pour les 3 prochains jours et les nuits
      tags:
        - Weather
      parameters:
        - name: zip
          type: string
          in: query
          description: A 5 number zip code
          x-ibm-languages:
            description:
              fr: Un code postal à 5 numéros
        - name: country
          type: string
          in: query
          description: A 2 letter country code
          x-ibm-languages:
            description:
              fr: Un code de pays de 2 lettres
        - name: lat
          type: string
          in: query
          description: A latitude value between -90 and 90
          x-ibm-languages:
            description:
              fr: Une valeur de latitude entre -90 et 90
        - name: lon
          type: string
          in: query
          description: A longitude value between -180 and 180
          x-ibm-languages:
            description:
              fr: Une valeur de longitude entre -180 et 180
      responses:
        '200':
          description: Success
          x-ibm-languages:
            description:
              fr: Succès
        '400':
          description: Bad Request
          x-ibm-languages:
            description:
              fr: Mauvaise Demande
        '408':
          description: Request Timeout
          x-ibm-languages:
            description:
              fr: Délai de délai de demande
        '500':
          description: Internal Server Error
          x-ibm-languages:
            description:
              fr: Erreur Interne du Serveur
basePath: /
tags:
  - name: Weather
    description: Sample API to get weather forecast data
    x-ibm-languages:
      description:
        fr: Exemple d'API pour obtenir des données météorologiques
securityDefinitions:
  client-secret:
    type: apiKey
    description: ''

```

```

    in: header
    name: X-IBM-Client-Secret
  client-id:
    type: apiKey
    description: ''
    in: header
    name: X-IBM-Client-Id
  security:
    - client-secret: []
    - client-id: []

```

An example Product OpenAPI YAML file that contains the `fr` language extension, where English is the default language:

```

product: 1.0.0
info:
  name: climbing-weather
  title: Climbing Weather
  version: 1.0.0
  description: This is a product about weather.
  x-ibm-languages:
    title:
      fr: Météo traduite
    description:
      fr: C'est un produit sur la météo.
    termsOfService:
      fr: Quid ergo aliud intellegetur en francais.
  contact:
    name: Ralph Renli
    email: ralph.renli@example.com
    url: 'https://weather.example.com/climbing/info'
  license:
    name: MIT License
    url: 'https://choosealicense.com/licenses/mit/'
    x-ibm-languages:
      name:
        fr: License de MIT
    termsOfService: Quid ergo aliud intellegetur nisi uti ne quae pars naturae neglegatur? Quis est tam dissimile homini. De
    quibus cupio scire quid sentias.
  categories:
    - Portal/Testing/Language
    - Portal/Testing/Language/UTF8
    - Portal/Testing/Weather
  visibility:
    view:
      enabled: true
      type: public
      tags: []
      orgs: []
    subscribe:
      enabled: true
      type: authenticated
      tags: []
      orgs: []
  apis:
    climbing-weather:
      id: 593f972be4b06fb4e0dce879
  plans:
    a-call-a-day:
      title: A call a day
      description: "One call every day. That's it!"
      x-ibm-languages:
        title:
          fr: "Un appel par jour"
        description:
          fr: "Un appel tous les jours. C'est tout!"
      apis: {}
      rate-limits:
        One Call Per Day:
          hard-limit: true
          value: 1/1day
    ten-calls-per-day:
      title: 10 Calls a day
      description: '10 times more calls than the next best competitor plan!'
      x-ibm-languages:
        title:
          fr: "10 appels par jour"
        description:
          fr: "10 fois plus d'appels que le prochain meilleur plan concurrent!"
      apis: {}
      rate-limits:
        10 calls per day:
          hard-limit: true
          value: 10/1minute
    11-calls-every-day:
      title: 11 Calls a day
      description: 'Need just one more call? Then this is the plan for you!'
      x-ibm-languages:
        title:
          fr: "11 appels par jour"
        description:
          fr: "Vous n'avez besoin que d'un appel supplémentaire? Alors c'est le plan pour vous!"
      apis: {}
      rate-limits:
        11 calls per day:
          hard-limit: true
          value: 11/1day

```



```

25-calls-per-day:
  title: 25 Calls per day
  description: So you want even more calls?
  apis: {}
  x-ibm-languages:
    title:
      fr: "25 appels par jour"
    description:
      fr: "Vous voulez donc encore plus d'appels?"
  rate-limits:
    25 calls per day:
      hard-limit: true
      value: 25/1day
100-calls-per-day:
  title: 100 Calls every day
  description: 'You get 100 calls each and every day!'
  x-ibm-languages:
    title:
      fr: "100 appels par jour"
    description:
      fr: "Vous recevez 100 appels chaque jour"
  apis: {}
  rate-limits:
    100 Calls per day:
      hard-limit: true
      value: 100/1day

```

An example of how to write an OpenAPI YAML file with French as the default language:

```

info:
  name: climat-d-escalade
  title: Climat d'escalade
  version: 1.0.0
  x-ibm-languages:
    title:
      en: Climbing Weather API
      fr: Climat d'escalade
...

```

This example shows that both the default French language, and the English language translation, must be provided in the `x-ibm-languages` section.

## Related information

- [Expand your API Initiatives Globally with Multi-Lingual Support of API and Product Definitions](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using `x-example` to control the examples displayed in the Developer Portal

OpenAPI does not support the use of `example` attributes on parameters, only on request and response objects and their properties. To allow API Developers to be able to control the examples displayed in the portal you can use `x-example` in OpenAPI parameters.

In this example, you cannot use `example` as it isn't supported by the OpenAPI spec, but you can use `x-example`.

```

- name: city
  in: query
  required: false
  type: string
  maxLength: 50
  description: "Location"
  x-example: "New York"

```

An `example` attribute is honored where set and allowed by the spec. If `example` is not found, `default` is honored where set and allowed by the spec. An `x-example` attribute is honored if found, if not found, a random example to match the schema is generated.

If the spec allows you to use `example`, then you should use it in preference to `x-example`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using `x-embedded-doc` to add additional documentation to products and APIs

You can use `x-embedded-doc` to add additional documentation to a product or an API as part of their definition.

You can embed additional documentation pages within the api swagger documentation. These pages can be base64 encoded strings or markdown strings of the html content of the page. These pages show up in the navigation, on the left side, underneath the overview for that product or API, and above the paths. Any styling that is

needed to theme the documentation can be included in the Developer Portal custom theme.

## Embedded docs for products and APIs

You can embed documentation for individual products and APIs by specifying an additional `x-embedded-doc` object in the product or API specification.

Field name	Type	Description
name	string	Required, the name of the embedded document
title	string	Required, The display title of the embedded document
format	string	Optional, the format of the embedded document content value. Values are <code>b64html</code> or <code>md</code> . Default value is <code>md</code> .
content	string	The base64 encoded html of page, or, the markdown string.
docs	x-embedded-doc	Additional embedded documentation. If no content is specified for parent, then the first child is the content that is shown.

Hierarchies of documents are supported by nesting of further doc properties. Language support is available by using `x-ibm-languages`.

## Examples

Documentation can be added to an api, by using the following JSON:

```
"x-embedded-doc": [
  {
    "name": "introduction",
    "title": "Introduction",
    "format": "md",
    "content": "# some markdown"
  },
  {
    "name": "concepts",
    "title": "Concepts",
    "format": "md",
    "content": "## more markdown"
  },
  {
    "name": "authentication",
    "title": "Authentication",
    "docs": [
      {
        "name": "auth_clientid_secret",
        "title": "Obtaining a Client ID and Secret",
        "format": "md",
        "content": "### even more markdown"
      },
      {
        "name": "auth_bearertoken",
        "title": "Getting and Using a Bearer Token",
        "format": "md",
        "content": "#### yet more markdown"
      }
    ]
  }
]
```

Alternatively, by using base 64 encoded HTML:

```
"x-embedded-doc": [
  {
    "name": "introduction",
    "title": "Introduction",
    "format": "b64html",
    "content": "PGFydG1jbGUgaWQ9ImIudHJvZHVjdGlvbiIqY2xhc3M...."
  },
  {
    "name": "concepts",
    "title": "Concepts",
    "format": "b64html",
    "content": "PGFydG1jbGUgaWQ9ImNvbmlcHRzIiBjbGFzc0icGFnZSI...."
  },
  {
    "name": "authentication",
    "title": "Authentication",
    "docs": [
      {
        "name": "auth_clientid_secret",
        "title": "Obtaining a Client ID and Secret",
        "format": "b64html",
        "content": "PGFydG1jbGUgaWQ9ImNsaWVudC1pZC1zZWNYZX...."
      },
      {
        "name": "auth_bearertoken",
        "title": "Getting and Using a Bearer Token",
        "format": "b64html",
        "content": "ICA8YXJ0aWNsZSBpZD0iYmVhcmVYLXRva2VuIiBjbGFz...."
      }
    ]
  }
]
```

Or mixed formats:

```
"x-embedded-doc": [
  {
    "name": "introduction",
    "title": "Introduction",
```

```

"format": "b64html",
"content": "PGFydGljbGUgaWQ9ImludHJvZHVjdGlubiIqY2xhc3M..."
},
{
"name": "concepts",
"title": "Concepts",
"format": "md",
"content": "# some markdown"
}
]

```

An example of some `x-embedded-doc` code and how it might render:

```

x-embedded-doc:
- name: document
  title: Interesting
  format: b64html
  content: >-
    PGFydGljbGUgaWQ9ImNsaWVudC1pZ...=
securityDefinitions:
  api_key:
    type: apiKey
    in: header
    name: api_key

```

[Products / My Doc APIs](#)



**Overview**

**Interesting**

GET /pet/{petId}

POST /pet/{petId}

DELETE /pet/{petId}

POST /pet/{petId}/uploadImage

POST /pet

PUT /pet

GET /pet/findByStatus

GET /pet/findByTags

GET /store/inventory

## Obtaining a Client ID and Secret

---

Each client app that accesses the API Connect REST API must be registered and configured to determine which operations the app may access.

To register a client app, use the `apic registrations:create` API Connect REST API.

**name**  
short name to identify the client

**title**  
display name of the client

**client\_id**  
a generated client ID value

**client\_secret**  
a generated client secret value

**client\_type**  
one of portal, gateway, toolkit, consumer\_toolkit, ui, consumer\_ui, ibm\_cloud

An example of some `x-embedded-doc` code into a product and how it might render:

```

x-embedded-doc:
- name: introduction
  title: Introduction
  format: md
  content: >-
    ## Overview

    ### Philosophy

    Markdown is intended to be as easy-to-read and easy-to-write as is
    feasible. Compellingly facilitate 2.0 intellectual capital for
    cross-platform networks. Dynamically incubate ethical sources after
    optimal technologies. Phosfluorescently restore global users via excellent
    niche markets. Credibly impact cross-media mindshare through strategic
    best practices. Synergistically impact interdependent web services rather
    than unique interfaces.

```

Continually reintermediate enabled sources rather than professional architectures. Appropriately embrace viral collaboration and idea-sharing whereas granular solutions. Energistically revolutionize competitive paradigms vis-a-vis highly efficient models. Objectively incentivize enterprise customer service through compelling e-markets. Competently grow multifunctional e-markets with business methods of empowerment.

Dramatically iterate equity invested scenarios and focused data. Compellingly reconceptualize next-generation deliverables rather than out-of-the-box architectures. Competently foster cross-unit web services with interactive innovation. Quickly revolutionize enabled communities and alternative content. Globally administrate resource maximizing deliverables whereas corporate e-business.

Progressively develop maintainable leadership.

```
- name: concepts
  title: Concepts
  format: md
  content: '## some more markdown'
- name: moreinfo
  title: Even more info
  docs:
    - name: moreinfo1
      title: More Info 1
      format: md
      content: '### even more markdown'
    - name: moreinfo2
      title: More Info 2
      format: md
      content: '#### yet more markdown'
```

Products /

## My Product with Docs 1.0.0 ★★★★★

### Introduction

- Concepts
- Even more info
  - More Info 1
  - More Info 2
- APIs and Plans

## Overview

### Philosophy

Markdown is intended to be as easy-to-read and easy-to-write as is feasible. Compellingly facilitate 2.0 intellectual capital for cross-platform networks. Dynamically incubate ethical sources after optimal technologies. Phosfluorescently restore global users via excellent niche markets. Credibly impact cross-media mindshare through strategic best practices. Synergistically impact interdependent web services rather than unique interfaces. Continually reintermediate enabled sources rather than professional architectures. Appropriately embrace viral collaboration and idea-sharing whereas granular solutions. Energistically revolutionize competitive paradigms vis-a-vis highly efficient models. Objectively incentivize enterprise customer service through compelling e-markets. Competently grow multifunctional e-markets with business methods of empowerment. Dramatically iterate equity invested scenarios and focused data. Compellingly reconceptualize next-generation deliverables rather than out-of-the-box architectures. Competently foster cross-unit web services with interactive innovation. Quickly revolutionize enabled communities and alternative content. Globally administrate resource maximizing deliverables whereas corporate e-business. Progressively develop maintainable leadership.

## Related reference

- [Using x-ibm-languages to create multilingual API and Product documentation](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using x-pathalias to give consistent URLs for products and APIs

You can use `x-pathalias` to assign a URL to a product or an API.

The information section of a YAML representation of a Product or API can include `x-pathalias`. This option can be useful as it allows the same URL to be used on different revisions of your product. It also allows the use of a vanity URL, which might better represent the branding of your site, or make the URL easier to pronounce.

In this example, you set an API called `Climbing weather API` in product `Weather watch` to have a URL of `portalURL/product/weather/api/climbing`.

Product information:

```
info:
  version: 1.0.0
  title: Weather watch
  name: weather-watch
  x-pathaliases: weather
```

API information:

```
info:
  title: Climbing Weather API
  x-ibm-name: climbing-weather-api
  version: 1.0.0
  x-pathaliases: climbing
```

## Related tasks

---

- [Editing an API definition](#)

## Related information

---

- [Creating and configuring Catalogs](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Validating the YAML or JSON definition of an API or Product

You can validate a YAML or JSON definition by using the IBM® API Connect developer toolkit.

## Before you begin

---

To complete the steps that are described in this topic, you must have installed the developer toolkit. For more information, see [Installing the toolkit](#).

## Procedure

---

To perform validation by using the developer toolkit, enter the following command:

```
apic validate filename
```

where *filename* is the file name of the API definition file you want to validate.

- Include `--product-only` to validate only a Product definition and not any APIs that it references.
- Include `--no-extensions` to validate only the default OpenAPI section of the API and none of its extensions.

Note:

- If the OpenAPI file that defines your API uses a `$ref` field to reference a fragment of OpenAPI code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the draft API is created with the `apic drafts:validate` command. For more information, see [Using \\$ref to reuse code fragments in your OpenAPI files](#).
- Products that contain an API with a Swagger property using `regex` that include lookahead assertions, such as "(?)" cannot be validated or published. An error message is returned. For example:

```
Product has not been published!
The multipart 'openapi' field contains an OpenAPI definition with validation errors.
  definitions.properties.pattern Does not match format 'regex' (context: (root).definitions.properties.pattern, line: 0,
col: 0)
400
```

## Related concepts

---

- [Creating an OpenAPI definition file](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Creating and using API and Product definitions templates

You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition.

## Before you begin

To complete the steps that are described in this topic, you must have installed the IBM® API Connect developer toolkit. For more information, see [Installing the toolkit](#).

## Procedure

1. Create a Product or API definition template, either from scratch, or by copying one of the examples provided in [API and Product definition template examples](#). The template file must have `.hbs` filename extension, and may contain any of the handlebars variables described in [Template variables for API and Product definitions](#).
2. To create a Product or API definition from a template, enter the following command:

```
apic create:[api | product ] --template template_file --title product_title options
```

where *template\_file* is the template `.hbs` file to use, *product\_title* is the title of the Product to create, and *options* is any additional command-line options. The path to the template file can be either an absolute path or relative to the location where the command is executed.

When the Product or API definition is created, the value of each command-line option is substituted for the corresponding handlebars template variable. For example, the value of the required `--title` option is substituted for the `info.title` field in the template file. The command creates a Product definition YAML file with the name specified in the `--name` option. If you don't supply the `--name` option, the command derives the name of the Product YAML file from the specified title by down-casing the title and replacing spaces with dashes.

## Related reference

- [Template variables for API and Product definitions](#)
- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OpenAPI 3.0 support in IBM API Connect

From Version 2018.4.1.4, IBM® API Connect supports the OpenAPI 3.0 specification, with some limitations.

### Overview

A Product can contain any combination of OpenAPI 2.0 and OpenAPI 3.0 APIs. When you publish a Product that contains an OpenAPI 3.0 API, that API is validated to ensure that it is syntactically correct, and that references to configuration resources and policies resolve correctly, in the same way that OpenAPI 2.0 APIs are validated.

You can also validate OpenAPI 3.0 APIs in your local file system by using the `apic validate` command provided by the developer toolkit CLI; for details, see [Validating the YAML or JSON definition of an API or Product](#).

If you retrieve an API object by using the developer toolkit CLI or the API Connect REST APIs, there is an `oai_version` property that defines which OpenAPI version the API represents.

There is no OpenAPI 3.0 API support with the DataPower® Gateway (v5 compatible); OpenAPI 3.0 API support is provided by the DataPower API Gateway **only**.

### Limitations

The limitations to the OpenAPI 3.0 support in IBM API Connect are as follows:

- User interface limitations.
  - There is no support for working with draft OpenAPI 3.0 APIs in the API Manager or API Designer user interfaces. You must stage or publish OpenAPI 3.0 APIs directly by using the `apic products:publish` developer toolkit CLI command, or the API Connect REST APIs.
  - There is no support for editing OpenAPI 3.0 APIs in the API Manager user interface.
  - There might be some rendering errors in the Developer Portal test tool when OpenAPI 3.0 examples or schemas are displayed.
- Developer toolkit limitations.
  - You cannot create a draft API on a management server by uploading a local OpenAPI 3.0 API definition file with the `apic draft-apis:create` command. You can, however, stage or publish OpenAPI 3.0 APIs directly by using the `apic products:publish` developer toolkit CLI command, or the API Connect REST APIs.
- General limitations.
  - The `servers` array cannot contain more than one server.
  - The `url` entry in the `servers` array cannot contain `variables`.
  - `path` objects cannot contain server object overrides.
  - Wildcarding in response objects is not supported for error codes or success codes.
  - There is no support for converting a WSDL defined SOAP service into an OpenAPI 3.0 API.
- Security.
  - The following security schemes are not supported:
    - HTTP JWT Bearer (HTTP Basic scheme is supported).
    - OAuth2 with multiple flows (a single flow is supported).
    - OpenID Connect (OIDC).
  - The use of cookies in an API key is not supported.

- Assembly policies.
  - Only the following policies are supported:
    - `invoke`
    - `jwt-generate`
    - `jwt-validate`
    - `oauth`
    - `throw`
    - `user-security`
  - The `validate` and `map` policies are not supported.
  - Schema support is limited to Draft 4 and earlier. Therefore, the following policies are limited to those drafts:
    - `gatewayscript`
    - `set-variable`
    - `switch`
    - `json-to-xml`
    - `xml-to-json`
    - `parse`
    - `xslt`

For draft specification details, see <https://json-schema.org/specification-links.html>.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with global policies

Use global policies to configure policy assemblies that are called just before, or just after, each of your API assemblies is called. You can upload global policies into each of the gateway services in your Catalogs, and then designate, for each gateway service, which global policy is to be called before an API assembly is called, and which one is to be called after. The designated global policies are applied to all the APIs that are deployed to the associated gateway service.

## Prerequisites

IBM® DataPower® Gateway 7.7.1.3 or later.

To complete the global policy management tasks described in this topic, you must be the owner of the provider organization, or be assigned a role that has the **Settings:Manage** permission. The pre-supplied Administrator role has this permission by default; if you are assigned a custom role it must have this permission. For more information, see [Creating custom roles](#).

## Overview

A global policy is defined in a .yaml file. The global policy file specifies the policy assembly that you want to call. A global policy file has the following structure:

```
global-policy: 1.0.0

info:
  name: global_policy_name
  title: global_policy_title
  version: global_policy_version

gateways:
  - gateway_type

assembly:
  - policy_assembly_to_be_called
```

where *gateway\_type* has the value `datapower-gateway` if you want to use the global policy in a DataPower Gateway (v5 compatible), and `datapower-api-gateway` if you want to use the global policy in a DataPower API Gateway. For more information on the different types of gateway, see [API Connect gateway types](#). The `assembly` section can contain any valid API assembly flow. For more information, see the [execute](#) section.

For each of the gateway services in a Catalog, a global policy that you want to be called before any API assembly is called is designated as a *pre-request* global policy, and a global policy that you want to be called after any API assembly is called is designated as a *post-response* global policy. You can upload as many global policies into a gateway service as you want, but you can designate only one global policy to be pre-request, and one to be post-response, at any one time.

Note:

- If you are using the DataPower Gateway (v5 compatible), there can be only one **proxy** policy across the combined pre-request, API, and post-response assembly flows.
- Catalog properties, as described in [Creating and configuring Catalogs](#), and API properties, as described in [Setting API properties](#), are not supported with global policies. Therefore, if you use such properties in a global policy, they are **not** replaced with the values specified in the API or Catalog property definitions.

## Sample global policy .yaml file

```
global-policy: 1.0.0

info:
  name: sample-policy
  title: Sample Policy
  version: 2.0.0

gateways:
```

```

- datapower-gateway

assembly:
  execute:
    - invoke:
        target-url: http://myhost.com/mytarget
        version: 1.0.0

  catch:
    - errors:
        - ConnectionError
    execute:
      - activity-log:
          title: activity-log
          content: activity
          error-content: payload
          version: 1.0.0

```

## Logging in to the Management server

You work with global policies by using developer toolkit CLI commands. Before you can run the commands, you must log in to your management server from the CLI. Use the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

## Uploading a global policy to a gateway service

You upload a global policy to a gateway service by using the `apic global-policies:create` command. The command has the following format:

```
apic global-policies:create --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space] filename
```

where:

- `catalog_name` is the value of the `name` property of the Catalog that contains the gateway service to which you want to upload the global policy.
- `gateway_service_name` is the value of the `name` property of the gateway service.
- `organization_name` is the value of the `name` property of the provider organization that contains the Catalog.
- `mgmt_endpoint_url` is the platform API endpoint URL.
- `scope` has one of the following values:
  - `catalog` if the Catalog does not have Spaces enabled.
  - `space` if the Catalog has Spaces enabled. If you specify `space` for the `--scope` parameter you must also supply the `--space` parameter to specify the Space that contains the gateway service.
- (optional) `space` is the name of the Space that contains the gateway service. The `--space` parameter is required if the Catalog has Spaces enabled, in which case you must also include `--scope space` in the command.
- `filename` is the name of the global policy .yaml file that you want to upload.

Note: If Spaces are enabled in a Catalog, a global policy that you apply to one Space is applied to **all** Spaces; you cannot apply a global policy to an individual Space in the Catalog. Any subsequent updates are also applied to all Spaces.

You can confirm the global policies that have been uploaded by using the `apic global-policies:list-all` command, which has the following format:

```
apic global-policies:list-all --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space]
```

For reference details of the `apic global-policies` commands, see [apic global-policies](#).

## Designating the pre-request global policy

To designate the pre-request policy for a gateway service, complete the following steps:

1. Write the URL of the required global policy to a .yaml file by using the following command:

```
apic global-policies:get --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space] policy_name:policy_version --fields url
```

where `policy_name:policy_version` specifies the name and version number of the required global policy, as defined in the global policy .yaml file that was originally uploaded to the gateway service.

The value of the URL is written to a file called `GlobalPolicy.yaml`.

2. Edit the `GlobalPolicy.yaml` file and replace the string `url` with `global_policy_url`. The resulting file has the following format:

```
global_policy_url: >-
  https://server_host_name/api/catalogs/catalog_id/configured-gateway-services/gateway_service_id/global-policies/policy_id
```

3. Designate the global policy to be the pre-request policy by using the following command:

```
apic global-policy-prehooks:create --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space] GlobalPolicy.yaml
```

To replace the designated pre-request global policy with another policy, repeat steps [1](#) and [2](#), then use the `apic`

`global-policy-prehooks:update` command. To remove the pre-request global policy designation, use the `apic global-policy-prehooks:delete` command.

For reference details of the `apic global-policy-prehooks` commands, see [apic global-policy-prehooks](#).



## Designating the post-response global policy

---

To designate the post-response policy for a gateway service, complete the following steps:

1. Write the URL of the required global policy to a .yaml file by using the following command:

```
apic global-policies:get --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space] policy_name:policy_version --fields url
```

where `policy_name:policy_version` specifies the name and version number of the required global policy, as defined in the global policy .yaml file that was originally uploaded to the gateway service.

The value of the URL is written to a file called GlobalPolicy.yaml.

2. Edit the GlobalPolicy.yaml file and replace the string `url` with `global_policy_url`. The resulting file has the following format:

```
global_policy_url: >-
  https://server_host_name/api/catalogs/catalog_id/configured-gateway-services/gateway_service_id/global-policies/policy_id
```

3. Designate the global policy to be the post-response policy by using the following command:

```
apic global-policy-posthooks:create --catalog catalog_name --configured-gateway-service gateway_service_name --org organization_name --server mgmt_endpoint_url --scope scope [--space space] GlobalPolicy.yaml
```

To replace the designated post-response global policy with another policy, repeat steps 1 and 2, then use the `apic global-policy-posthooks:update` command. To remove the post-response global policy designation, use the `apic global-policy-posthooks:delete` command.

For reference details of the `apic global-policy-posthooks` commands, see [apic global-policy-posthooks](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Reference

Reference information for developing your APIs in API Connect, including context variables, and template variables.

Note: For product-wide reference material, including command-line tool and REST APIs documentation, see [Reference information for IBM® API Connect](#).

- [API Connect context variables](#)  
List of IBM API Connect context variables that you can reference when defining default parameter values in an assembly operation, or by using the `getContext()` function when defining a policy.
- [OAuth context variables](#)  
You can customize the OAuth security flow in a native OAuth provider by adding multiple OAuth policies to the assembly. Each OAuth policy takes input from context variables and writes output to context variables. By using these OAuth context variables you can manipulate the original request, and process the output from an OAuth policy.
- [API and Product definition template examples](#)  
You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition. This topic provides the default Handlebar templates used by `apic` to create Products and APIs as examples you can use to copy and customize for your own use.
- [Template variables for API and Product definitions](#)  
You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition.

## Related information

---

- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## API Connect context variables

List of IBM® API Connect context variables that you can reference when defining default parameter values in an assembly operation, or by using the `getContext()` function when defining a policy.

The following tables are presented:

- [General context variables](#).
- [OAuth context variables - DataPower Gateway \(v5 compatible\)](#).
- [Application certificate context variables](#).
- [API activity logging context variables](#).

For more information about implementing an assembly component, see [Including components in your assembly](#), and for information about how to reference context variables in API Connect see [Variable references in API Connect](#) and [Using context variables in GatewayScript and XSLT policies with the DataPower API Gateway](#).

For more information about creating a user-defined policy, see [Authoring policies](#).

## General context variables

Note:

- For plan variables (such as `plan.name` or `plan.version`), plan information is available only when the requested operation requires identification and the client passes the authentication check.
- If you deploy your API to the DataPower® Gateway (v5 compatible) then, with the exception of client ID and client secret, the passing of form input as a parameter into an API is not supported. This restriction does not apply if you deploy your API to the DataPower API Gateway.

Table 1. API Connect context variables

Name	Description
<code>api.catalog.id</code>	The ID of the Catalog in which the API is published.
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> <code>api.catalog.name</code>	The name of the Catalog in which the API is published.
<span style="border: 1px solid green; padding: 2px;">API Gateway only</span> <code>api.catalog.path</code>	The path segment that represents this Catalog.
<span style="border: 1px solid blue; padding: 2px;">DataPower Gateway (v5 compatible) only</span> <code>api.document</code>	The OpenAPI document.
<code>api.endpoint.address</code>	The address of the API Gateway endpoint.
<code>api.endpoint.hostname</code>	The host name of the API Gateway endpoint, as requested by the application.
<code>api.name</code>	The name of the API; this corresponds to the <code>x-ibm-name</code> field in the OpenAPI definition for the API.
<code>api.operation.id</code>	The ID of the operation.
<code>api.operation.path</code>	The path of the operation.
<code>api.org.id</code>	The organization ID of the API provider.
<code>api.org.name</code>	The organization short name of the API provider.
<code>api.properties.propertyname</code>	<p>The name of a custom API property. Property values are Catalog specific.</p> <p>Note:</p> <ul style="list-style-type: none"> <li>• You have write permission to a custom property <b>only</b> from the user interface, <b>not</b> from GatewayScript.</li> <li>• To access a Catalog specific property value from GatewayScript, you must refer to the property by using the following syntax:</li> </ul> <pre>apim-catalog-name</pre> <p>where <i>catalog</i> is the name of the Catalog, and <i>name</i> is the property name. For example:</p> <pre>var mypropertyvalue = \$(apim-mycatalog-mypropertyname)</pre>
<code>api.root</code>	The API basepath.
<code>api.type</code>	The API type; REST or SOAP.
<code>api.version</code>	The version string of the API.
<code>client.app.id</code>	The client ID or application key that is received on the request.
<code>client.app.lifecycle-state</code>	<p>The lifecycle status of the calling client application. The possible values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>DEVELOPMENT</b></li> <li>• <b>PRODUCTION</b> (default)</li> </ul>
<code>client.app.metadata.key</code>	<p>The string value of an application metadata key, where <i>key</i> is the name of the key.</p> <p>You can add metadata keys to an application by using either the <code>apic apps:create</code> or the <code>apic apps:update</code> command; you include the metadata keys in the configuration file parameter that is passed to the command.</p> <p>For example:</p> <pre>apic apps:create myapp.yaml --server myserver.com --org myorg --catalog mycatalog --consumer-org mycorg</pre> <p>where myapp.yaml contains the following:</p> <pre>name: myapp title: My test application metadata:   key1: value1   key2: value2   key3: value3   key4: value4   key5: value5</pre> <p>You can then retrieve the value of a metadata key in an API assembly policy by using a context variable such as the following:</p> <pre>client.app.metadata.key3</pre> <p>Note that adding application metadata might impact gateway transaction performance.</p>
<code>client.app.name</code>	The name of the application that is identified as having issued the request.
<code>client.app.secret</code>	The client secret that is received in the request.
<code>client.org.id</code>	The unique identification key of the organization that owns this application.
<code>client.org.name</code>	The name of the organization that owns this application.

Name	Description
<b>API Gateway only</b> client.result	The result of the client security policy, which is <b>SUCCESS</b> or <b>FAILURE</b> .
<b>API Gateway only</b> client.third_party.type	The type of user registry used for third-party authentication of the extracted client credentials. The possible values are <b>LDAP</b> and <b>auth-url</b> .
<b>API Gateway only</b> client.third_party.headers	The array of headers added to the request that was sent to that API authentication URL during third-party authentication.
<b>API Gateway only</b> client.third_party.response.authenticating	The third-party authentication results. The possible values are as follows: <ul style="list-style-type: none"> <li><b>true</b>: the authentication was successful.</li> <li><b>false</b>: the authentication failed.</li> </ul>
<b>API Gateway only</b> client.third_party.response.user	The user for third-party authentication.
<b>API Gateway only</b> client.title	The title for the credentials that are received in the request.
<b>DataPower Gateway (v5 compatible) only</b> env.name	The name of the Catalog in which the API is published.
<b>DataPower Gateway (v5 compatible) only</b> env.path	The path segment that represents this Catalog.
message.body	The payload of the request or response message. Note: The <b>message.body</b> context variable is not supported with <b>getContext ()</b> function. Use the <b>getvariable ()</b> function instead.
message.headers.name	The value of the current named header of the message or of the current named header of the root part of a multipart message. The <b>name</b> segment is case-insensitive.
message.status.code	The HTTP status code of the response.
message.status.reason	The HTTP reason phrase of the response.
plan.name	The name of the plan.
plan.id	The unique identifier of the plan.
plan.version	The version number of the plan.
plan.rate-limit	The rate limit (the number of API calls per time interval) of the plan.
request.authorization	The parsed HTTP <b>authorization</b> header.
request.body	The payload from the incoming request.
request.content-type	Normalized content-type value.
request.date	A date object that represents approximately when the request was received by the Gateway.
request.headers.headername	The value of the original named header of the HTTP request, or the value of the current named header of the root part of a multipart request. The <b>headername</b> segment is case-insensitive.
request.parameters	You can obtain your incoming parameters from path and query parameters.
request.path	The path section of the <b>request.uri</b> that starts with the base path of the API, including the '/' character that begins the base path.
request.querystring	The request query string without the leading question mark.
request.search	The request query string with the leading question mark.
request.uri	The full HTTP request URI from the application.
request.verb	The HTTP verb of this request.
session.apiGateway	The gateway that receives the request.
session.apiGatewayName	The name of the API gateway as defined in the API Manager.
session.clientAddress	The address of the client that sent the request.
session.domainName	The name of the domain that the gateway belongs to.
session.globalTransactionID	The global transaction ID in the logs.
session.localAddress	The address of the gateway on the DataPower® Gateway.
session.timeStarted	The time that the gateway started to process the request.
session.transactionID	The transaction ID of the gateway request.
system.datetime	Returns a string that represents the current date and time in the system time zone of the gateway.
system.time	Returns a string that represents the current time in the system time zone of the gateway.
system.time.hour	Returns a number 0 - 23 inclusive, representing the hour of the current time in the system time zone of the gateway.
system.time.minute	Returns a number 0 - 59 inclusive representing the minute of the current time in the system time zone of the gateway.
system.time.seconds	Returns a number 0 - 59 inclusive representing the seconds of the current time in the system time zone of the gateway.
system.date	Returns a string that represents the current date in the system time zone of the gateway.
system.date.day-of-week	Returns a number 1 - 7 (Monday to Sunday) inclusive representing the day of the week in the system time zone of the gateway.
system.date.day-of-month	Returns a number 1 - 31 representing the day of the month in the system time zone of the gateway.
system.date.month	Returns a number 1 - 12 representing the month in the system time zone of the gateway.
system.date.year	Returns a four-digit number that represents the year in the system time zone of the gateway.
system.timezone	Returns a system time zone ISO 8601 identifier for the gateway, which might include a sign, a two-digit hour, and minutes. For example, <b>-04:00</b> .

## OAuth context variables - DataPower Gateway (v5 compatible)

The OAuth context variables described in this section apply only to the DataPower Gateway (v5 compatible). For details of the OAuth context variables that apply to the DataPower API Gateway, see [OAuth context variables](#).

Table 2. OAuth context variables (DataPower Gateway (v5 compatible)).

Note: Most OAuth context variables are available only when IBM API Connect is acting as the OAuth resource server. However, the `oauth.introspect` variables are also available when integrating with third party providers.

Name	Description
<code>oauth.access-token</code>	If the request is authenticated with OAuth, this variable contains the access token string.
<code>oauth.miscinfo</code>	This variable contains information explicitly included in the Authentication URL and Metadata URL headers. For more information, see <a href="#">Authenticate URL</a> .
<code>oauth.not-after</code>	If the request is authenticated with OAuth, this variable contains the date when the token expires.
<code>oauth.not-before</code>	If the request is authenticated with OAuth, this variable contains the date when the token was issued.
<code>oauth.resource-owner</code>	If the request is authenticated with OAuth, this variable contains the name of the resource owner.
<code>oauth.scope</code>	If the request is authenticated with OAuth, this variable contains the scope of this access token.
<code>oauth.introspect.active</code>	Always available to introspection. Boolean value.
<code>oauth.introspect.response</code>	Always available to introspection. Shows the complete current response payload. Example payload value: <code>{"active":true, "client_id", "xxx-xxx", "token_type", "bearer", "scope":"neon"}</code>
Other variables might be available from the third party, in the form of: <code>oauth.introspect.&lt;variable&gt;</code>	Decoding the previous example payload, the following variables are made available for further processing.  <code>oauth.introspect.client_id: xxx-xxx</code> <code>oauth.introspect.token_type: bearer</code> <code>oauth.introspect.scope: neon</code>

## Application certificate context variables

The following table describes context variables that are available when a certificate is used to verify access to an API, although these will vary depending on the signature mechanism that is being used; for more information, see the [Internet X.509 Public Key Infrastructure Certificate and CRL Profile specification](#).

Table 3. Application certificate context variables

Name	Description
<code>application.certificate.Base64</code>	Base64 format.
<code>application.certificate.fingerprint</code>	Fingerprint
<code>application.certificate.Version</code>	Version
<code>application.certificate.SerialNumber</code>	Serial number
<code>application.certificate.SignatureAlgorithm</code>	Signing algorithm
<code>application.certificate.Issuer</code>	The issuer of the certificate
<code>application.certificate.Subject</code>	Subject
<code>application.certificate.NotBefore</code>	Not valid before this date
<code>application.certificate.NotAfter</code>	Not valid after this date
<code>application.certificate.SubjectPublicKeyAlgorithm</code>	Algorithm for the subject public key
<code>application.certificate.SubjectPublicKeyBitLength</code>	Length for the subject public key
<code>application.certificate.KeyValue.type</code>	Various context variables that depend on the algorithm and key. The following are possible context variables: <ul style="list-style-type: none"> <li><code>application.certificate.KeyValue.RSAKeyValue.Modulus</code></li> <li><code>application.certificate.KeyValue.RSAKeyValue.Exponent</code></li> </ul>


## API activity logging context variables

If activity logging is enabled for an API, a `log` context variable is created; the `log` context variable contains the data relating to an API execution event. On completion of API execution, the `log` context variable is written to an API event record that is stored for subsequent access by API analytics. For details of the fields contained in the `log` context variable, see [API event record fields](#).

The way in which you enable and configure activity logging depends on the type of gateway you are using, as follows:

- If you are using the DataPower API Gateway, you configure activity logging in the API configuration settings; see [Activity logging with the DataPower API Gateway](#).
- If you are using the DataPower Gateway (v5 compatible), you configure activity logging by adding an [Activity Log](#) policy to the API assembly.

The activity logging configuration defines the default content of the `log` context variable, but you can modify it in an API assembly; for example:

- Add your own data values by using a [Set Variable](#) policy.
- Remove or redact data values by using a Redaction policy; see [Redaction - DataPower API Gateway](#) or [Redaction - DataPower Gateway \(v5 compatible\)](#).
-  Modify or replace the `log` context variable by using a [Log](#) policy.

## Related concepts

- [Variable references in API Connect](#)
- [The behavior of an assembly](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OAuth context variables

You can customize the OAuth security flow in a native OAuth provider by adding multiple OAuth policies to the assembly. Each OAuth policy takes input from context variables and writes output to context variables. By using these OAuth context variables you can manipulate the original request, and process the output from an OAuth policy.

Note: The OAuth context variables described here apply only to the DataPower® API Gateway. For details of the OAuth context variables that apply to the DataPower Gateway (v5 compatible), see [OAuth context variables - DataPower Gateway \(v5 compatible\)](#).

There are two main types of OAuth context variables:

- [Information variables](#), outputs of an OAuth policy.
- [Processing variables](#), processed during the execution of an OAuth policy.

## Information variables

These context variables are outputs of an OAuth policy. They are not used in the processing of the OAuth action.

### `oauth.result`

The result of the latest action; returns **SUCCESS** or **FAILURE**.

### `oauth.settings.variable_name`

Basic settings from the OAuth provider. The following context variables are available:

```
oauth.settings.allowed_scopes
oauth.settings.access_token_ttl
oauth.settings.authorization_code_ttl
oauth.settings.refresh_token_ttl
oauth.settings.refresh_token_limit
oauth.settings.maximum_consent_ttl
```

### `oauth.executed_components[0].variable_name`

The components that have executed in this transaction, with their result. If a failure occurs then the error and error description will be added. The following context variables are available:

```
oauth.executed_components[0].result
oauth.executed_components[0].type
oauth.executed_components[0].error_description
oauth.executed_components[0].error
```

The possible values for the `type` variable are as follows:

```
ValidateRequestComponent
GenerateAZCodeComponent
VerifyAZCodeComponent
VerifyRefreshTokenComponent
GenerateAccessTokenComponent
IntrospectTokenComponent
RevokeTokenComponent
CollectMetaDataComponent
```

The following example shows a corresponding JSON extract from the OAuth context:

```
"executed_components": [
  {
    "type": "ValidateRequestComponent",
    "result": "SUCCESS"
  },
  {
    "type": "GenerateAccessTokenComponent",
    "result": "SUCCESS"
  }
]
```

### `oauth.verified_access_token.variable_name`

The token that was verified in the security requirement for a native OAuth provider. The following context variables are available:

```
oauth.verified_access_token.access_token

oauth.verified_access_token.client_id
oauth.verified_access_token.consented_on
oauth.verified_access_token.consented_on_text
oauth.verified_access_token.grant_type
oauth.verified_access_token.misc_info
oauth.verified_access_token.not_after
oauth.verified_access_token.not_after_text
oauth.verified_access_token.not_before
oauth.verified_access_token.not_before_text
oauth.verified_access_token.one_time_use
oauth.verified_access_token.resource_owner
oauth.verified_access_token.scope
```

### `oauth.third_party.variable_name`

The token that was verified in the security requirement for a third party OAuth provider. The following context variables are available:

```
oauth.third_party.headers
oauth.third_party.response
```

#### `oauth.code.variable_name`

The code that was generated during the execution of a component of type `GenerateAZCodeComponent`. The following context variables are available:

```
oauth.code.client_id
oauth.code.code
oauth.code.redirect_uri
oauth.code.resource_owner
oauth.code.scope
```

#### `oauth.token.variable_name`

The token that was generated during the execution of a component of type `GenerateAccessTokenComponent`. The following context variables are available:

```
oauth.token.token_type
oauth.token.access_token
oauth.token.scope
oauth.token.expires_in
oauth.token.consented_on
oauth.token.redirect_uri
oauth.token.resource_owner
oauth.token.client_id
oauth.token.refresh_token
oauth.token.refresh_token_expires_in
oauth.token.refresh_token_count
```

#### `oauth.introspect.variable_name`

The token that was introspected during execution of a component of type `IntrospectTokenComponent`. The following context variables are available:

```
oauth.introspect.active
oauth.introspect.scope
oauth.introspect.client_id
oauth.introspect.resource_owner
oauth.introspect.token_type
oauth.introspect.grant_type
oauth.introspect.ttl
oauth.introspect.expires
oauth.introspect.expires_text
oauth.introspect.iat
oauth.introspect.not_before
oauth.introspect.not_before_text
oauth.introspect.consented_on
oauth.introspect.consented_on_text
oauth.introspect.one_time_use
```

## Processing variables

---

These context variables are processed during the execution of an OAuth policy in your API assembly. If they are set prior to any OAuth policy then the value overrides what was sent in the request. They can also be modified between OAuth policies to manipulate the next component that executes. The following context variables are available:

```
oauth.processing.assertion
oauth.processing.client_id
oauth.processing.client_secret
oauth.processing.grant_type
oauth.processing.redirect_uri
oauth.processing.scope
oauth.processing.response_type
oauth.processing.state
oauth.processing.resource_owner
oauth.processing.refresh_token
oauth.processing.code
oauth.processing.token
oauth.processing.token_type_hint
oauth.processing.nonce
oauth.processing.max_age
oauth.processing.oidc_values_requested
oauth.processing.id_token_requested
oauth.processing.oidc_signing_algorithm
oauth.processing.code_challenge
oauth.processing.code_challenge_method
oauth.processing.code_verifier

oauth.processing.metadata.access_token
oauth.processing.metadata.payload
oauth.processing.metadata.azcode_miscinfo

oauth.processing.verified_code.client_id
oauth.processing.verified_code.resource_owner
oauth.processing.verified_code.misc_info
oauth.processing.verified_code.scope
oauth.processing.verified_code.is_verified
oauth.processing.verified_code.nonce

oauth.processing.verified_refresh_token.client_id
oauth.processing.verified_refresh_token.resource_owner
oauth.processing.verified_refresh_token.misc_info
oauth.processing.verified_refresh_token.scope
oauth.processing.verified_refresh_token.refresh_token_count
oauth.processing.verified_refresh_token.is_verified
oauth.processing.verified_refresh_token.one_time_use
oauth.processing.verified_refresh_token.grant_type
```

The following example shows the OpenAPI source code for a `gateway-script` policy that executes before an OAuth policy in your API assembly and adds a custom scope to the request:

```
// Add another custom scope to the request
let scope = context.get("request.parameters.scope.values[0]");
if (scope)
  context.set("oauth.processing.scope", scope + " custom");
```

The following example shows the OpenAPI source code for a `gateway-script` policy that between OAuth policies in your assembly and modifies the scope depending on the resource owner:

```
// Check resource owner and modify the scope
let owner = context.get("oauth.processing.resource_owner");
let scope = context.get("oauth.processing.scope");

if (owner === 'admin') {
  context.set("oauth.processing.scope", scope + " admin");
} else {
  context.set("oauth.processing.scope", scope + " customer");
}
```

## Advanced scope context variables

The following context variables are used in OAuth advanced scope checking. They are populated by the values entered in the Endpoint and TLS Client Profile fields for the Application scope check and Owner scope check in the native OAuth provider configuration. Do not change these values as this might result in incorrect advanced scope operation.

For information on scope configuration, see [Configuring scopes for a native OAuth provider](#). For information on advanced scope checking, see [Scope](#).

```
oauth.advscope.app.url
oauth.advscope.app.tls-profile
oauth.advscope.own.url
oauth.advscope.own.tls-profile
```

## Custom error context variables

These context variables are used to stop OAuth processing and specify the respective error details. They can be used before OAuth processing takes place.

```
oauth.custom_error.status.code
oauth.custom_error.status.reason
oauth.custom_error.message
oauth.custom_error.description
```

## Related reference

- [Example - using multiple OAuth policies in an OAuth provider assembly](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API and Product definition template examples

You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition. This topic provides the default Handlebar templates used by `apic` to create Products and APIs as examples you can use to copy and customize for your own use.

## Default API definition template

The following example template is the default template that developer toolkit uses when you create an API definition. Copy this example template into your own template file (which must have the `.hbs` extension), then edit it for your purposes. Template variables are enclosed by double curly braces, for example `{{name}}`. For information on template variables, see [Template variables for API and Product definitions](#). For more information on Handlebars, see <https://handlebarsjs.com/>.

```
swagger: '2.0'

info:
  x-ibm-name: {{name}}
  title: {{title}}
  version: {{version}}

schemes:
  {{#if schemes}}
    {{each schemes}}
      - {{this}}
    {{/each}}
  {{else}}
    - https
  {{/if}}
host: {{hostname}}
basePath: {{basepath}}
```

```

consumes:
- application/json
produces:
- application/json

securityDefinitions:
  clientIdHeader:
    type: apiKey
    in: header
    name: X-IBM-Client-Id
  clientSecretHeader:
    in: "header"
    name: "X-IBM-Client-Secret"
    type: "apiKey"

security:
-
  clientIdHeader: []
  clientSecretHeader: []

x-ibm-configuration:
  testable: true
  enforced: true
  gateway: datapower-gateway
  cors:
    enabled: true
  catalogs:
    apic-dev:
      properties:
        runtime-url: ${TARGET_URL}
    sb:
      properties:
        runtime-url: 'http://localhost:4001'
  assembly:
    execute:
      - invoke:
          {{if targeturl}}
            target-url: {{targeturl}}
          {{else}}
            target-url: ${runtime-url}${request.path}
          {{/if}}

```

paths:

/users:

```

post:
  summary: Create a user
  description: Create a new user
  operationId: userCreate
  externalDocs:
    description: Blah
    url: http://host/docs-about-routes-post
  tags:
  - Users
  responses:
    '201':
      description: 'Success'
      schema:
        $ref: '#/definitions/User'
    default:
      description: 'Unexpected error'
      schema:
        $ref: '#/definitions/Error'

```

```

get:
  summary: User list
  description: Get a list of users
  operationId: userList
  externalDocs:
    description: Blah
    url: http://host/docs-about-routes-post
  tags:
  - Users
  responses:
    '200':
      description: 'Success'
      schema:
        $ref: '#/definitions/UserList'
    default:
      description: 'Unexpected error'
      schema:
        $ref: '#/definitions/Error'

```

/users/{user}:

```

get:
  summary: Retrieve the User
  description: Retrieve the User
  operationId: userGet
  tags:
  - Users
  parameters:
  - name: user
    in: path
    description: User id or name
    required: true

```



```

    type: string
  responses:
    '200':
      description: 'Success'
      schema:
        $ref: '#/definitions/User'
    default:
      description: 'Unexpected error'
      schema:
        $ref: '#/definitions/Error'

```

```

patch:
  summary: Update User
  description: Update User
  operationId: userUpdate
  tags:
    - Users
  parameters:
    - name: user
      in: path
      description: User id or name
      required: true
      type: string
    - name: payload
      in: body
      description: User to update
      required: true
      schema:
        $ref: '#/definitions/UserUpdate'
  responses:
    '200':
      description: 'Success'
      schema:
        $ref: '#/definitions/User'
    default:
      description: 'Unexpected error'
      schema:
        $ref: '#/definitions/Error'

```

```

delete:
  summary: Delete the User
  description: Delete the User
  operationId: userDelete
  tags:
    - Users
  parameters:
    - name: user
      in: path
      description: User id or name
      required: true
      type: string
  responses:
    '204':
      description: 'Successful delete'
    default:
      description: 'Unexpected error'
      schema:
        $ref: '#/definitions/Error'

```

#### definitions:

```

User:
  type: object
  additionalProperties: false

```

```

UserUpdate:
  type: object
  additionalProperties: false

```

```

UserList:
  type: object
  additionalProperties: false

```

```

Error:
  type: object
  additionalProperties: false
  properties:
    status:
      type: integer
    message:
      type:
        - string
        - array

```

```

tags:
  - name: Users
    description: Tags on all the user operations
    externalDocs:
      description: External information about Users
      url: http://host/url-of-my-entire-set-of-tag-docs-for-this-tag
  - name: Routes
    description: Tags on all the route operations
    externalDocs:
      description: External information about Routes
      url: http://host/url-of-my-entire-set-of-tag-docs-for-this-tag

```

## OAuth 2.0 API definition template

The following example template is the default template that developer toolkit uses when you create an OAuth 2.0 API definition with the command `apic create:api -template oauth`. Copy this example template into your own template file (which must have the `.hbs` extension), then edit it for your purposes. Template variables are enclosed by double curly braces, for example `{{name}}`. For information on template variables, see [Template variables for API and Product definitions](https://handlebarsjs.com/). For more information on Handlebars, see <https://handlebarsjs.com/>.

```
swagger: "2.0"

info:
  x-ibm-name: {{name}}
  title: {{title}}
  version: {{version}}

schemes:
  {{#if schemes}}
    {{#each schemes}}
      - {{this}}
    {{/each}}
  {{else}}
    - https
  {{/if}}
host: {{hostname}}
basePath: {{basepath}}

securityDefinitions:
  clientID:
    description: "application's client_id"
    in: "query"
    name: "client_id"
    type: "apiKey"

security:
- clientID: []

paths:
  /oauth2/authorize:
    get:
      produces:
        - text/html
      summary: endpoint for Authorization Code and Implicit grants
      description: description
      parameters:
        - name: response_type
          in: query
          description: request an authorization code or or access token (implicit)
          required: true
          type: string
          enum:
            - code
            - token
        - name: client_id
          in: query
          description: Application client ID
          required: true
          type: string
        - name: scope
          in: query
          description: Scope being requested
          type: string
          required: true
        - name: redirect_uri
          in: query
          type: string
          description: URI where user is redirected to after authorization
          required: false
        - name: state
          in: query
          type: string
          description: This string will be echoed back to application when user is redirected
          required: false
      responses:
        302:
          description: |
            Redirect to the clients redirect_uri containing one of the following
            - **authorization code** for Authorization code grant
            - **access token** for Implicit grant
            - **error** in case of errors, such as the user has denied the request
        200:
          description: An HTML form for authentication or authorization of this request.
      security:
        - clientID: []

    post:
      consumes:
        - application/x-www-form-urlencoded
      produces:
        - text/html
      summary: submit approval to authorization code or access token
      description: |
        Submit resource owners approval (or rejection) for the OAuth2 Server to issue an
        authorization code or access token to the application.
```

```

security:
- clientID: []
parameters:
- name: client_id
  in: formData
  description: application requesting the access code or token
  required: true
  type: string
- name: scope
  in: formData
  description: requested scope of this authorization
  required: true
  type: string
- name: resource-owner
  in: formData
  description: resource owners user name
  required: true
  type: string
- name: redirect_uri
  in: formData
  description: URI the application is requesting this code or token to be redirected to
  required: true
  type: string
- name: original-url
  in: formData
  description: URL of the original authorization request
  required: true
  type: string
- name: dp-state
  in: formData
  description: state information provided in the authorization form
  required: true
  type: string
- name: dp-data
  in: formData
  description: state information provided in the authorization form
  required: true
  type: string
#- name: response_type
# in: formData
# description:
# required: true
# type: string
responses:
200:
  description: Cool

```

/oauth2/token:

```

post:
  consumes:
  - application/x-www-form-urlencoded
  produces:
  - application/json
  summary: Request Access Tokens
  description: |
    This endpoint allows requesting an access token following one of the flows below:
    - Authorization Code (exchange code for access token)
    - Client Credentials (2-legged, there isnt resource owner information)
    - Resource Owner Password Credentials (2-legged, client provides resource owner name and password)
    - Refresh Token (exchange refresh token for a new access code)

```

The table below indicates the required parameters for each specific grant\_type options. Empty cells indicate a parameter is ignored for that specific grant type.

Client authentication:

- Confidential clients should authenticate using HTTP Basic Authentication. Alternatively, they may post their client\_id and client\_secret information as a formData parameter.
- Public clients should send their client\_id as formData parameter.

grant_type	code	client_credentials	password	refresh_token
client_id	required*	required*	required*	required*
client_secret	required*	required*	required*	required*
code	required			
redirect_uri	required			
username			required	
password			required	
scope		optional	optional	
refresh_token				required

The implicit grant requests, see /oauth2/authorize.

```

security: []
parameters:
- name: grant_type
  in: formData
  description: Type of grant
  type: string
  required: true
  enum:
  - authorization_code
  - password
  - client_credentials
  - refresh_token

```

```

- name: client_id
  in: formData
  description: Application client ID, can be provided in formData or using HTTP Basic Authentication
  required: false
  type: string
- name: client_secret
  in: formData
  description: Application secret, must be provided in formData or using HTTP Basic Authentication
  required: false
  type: string
- name: code
  in: formData
  description: Authorization code provided by the /oauth2/authorize endpoint
  required: false
  type: string
- name: redirect_uri
  in: formData
  description: required only if the redirect_uri parameter was included in the authorization request /oauth2/authorize;
their values MUST be identical.
  required: false
  type: string
- name: username
  in: formData
  type: string
  description: Resource owner username
  required: false
- name: password
  in: formData
  type: string
  description: Resource owner password
  required: false
- name: scope
  in: formData
  type: string
  description: Scope being requested
  required: false
- name: refresh_token
  in: formData
  type: string
  description: The refresh token that the client wants to exchange for a new access token (refresh_token grant_type)
  required: false
responses:
  200:
    description: json document containing token, etc.
    schema:
      $ref: "#/definitions/access_token_response"
  400:
    description: json document that may contain additional details about the failure

x-ibm-configuration:
  testable: true
  enforced: true
  phase: "realized"
  oauth2:
    client-type: public #or confidential
    scopes:
      scope1: Description 1
      scope2: Description 2
      scope3: Description 3
    grants:
      - application
      - password
      - accessCode
      - implicit

identity-extraction:
  type: default-form #If identity extraction is not there use this form.
  #type: basic
  #type: custom-form #Customer provided form (needs location)
  #type: redirect #Redirects user to authenticate somewhere else
  #custom-form:
  # url: https://example.com/authentication/form
  # tls-profile: tls-profile-1
  #redirect-url: https://example.com/external/form

authentication:
  x-ibm-authentication-url:
    url: https://example.com/auth/url
    tls-profile: tls-profile-4
  #x-ibm-authentication-registry: ldap-1

authorization:
  type: authenticated #If the authorization section is missing this is the default
  #type: default-form
  #type: custom-form
  #custom-form:
  # url: https://example.com/authorization/form
  # tls-profile: tls-profile-2

refresh-token:
  count: 2048 # If this section is missing default is 0
revocation:
  url: ""
  tls-profile: ""

definitions:

```

```

access_token_response:
  type: object
  additionalProperties: false
  required:
    - token_type
    - access_token
    - expires_in
  properties:
    token_type:
      enum:
        - bearer
    access_token:
      type: string
    expires_in:
      type: integer
    scope:
      type: string
    refresh_token:
      type: string

```

## Product definition template

The following example template is the default template that developer toolkit uses when you create a Product definition. Copy this example template into your own template file (which must have the `.hbs` extension), then edit it for your purposes. Template variables are enclosed by double curly braces, for example `{{name}}`. For information on template variables, see [Template variables for API and Product definitions](#). For more information on Handlebars, see <https://handlebarsjs.com/>.

```

product: '1.0.0'

info:
  name: {{name}}
  title: {{title}}
  version: {{version}}

{{#isEmpty apis}}
{{else}}
apis:
  {{/isEmpty}}
  {{#each apis}}
    '{{@key}}':
      $ref: {{this}}
  {{/each}}

visibility:
  view:
    type: public
  subscribe:
    type: authenticated

plans:
  default:
    title: Default Plan
    description: Default Plan
    approval: false
    rate-limit:
      value: 100/hour
    hard-limit: false

```

## Related tasks

- [Creating and using API and Product definitions templates](#)

## Related reference

- [Template variables for API and Product definitions](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Template variables for API and Product definitions

You can use template files when creating API and Product definitions. Template files are Handlebars templates containing variables of the form `{{variable-name}}` that are substituted with values when you create the API or Product definition.

## Product definition variables

The following table describes the Handlebars template variables that can be used in a product definition file. For more information on Handlebars, see <https://handlebarsjs.com/>. Product definition template files must have a `.hbs` filename extension.

Table 1. Product definition Handlebars template variables

Variable	Type	Description
{{apis}}	Array of string	The APIs to which the product definition refers. After substitution, the array values become the values of <code>apis.routes[n].\$ref</code> fields, for example:  <pre>apis:   'routes':     \$ref: apidef.yaml     ...</pre>
{{name}}	String	Value of <code>info.x-ibm-name</code> field.
{{title}}	String	Value of <code>info.title</code> field.
{{version}}	String	Value of <code>info.version</code> field.

## API definition variables

The following table describes the Handlebars template variables that can be used in an API definition file. For more information on Handlebars, see <https://handlebarsjs.com/>. API definition template files must have a `.hbs` filename extension.

Table 2. API definition Handlebars template variables

Variable	Type	Description
{{basepath}}	String	Base path on which the API is served, which is relative to the <code>host</code> .
{{definitions}}	<a href="#">OpenAPI definitions object</a> converted to a YAML string ("stringified").	For a LoopBack API, contains data types that can be consumed and produced by operations. These data types can be primitives, arrays or models.
{{definitionsObj}}	<a href="#">OpenAPI definitions object</a> converted to a YAML string ("stringified").	For a LoopBack API, contains data types that can be consumed and produced by operations. These data types can be primitives, arrays or models.
{{hostname}}	String	Value of the <code>host</code> field.
{{name}}	String	Value of <code>info.x-ibm-name</code> field.
{{paths}}	<a href="#">OpenAPI paths object</a>	For a LoopBack API, contains the relative paths to the individual endpoints. The path is appended to <code>{basePath}</code> to construct the full URL.
{{pathsObj}}	<a href="#">OpenAPI paths object</a>	For a LoopBack API, contains the relative paths to the individual endpoints. The path is appended to <code>{basePath}</code> to construct the full URL.
{{schemes}}	Array of string	Transfer protocol of the API. Values must be one of: <code>"http"</code> , <code>"https"</code> , <code>"ws"</code> , or <code>"wss"</code> .
{{targeturl}}	String	Value of <code>x-ibm-configuration.assembly.execute[invoke]</code> . Default is <code>\$(runtime-url)\$(request.path)</code> .
{{title}}	String	Value of <code>info.title</code> field.
{{version}}	String	Value of <code>info.version</code> field.

## Related tasks

- [Creating and using API and Product definitions templates](#)

## Related reference

- [API and Product definition template examples](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Managing your APIs

You manage your APIs by using the API Manager user interface of IBM® API Connect. You can also analyze your API usage by using the analytics that are provided and socialize your APIs in a developer portal.

## Why use APIs?

Whether you are a business user, an IT user, or an application developer, APIs are increasingly important to your business. You can use an API to publicize your company. The assets, data, or services of your company can be provided to external application developers to expand your enterprise and open new markets.

The API Manager UI provides a solution for companies to manage APIs for private internal APIs, and public external APIs. This on-premises offering provides the capabilities that are required so that you can externalize and manage your services as REST or SOAP APIs.

## What do you need to know?

Depending on your role, you can complete different tasks relating to managing your API Catalogs and these are outlined in the documentation. Each task is covered in the order that they are executed.

Each task introduces new features of the API Manager UI as they become relevant to the Catalog that you are building.

For security reasons, your session times out after a period of inactivity.

Before you start, check that the browser you are using is supported and meets the required minimum levels; for details, open the [Detailed system requirements for a specific product](#) page, search for the IBM API Connect product, then select the required offering and version.

The ways in which you can manage your APIs are described in the following subtopics:

- **[Activating your API Manager user account](#)**  
Before you can access the API Manager user interface of IBM API Connect, you must activate the user account that your Cloud Manager administrator invited you to join.
- **[Accessing the API Manager user interface](#)**  
The IBM API Connect API Manager user interface provides a range of options for working with your APIs and Products, and managing security.
- **[\[Technical Preview\] Searching for items in API Manager](#)**  
Use the search feature in IBM API Connect API Manager to easily locate items such as APIs, Catalogs, applications, and subscriptions.
- **[Working with Catalogs](#)**  
Products must be staged to a Catalog and then published; application developers are able to access the APIs in a Product by being members of Consumer organizations to which the Product is made available. In IBM API Connect, you can create multiple Catalogs. Catalogs are useful for separating Products and APIs for testing before you make them available to Consumer organizations. The syndication feature in API Connect means that you can also publish a Product to a Space in a Catalog.
- **[Using syndication in API Connect](#)**  
With the IBM API Connect syndication feature, you can partition your Catalogs into *Spaces*. Each Space is used by a different API provider development team and has its own set of management capabilities relating specifically to the APIs that the associated team publishes to that Space, enabling each team to manage their APIs independently.
- **[Administering Consumer organizations](#)**  
Manage the Consumer organizations that access your APIs and Plans when their users sign up to use the IBM API Connect Developer Portal.
- **[Security and authentication](#)**  
In API Manager, you can use TLS profiles to secure the transmission of data between the management server and other API Connect subsystems and external services, and also configure user registries to securely authenticate your Catalogs and APIs.
- **[Working with Products in the API Manager](#)**  
In IBM API Connect, Plans and APIs are grouped together in Products.
- **[API Analytics](#)**  
You can use IBM API Connect to filter, sort, and aggregate your API event data. You can present the results within correlated charts, tables, and maps to help you manage service levels, set quotas, establish controls, set up security policies, manage communities, and analyze trends.
- **[Administering user access](#)**  
If you have permission to administer users in IBM API Connect, you can add and delete users. After users are removed from an organization through deletion, the user account remains in API Manager.
- **[Changing your API Manager password](#)**  
You can change your API Manager password in IBM API Connect.
- **[Resolving login problems by increasing HTTP header size](#)**  
You can resolve login problems for the API Manager UI by increasing the maximum HTTP client header size.
- **[Reference](#)**  
Reference information for the API Manager component in IBM API Connect.
- **[API Manager tutorials](#)**  
Tutorials for using the API Manager user interface in IBM API Connect.

---

## Related information

- [IBM API Connect overview](#)
- [Using the Developer Portal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Activating your API Manager user account

Before you can access the API Manager user interface of IBM® API Connect, you must activate the user account that your Cloud Manager administrator invited you to join.

---

### Before you begin

The Cloud Manager administrator invited you to join API Connect .

---

### Procedure

1. Complete the following steps to create your API Manager account:

If the Identity Provider uses LDAP

An invitation email with an activation link is sent. Click the activation link, or paste it in a browser, to log in directly with your user credentials. Upon authentication, the API Connect user record is updated from the backend identity provider.

If the Identity Provider uses a local registry

An invitation email with an activation link is sent. Click the activation link or paste it in a browser. The activation link takes you to a sign up page where you enter your first name, last name and password. Passwords must have a minimum of 8 characters and contain characters from at least three of the four following categories:

- Uppercase letters
- Lowercase letters

- Numbers
- Special characters (for example: ! # \$ %)

Note:

- The email address that you enter on the sign up page must match the email address to which the invitation email was sent, otherwise the account activation fails.
- If you previously had an account that the administrator removed, and they are inviting you again, you must re-activate your account by using the Sign In option on the page, **not** by completing the registration form and using the Sign Up option; attempting to re-register will fail.

If the Identity Provider uses an authentication URL

An invitation email with an activation link is sent. Click the activation link or paste it in a browser. The activation link takes you to a sign up page where you enter your credentials, which then become your user name and password. Upon authentication, the API Connect user record is updated from the backend identity provider.

If multiple user registries are available for selection on the sign up page, then make sure that the correct registry for your API Manager account is selected. You might need to ask your administrator which user registry is appropriate for your account.

For information on configuring user registries and making them available for API Manager login, see [User registries overview](#) and [Selecting user registries for Cloud Manager and API Manager](#).

2. Click Sign up to complete your registration, then click Sign in to open the API Manager login page.

---

## Results

You have created your API Connect account.

---

## What to do next

Log in to the API Manager user interface and, depending on your role, start to work with Catalogs, Consumer organizations, APIs and Products. To access the API Manager login page in the future, use the following URL:

**`https://host/manager`**

where *host* is the fully qualified host name or IP address of the Management server.

---

## Related concepts

- [Working with Products](#)

---

## Related tasks

- [Creating and configuring Catalogs](#)
- [Administering Consumer organizations](#)

---

## Related reference

- [Accessing the API Manager user interface](#)

---

## Related information

- [Developing your APIs and applications](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Accessing the API Manager user interface

The IBM® API Connect API Manager user interface provides a range of options for working with your APIs and Products, and managing security.

To log in to the API Manager user interface, complete the following steps:

1. In a web browser, enter the following URL:

**`https://host/manager`**

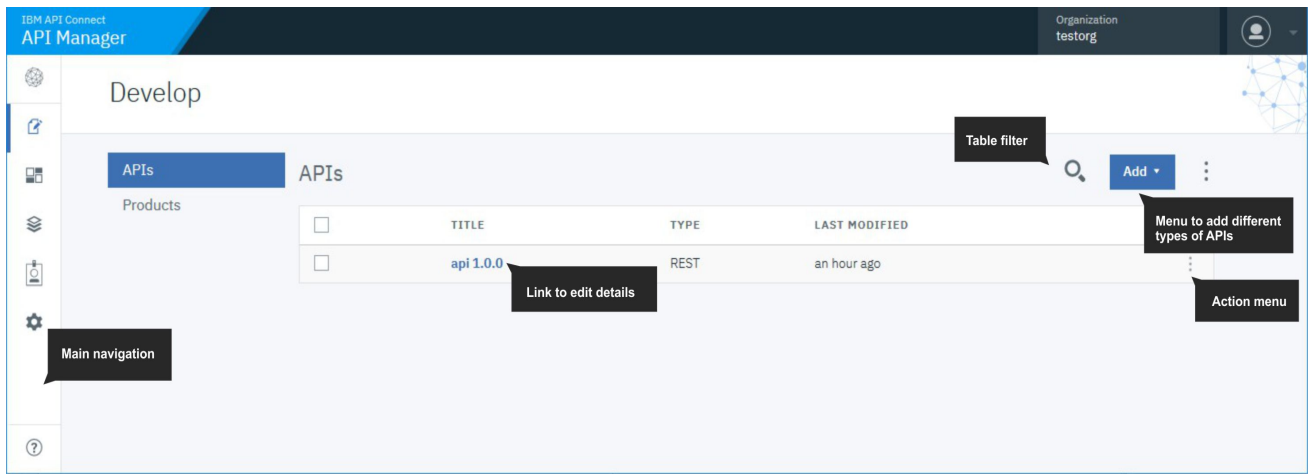
where *host* is the fully qualified host name or IP address of the Management server.

2. If your cloud contains multiple user registries, then select the user registry that contains the login credentials for your API Manager account. You might need to ask your administrator which user registry is appropriate for your account.

For information on configuring user registries and making them available for API Manager login, see [User registries overview](#) and [Selecting user registries for Cloud Manager and API Manager](#).

3. Enter your user name and password, then click Sign in. The API Manager user interface opens:





Depending on your role, you can now start to work with Catalogs, Consumer organizations, APIs and Products.

## Related concepts

- [Working with Products in the API Manager](#)
- [Security and authentication](#)

## Related tasks

- [Administering Consumer organizations](#)
- [Administering user access](#)

## Related reference

- [API Connect user roles](#)


## Related information

- [Managing your APIs](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## [Technical Preview] Searching for items in API Manager

Use the search feature in IBM® API Connect API Manager to easily locate items such as APIs, Catalogs, applications, and subscriptions.

Search is available from any page in API Manager and can be accessed by typing in the search field in the page banner, or by clicking  Search in the navigation list.

## This feature is provided as a Technical Preview

As such, the feature might not be fully functional in your environment and support is limited. In addition, the feature's appearance and functionality are subject to change both in this release (V2018) and in later releases of API Connect. Review the following disclaimer before enabling the Search technical preview:

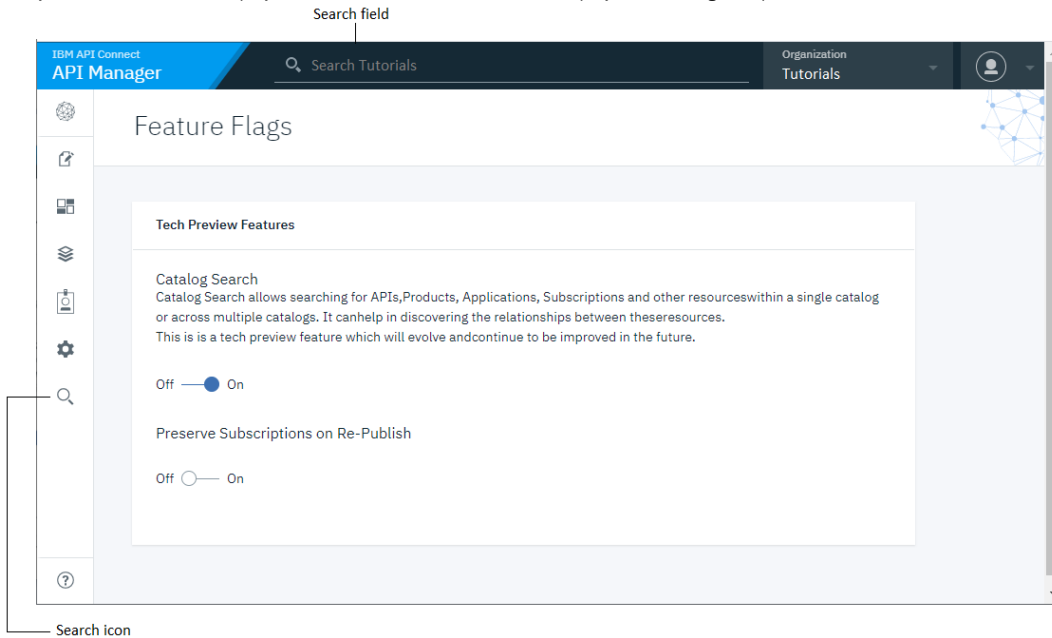
Notice: Technology Preview Code (TPC) may be included or distributed with the Program or updates to it but are not part of the Program. TPC is licensed under the same terms as the Program, except as provided below. TPC will be identified as such in the Notices File (or in an updated Notices File accompanying the updates). Some or all of the TPC may not be made generally available by IBM as or in a product. Licensee is permitted to use TPC only for internal use for evaluation purposes and not for use in a production environment. The Notices File may limit this evaluation use to an evaluation period. If so, at the end of such evaluation period Licensee must cease using and uninstall the TPC. IBM provides the TPC without obligation of support and "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF TITLE, NON-INFRINGEMENT OR NON-INTERFERENCE AND ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## Enabling Search


Enable Search for your API Connect user account by completing the following steps:

1. Log in to API Manager.
2. In the browser's address bar, append **/ff** to the URL and press Enter to display the Feature Flags page.  
For example: <https://example.com/my-org/ff>

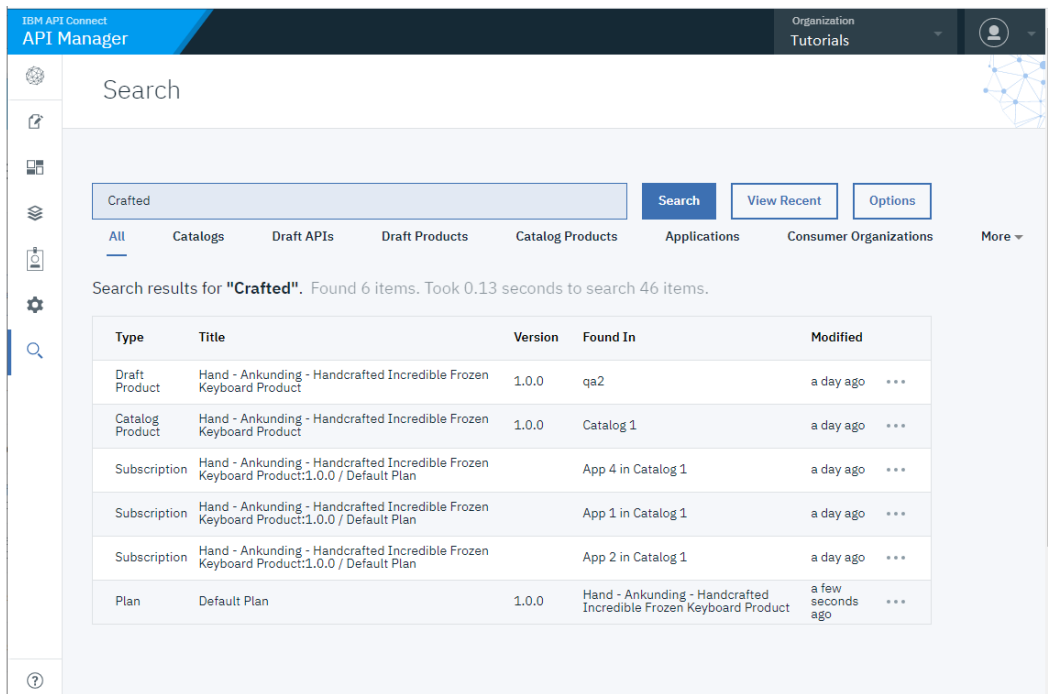
3. On the Feature Flags page, click the Off - On toggle for the "Catalog Search" option to set it to On.
4. When the "Enable Catalog Search" message displays, click Confirm.
5. Verify that the Search field displays in the banner and the Search icon displays in the navigation panel.



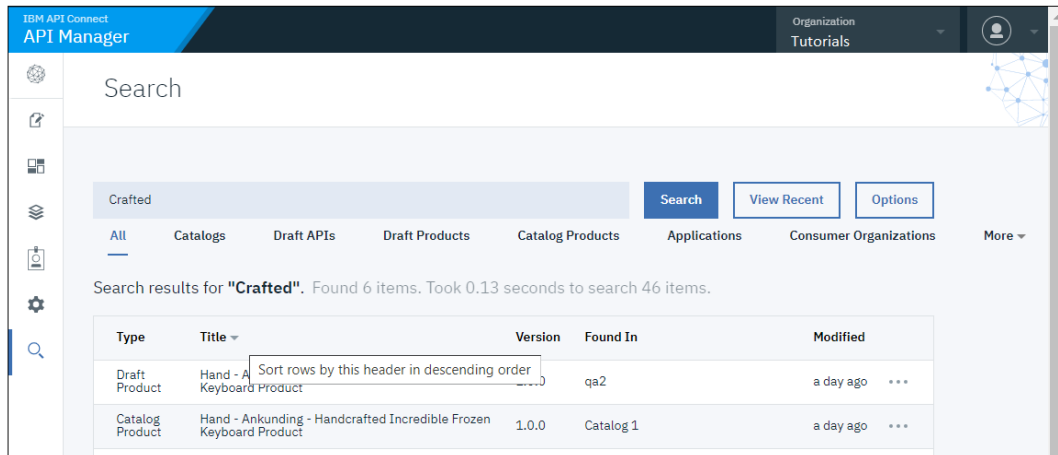
## Using Search

The Search feature is available on all pages of the API Manager. To start searching, just type a string into the Search field and press Enter. Whenever you press Enter in the Search field, the Search page displays with results and additional options. If you press Enter in an empty Search field, or click  Search in the navigation list, then the Search page displays a list all items. If you type a string before pressing Enter, the page displays the results from the corresponding search.

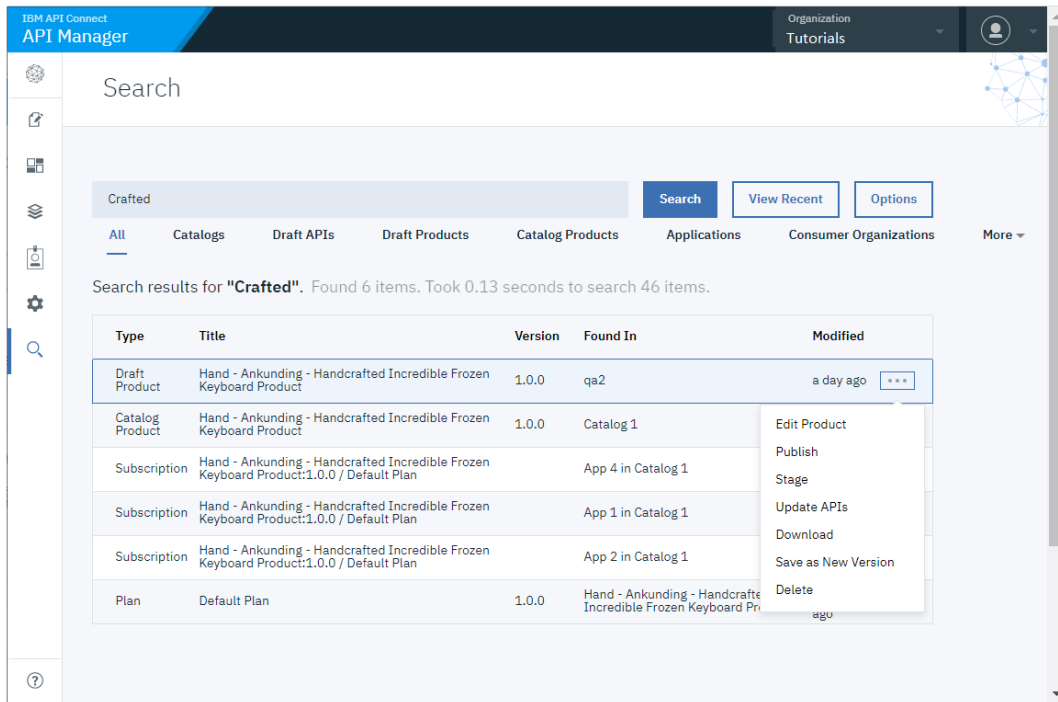
Search results include all items that contain the search string in a text field. For example, suppose you searched for "Crafted". The results might look like the following image, with items that don't match the search string exactly, but do include the string in a text field such as a title or a name:



By default, the search returns all types of items that contain the string. Sort results on a particular column by clicking the column header.

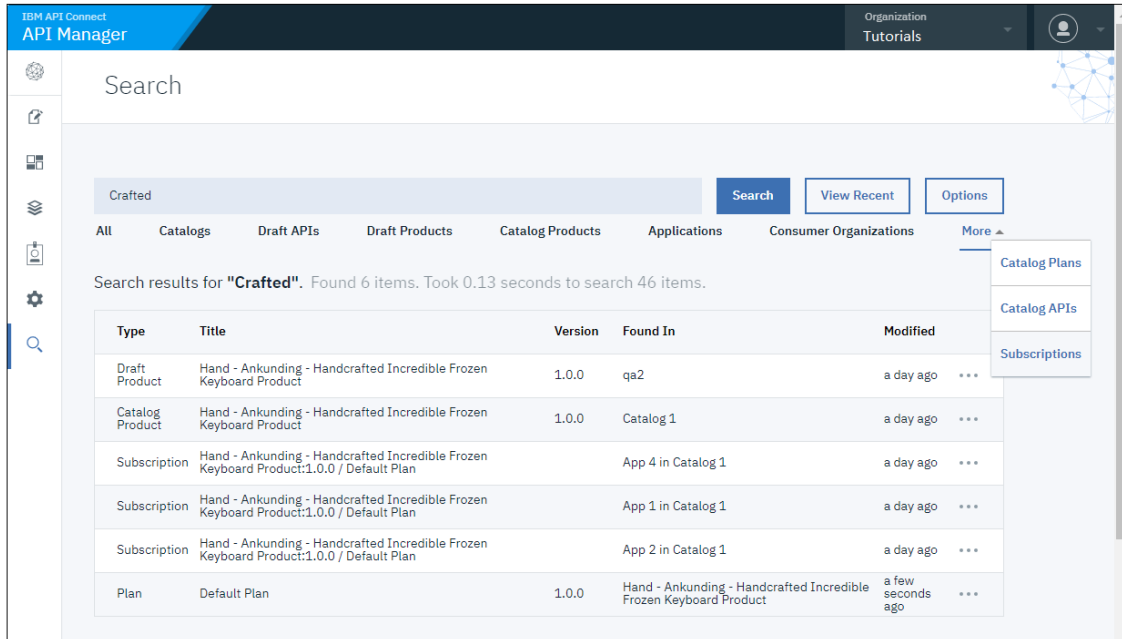


When you find an item you want to work with, click the ... actions menu next to the item's "Modified" date to see what you can do. The list of possible actions depends on the item's type.

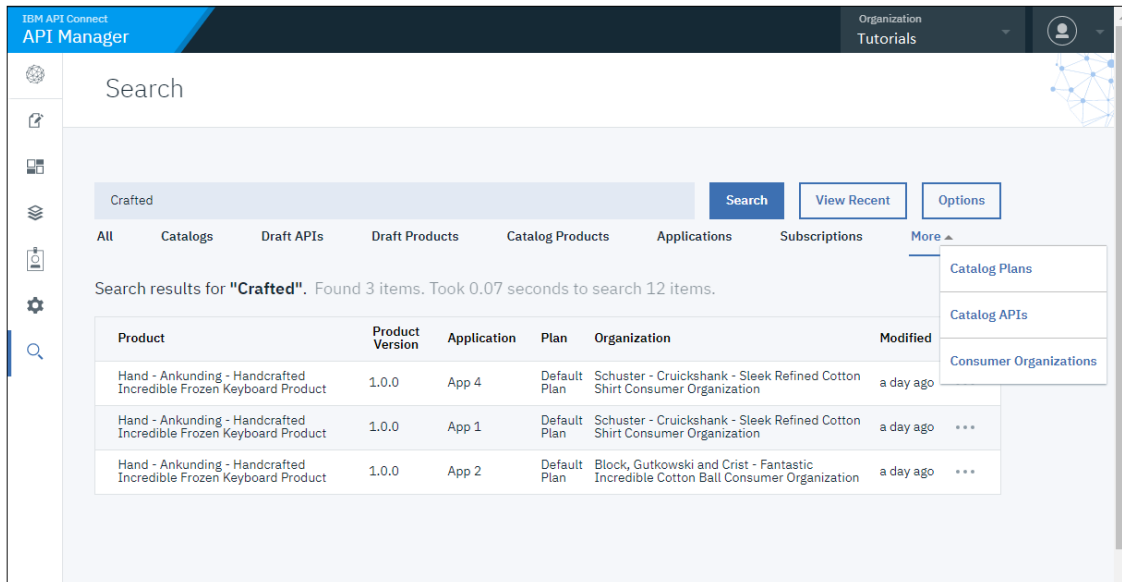


You can limit the search results by selecting an item type from the list that displays before the results. Click More to view additional types. If you select a type from the More menu, it replaces a type in the list.

In the following example, the results of the search for "Crafted" are filtered with the Subscriptions type.



Notice that "Subscriptions" now displays in the default list, and that "Consumer Organizations" moved to More menu.



If you want to refine the search further, click Options. On the Options panel, click Catalogs to Search and select which catalogs to search (the default is all catalogs). You can select up to 5 catalogs for a limited search.

You can also choose whether only exact matches (the complete string) are included in results, or items containing similar strings are also included.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Working with Catalogs

Products must be staged to a Catalog and then published; application developers are able to access the APIs in a Product by being members of Consumer organizations to which the Product is made available. In IBM API Connect, you can create multiple Catalogs. Catalogs are useful for separating Products and APIs for testing before you make them available to Consumer organizations. The syndication feature in API Connect means that you can also publish a Product to a Space in a Catalog.

A Catalog is a staging target, and behaves as a logical partition of the gateway and the Developer Portal. The URL for API calls and the Developer Portal are specific to a particular Catalog. In a typical configuration, an API provider organization uses a development Catalog for testing APIs under development and a production Catalog for hosting APIs that are ready for full use. A common approach is to have a development cloud with a development Catalog, a few test Catalogs and a production cloud that might have its own test Catalog.

You can use a *Space* to partition a Catalog so multiple teams can manage Products and APIs independently in a single Catalog. A Space is conceptually like a sub-catalog, except that Products and APIs in all Spaces in a given Catalog are published to the same developer portal. For more information about Spaces, see [Using syndication in API Connect](#).

Note:

Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a Developer Portal user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in to the Developer Portal: **A user already exists with this email address.**

For more information on using the Developer Portal, see [Developer Portal: Socialize your APIs](#).

- **Creating and configuring Catalogs**  
These instructions describe how to create and configure Catalogs in IBM API Connect.
- **Configuring sub paths for Developer Portal sites**  
You can increase the specificity of a IBM API Connect Developer Portal site URL by using sub paths.
- **Managing Catalog membership**  
You manage Catalog membership by adding new users to the Catalog and assigning roles to the users.
- **Importing a user-defined policy into a Catalog**  
You can make your user-defined policy available to API developers by importing it into an IBM API Connect Catalog, or a Space in a Catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating and configuring Catalogs

These instructions describe how to create and configure Catalogs in IBM® API Connect.

### Before you begin

You must possess Catalog create permissions to complete this task. For more information about permissions, see [API Connect user roles](#).



Note:

As the user who creates the Catalog, you are automatically the Catalog owner and have all Catalog permissions.


Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a Developer Portal user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in to the Developer Portal: **A user already exists with this email address.**

### Procedure

To create and configure your Catalog, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage.
2. Create the Catalog. You can create the Catalog in either of the following ways:
  - Specify the owner, then configure the Catalog yourself.
  - Send an invitation email, with an activation link, to the intended Catalog owner. The Catalog owner configures the Catalog after activating it.To specify the owner and configure the Catalog yourself, complete the following steps:
  - a. Click **Add > Create catalog**.
  - b. In the **Catalog Owner** field, select the owner of the Catalog; by default, you are the owner. You can specify any user who is a member of the provider organization.
  - c. Continue from step [3](#) to configure the Catalog.To invite the Catalog owner, complete the following steps:
  - a. Click **Add > Invite catalog owner**.
  - b. Enter the email address of the Catalog owner, then click **Invite**.On receipt of the invitation email, the Catalog owner should complete the following steps.
  - a. Open the activation link, complete the sign up form, and sign in to the API Manager UI.
  - b. Click **Manage Catalogs**, click the Catalog name that was provided on the sign up form, then click  **Settings**.
  - c. Continue from step [6](#) to configure the Catalog.
3. Enter the Catalog Title. A Name is entered automatically and cannot be changed.

Note: The value in the Name field is a single string that is used to identify the Catalog in developer toolkit CLI commands. The Title is used for display; you can change the title subsequently but the name does not change.

To view the CLI commands to manage Catalogs, see [apic catalogs](#).
4. Click **Create**.  
Your new Catalog is displayed on the **Manage** page.
5. To continue to configure your Catalog, select the Catalog on the **Manage** page, then click  **Settings**.
6. To configure the general settings for the Catalog, click **Overview**, then proceed with the following steps:
  - a. By default, the new Catalog is a development Catalog. To use the Catalog in production, set the **Production Mode** slider control to the **On** position, then click **Confirm**.  
In a development Catalog, all publish operations are forced, and any conflicts are resolved automatically. If you republish a previously published Product version it is overwritten without warning, allowing you, for example, to repeatedly republish the same Product version during testing. If Spaces are enabled in a development Catalog and you republish a previously published Product version to a different Space, it is removed from the previous Space.

A development Catalog behaves the same as any other Catalog with regard to requiring approval for staging and publishing actions, if the Catalog has been configured to require approval.

Note: In a production Catalog, the following conditions apply:

- You will be prevented from publishing a Product to the Catalog if there is already a Product in the Catalog with the same name and version, you must create a new version of the Product for publication; see [Creating a new version of your Product](#). If Spaces are enabled in a production Catalog, you cannot publish a Product with the same name and version to more than one Space in the Catalog.
- If this new Product contains one or more modified APIs, you **must** create new versions of these APIs for inclusion in the Product; see [Creating a new version of an API definition](#). If the Product contains a modified API and a published API of the same name and version already exists, the changes will **not** be published.

b. To enable the partition of this Catalog into Spaces, set the Spaces slider control to the On position, then click Confirm.

For more information, see [Using syndication in IBM API Connect](#).

7. To configure the gateway service settings for the Catalog, click Gateway Services, then proceed with the following steps:

Note: If Spaces are enabled in the Catalog then you configure the gateway settings for each Space separately, **not** for the Catalog as a whole. For more information about enabling and managing Spaces, see [Using syndication in API Connect](#).

a. Click Edit.

The Enable Gateway Services page opens.

b. Select the gateway services that you want to use with this Catalog.

Gateway services are configured by the using the Cloud Manager UI. For details, see [Registering a gateway service](#). If any Catalog default gateway services have been configured, they will already be selected when you create a new Catalog; see [Configuring default gateway services for Catalogs](#).

The TYPE column indicates the gateway type for each gateway service listed, DataPower® Gateway (v5 compatible) or DataPower API Gateway. For more information, see [API Connect gateway types](#).

c. Click Save to save your changes.

You publish APIs by adding them to a Product and then publishing the Product to a Catalog. To be able to publish Products to a Catalog, the Catalog must be assigned at least one gateway service so that the APIs in the Product are available to be called at a gateway service endpoint. You can assign more than one gateway service to a Catalog, and they can be of mixed types, DataPower API Gateway and DataPower Gateway (v5 compatible).

A Product is gateway type specific, either DataPower API Gateway or DataPower Gateway (v5 compatible). By default, when you publish a Product to a Catalog, it is published to all the assigned gateway services of the same gateway type as the Product, but you can choose to publish the Product only to selected gateway services, of that type, at publish time; see [Publishing a draft Product](#) for more information. The APIs in the Product will be available to be called at all the specified gateway service endpoints.

If you select two or more Gateway services then, for analytics data to be available in the Developer Portal for this Catalog, the Gateway services must all be associated with the **same** analytics service. If the set of associated analytics services for the selected gateway services includes two or more different analytics services then the Developer Portal does **not** display analytics data. An analytics service is associated with a gateway service in the Cloud Manager user interface; see [Associating an analytics service with a gateway service](#).

Note: You can also configure the gateway service settings for the Catalog by using the developer toolkit CLI; for details, see [apic configured-gateway-services](#).

8. To specify the Product lifecycle state changes for which you want to enforce approval, complete the following steps:

a. Click Lifecycle Approvals in the Catalog settings navigation pane, then click Edit.

b. Select the required state changes, then click Save when done.

For example, if you select Publish and leave the other check boxes cleared, approval is required when anyone attempts to publish a Product, but no approval is required for any of the other lifecycle state changes.

Note: Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.

c. To allow the originator of a Product lifecycle change to approve it themselves, move the Task self approval slider control to the On position.


Note: In a development Catalog, the originator of a Product lifecycle change can **always** approve it themselves.

9. Click Save when done.

10. To configure the permissions that each role in API Manager has, complete the following steps:

a. Click Roles in the Catalog settings navigation pane.

To see the current permissions for a role, expand the role.

b. Click the options icon  alongside the role that you want to work with, then click Edit.

c. Select or clear the permissions check boxes as required.

To ensure a role can manage user-defined policies, select Manage for the Settings permission.

To grant a role the permission to approve a specific lifecycle state change, select the state change in the Product-Approval section.


Note: You cannot remove a permission that has been assigned to the role at the provider organization level. However, you can add permissions that haven't been assigned at the provider organization level.

d. Click Save when done.

11. To configure the default role permissions for consumer organizations that are created in this Catalog, complete the following steps:

a. Click Role Defaults in the Catalog settings navigation pane.

To see the current default permissions for a role, expand the role.

b. Click the options icon  alongside the role that you want to work with, then click Edit.

c. Select or clear the permissions check boxes as required.

d. Click Save when done.


e. To create a new role and assign default permissions to it, click Add, name the role and assign permissions, then click Save.

12. To manage the onboarding of application developers into this Catalog, complete the following steps:

a. Click Onboarding in the Catalog settings navigation pane.

b. To select the registries that are used to authenticate users of the Developer Portal associated with this Catalog, click Edit in the Catalog User Registries section, select the required registries, then click Save.

For details on how to configure a user registry, see [Authenticating by using your enterprise user registry](#).

c. To set a default registry, click the options icon  alongside the required registry, then select Set default.


When an application developer signs up to the Developer Portal, the specified registry is used by default, with the option to select another registry.

d. To allow application developers to complete their own sign-up process, move the Self service onboarding slider control to the On position.

If this option is disabled, an application developer cannot sign-up without an invitation from a consumer organization owner.

e. To configure the timeout for activation links that are sent in email invitations to application developers, click Edit in the Invitation Timeout section, specify the timeout length, then click Save.

13. To specify the user registries that are used to secure access to APIs in this Catalog, and to add the user registry to the Sandbox Catalog, complete the following steps:

- a. Select  Settings from the navigation panel, and then select API User Registries.
- b. Click Edit and select the user registries you want to use.  
For details on how to configure a user registry, see [Authenticating by using your enterprise user registry](#).
- c. Click Save when done.

Note:

- Only LDAP and Authentication URL registries can be used to secure access to APIs; therefore, only registries of these types are listed and available for selection.
- If Spaces are enabled in the Catalog then you specify the user registries for the Spaces, **not** for the Catalog as a whole. If you specify a user registry in one Space in a Catalog, it is deployed to the gateway services across all Spaces in that Catalog.  
For more information about enabling and managing Spaces, see [Using syndication in API Connect](#)

14. To specify the OAuth Providers that can be used to secure access to APIs in this Catalog, complete the following steps:

Note:

- If you want to specify an OAuth provider for a Catalog, ensure that the Sandbox Catalog is configured to use the same OAuth provider
- Any resources that the OAuth provider references, such as API user registries or TLS client profiles, must also be enabled in the Catalog
- If Spaces are enabled in the Catalog then you specify the OAuth providers for the Spaces, **not** for the Catalog as a whole. If you specify an OAuth provider in one Space in a Catalog, it is deployed to the gateway services across all Spaces in that Catalog.  
For more information about enabling and managing Spaces, see [Using syndication in API Connect](#).

- a. Click OAuth Providers in the Catalog settings navigation pane.
- b. Click Edit, then select the OAuth providers that you want to use.  
To configure an OAuth provider, see either [Configuring a native OAuth provider](#) using Cloud Manager or [Configuring a native OAuth provider](#) using API Manager.
- c. Click Save when done.

15. Optional: Configure vanity API endpoints for your Catalog.

For any API, there are two possible endpoints to consider:

- The gateway endpoint at which the API is invoked.
- The endpoint that is visible to the consumer in the Developer Portal.

If you do not configure vanity endpoints, these two endpoints are the same, and point to the gateway endpoint at which an API is invoked.

To have an endpoint visible to the consumer that is different to the gateway endpoint, you configure a vanity endpoint. A vanity endpoint represents the endpoint by which an API is known externally; that is, the endpoint that is published to the Developer Portal and is used by an application developer to invoke the API.

For any enforced API in API Connect, the gateway endpoints formats for the API are as follows:

- If the OpenAPI definition has no **host** field, the API endpoint has the following format:  

```
https://gateway_service_host/provider_organization/catalog/basepath
```
- If the OpenAPI definition has a **host** field (for example, **petstore.com**), the host is appended to the path after the *provider\_organization/catalog* segments, and the API endpoint has the following format:  

```
https://gateway_service_host/provider_organization/catalog/host_field_value/basepath
```

Note: This is a change in behavior from IBM API Connect Version 5.0, where the **host** field is not included in the API endpoint.

where:

- *gateway\_service\_host* is the host name of the gateway service on which the API is running.
- *provider\_organization* is the value of the **name** field of the provider organization that contains the Catalog in which the API is published.
- *catalog* is the value of the **name** field of the Catalog in which the API is published.
- *basepath* is the value of the **basepath** field in the OpenAPI definition for the API.
- *host\_field\_value* is the value of the host field in the OpenAPI definition for the API.

Note:

- IBM API Connect Version 5.0 provides a feature known as *host to catalog mapping*, whereby an API can be invoked without having to include the provider organization or Catalog name in the URL path; this feature is not supported in IBM API Connect Version 2018.
- IBM API Connect Version 5.0 provides a Default setting; calls to APIs that are published to the default Catalog do not need to include the Catalog name. This feature is not supported in IBM API Connect Version 2018.

To configure vanity API endpoints, so that you can publish endpoints that are different to these gateway endpoints, complete the following steps:

- a. Click API Endpoints in the Catalog settings navigation pane. The Vanity API Endpoint page opens.
- b. Click Edit.
- c. To display the current API base endpoint settings, select Display vanity endpoint.
- d. Select the required Preference setting. Choose one of the following settings:
  - Catalog priority: Any host field defined in the OpenAPI definition of the API is ignored and all API endpoints always point to the endpoints that you define here. For example, suppose you define a vanity API endpoint, with Catalog priority preference, to be `https://prod.acme.com/`. Then the API endpoint will be `https://prod.acme.com/basepath`
  - API priority: The host field defined in the OpenAPI definition for the API takes precedence. For example, suppose you define a vanity API endpoint, with API priority preference, to be `https://api.acme.com/`. Then the following rules determine the endpoint that is used when an API is called:
    - If the OpenAPI definition has a host field, `test.acme.com` for example, then that value is used to determine the API endpoint. For example: `https://test.acme.com/basepath`.
    - If the OpenAPI definition has no host field, the API endpoint will be `https://api.acme.com/basepath`, as determined by the vanity API endpoint setting.
    - If no vanity API endpoints are defined, then the default gateway endpoints are used to determine the API endpoint.
- e. Supply one or more endpoints, as follows:
  - If you selected Catalog priority, click Add to enter one or more endpoints URLs, and an optional summary for each. Any of the endpoint URLs can be used to invoke an API.
  - If you selected API priority, enter the endpoint URL, and an optional summary.

Note: After configuring your Catalog to support your vanity URL, you must configure your external network to map the vanity endpoints to the corresponding gateway endpoints. This typically includes the following:

- A DNS mapping to ensure that the vanity host resolves to the gateway.

- Additional URL routing for the API as required.

For example, if your gateway has a default domain name of `apigw.dc.zone.mycompany.com` and an IP address of `29.12.141.150`, then the generic DNS configuration might look like:

```
api.acme.com. CNAME apigw.dc.zone.mycompany.com.
```

or

```
api.acme.com. A 29.12.141.150
```

Consult your network administrator and the DNS provider's documentation to do this configuration.

- To configure the TLS client profiles that are used with this Catalog, complete the following steps:
 

Note: If Spaces are enabled in the Catalog then you configure the TLS client profiles for the Spaces, **not** for the Catalog as a whole. If you configure a TLS client profile in one Space in a Catalog, it is deployed to the gateway services across all Spaces in that Catalog.

For more information about enabling and managing Spaces, see [Using syndication in API Connect](#).

- Click TLS Client Profiles in the Catalog settings navigation pane.
- Click Edit, then select the TLS client profiles that you want to use with this Catalog.
 

For details on how to create a TLS client profile, see [Creating a TLS client profile](#).
- Click Save when done.

- To configure the Developer Portal, complete the following steps:
  - Click Portal in the Catalog settings navigation pane, then click Create.
  - Select the portal service that you want to use with this Catalog.
 

Portal services are configured by using the Cloud Manager UI. For details, see [Registering a portal service](#).
  - Optionally, override the default generated portal URL.
 

The URL must have the following format:

```
https://host_name.portal.com
```

where `host_name.portal.com` must resolve to the IP address of the Developer Portal machine.  
For example:

```
https://myhost.mydomain.com
```


Note:

- The URL must be unique across all Catalogs.
- To implement sub paths into your site URL, see [Sub paths for Developer Portal sites](#).

- Select the portal service that you want to use with this Catalog.
 

For more information on using the Developer Portal, see [Developer Portal: Socialize your APIs](#).
- Click Create to save your changes.

Note: The account profile for the owner of the Catalog must have an email address specified to receive the creation notification, otherwise the Developer Portal creation operation will fail. To check whether your account profile has an email address, and specify a value, complete the following steps:

- Click the User icon , and then select My Account.
- Check the contents of the Email field, and specify a value if necessary, then click Save.

- To define Catalog properties, complete the following steps:
  - Click Properties in the Catalog settings navigation pane, then click Create.
  - Enter the property Name and Value, then click Create.

The properties that you define can be referenced in any of the API definitions in this Catalog. It is also possible to define properties that are specific to an API definition; see [Setting API properties](#). For information on how to reference a property in an API definition, see [Variable references in API Connect](#).

Note:

- If you define a Catalog property of the same name as an API property, the API property takes precedence over the Catalog property.
- If you change the value of a Catalog property, any API that references that property must be republished for it use the new value.

## What to do next

---

Import your own policies into the Catalog. For details, see [Importing a user-defined policy into a Catalog](#).

Note: When you import a policy, its implementation is imported into all Gateway devices associated with the Gateway service that hosts the Catalog. Additionally, API Connect modifies some object names and file locations to mark them with the appropriate Catalog name and policy version.

## Related concepts

---

- [Authoring policies](#)

## Related tasks

---

- [Working with user registries](#)

## Related information

---

- [Working with Products](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Configuring sub paths for Developer Portal sites

You can increase the specificity of an IBM® API Connect Developer Portal site URL by using sub paths.

By adding additional text to the URL of a Developer Portal site, you can create more site permutations for an organization. Developer Portal site URLs must be unique across all Catalogs.

Important:

- Sub paths are case-sensitive.
- The Developer Portal host name must contain only **lowercase** characters.

You can create a Developer Portal in the following format:

```
https://host_name.portal.com/sub_path_1/sub_path_2/.../sub_path_n
```

Where *sub\_path* can be any text that you want to implement to specify your site URL. There is no limit to the number of sub paths that you can have for the first site that you want to create.

However, you must not create overlaps in the naming of your URLs. A URL of one site must not be able to be confused with pages belonging to another site. For example, if you create the following site URL:

```
https://host_name.portal.com/bank/safe/dollar
```

Then, you cannot have the following site URL permutations:

```
https://host_name.portal.com/bank/safe
```

or

```
https://host_name.portal.com/bank/safe/dollar/cent
```

However, you can lengthen or shorten a site URL if you change a single sub path value, or the *host\_name.portal.com* value. For example, if you created the following site URL:

```
https://host_name.portal.com/bank/safe/dollar
```

Then, you can create the following site URL permutations:

```
https://host_name.portal.com/bank/euro
```

and

```
https://host_name.portal.com/bank/euro/vault/cent
```

Important: If you already have a site URL that ends in *sub\_path* texts, you cannot create a new site URL with the sub path text at the start.

For example, if you create the following site URL:

```
https://host_name.portal.com/bank
```

then, you cannot create the following site URL:

```
https://bank.host_name.portal.com
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---



## Managing Catalog membership

You manage Catalog membership by adding new users to the Catalog and assigning roles to the users.

### Before you begin

To manage Catalog members in the API Manager UI, a user must be assigned a role that has the Member\_>\_Manage permission for that Catalog. For more information on assigning Catalog permissions to a role, see [Creating and configuring Catalogs](#).

### Procedure

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Members.
3. To add an existing user, click Add\_>\_Add member. To add a new user, click Add\_>\_Invite member.  
If you are using the Add member option, you can add any user who is already a member of another Catalog, or of a Space, in the provider organization, and is neither currently a member of this Catalog, nor the Catalog owner.
4. Enter the email address of a new user, or search for, and select, an existing user.
5. Select the roles that you want to assign to the user.  
For details of the roles and the default permissions assigned to them, see [API Connect user roles](#). For details on how to create your own roles, see [Creating custom roles](#).

If Spaces are enabled in the Catalog, any role that you assign to the user at the Catalog level is assigned automatically to the user in all Spaces in that Catalog. Furthermore, if a user had originally been assigned a role only in a specific Space in a Catalog and you subsequently assign the user that role at the Catalog level, the Space specific role assignment is lost and the user now has that role in all Spaces in the catalog.

Note: You can subsequently change the roles assigned to a user by selecting or clearing the appropriate check boxes alongside that user on the Members tab. If a user was originally added to the provider organization, rather than to the Catalog itself, the following conditions apply:

- Any role assigned to the user at the provider organization level is assigned automatically to the user in all Catalogs, and cannot be removed at the Catalog level.

For details on adding a user to a provider organization, see [Adding provider organization users and assigning roles](#).

- In the Catalog, the user has the permissions that are configured for the role at the Catalog level.
- Any role that hasn't been assigned to the user at the provider organization level can be assigned to the user at the Catalog level.

6. If you are adding an existing user, click Create. If you are adding a new user, click Invite.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Importing a user-defined policy into a Catalog

You can make your user-defined policy available to API developers by importing it into an IBM® API Connect Catalog, or a Space in a Catalog.

### Before you begin

You must possess Catalog edit permissions to complete this task.

Before you can import a user-defined policy into IBM API Connect, you must have completed the following tasks:

1. Described your policy in a YAML file.
2. Implemented your policy by using DataPower processing rules and actions.

Note: The tasks described on this page apply only to the DataPower® Gateway (v5 compatible), not to the DataPower API Gateway.

### Procedure

To import a policy into a Catalog or Space in API Manager, complete the following steps.

1. Create a .zip file for your policy that contains the following folder structure:

```
policy.yaml
implementation/mypolicy.zip
```

where

- *policy.yaml* is your policy definition file.
- *implementation/mypolicy.zip* is your policy implementation for a policy that is deployed to the DataPower Gateway. The policy implementation file contains the DataPower processing rules and actions that were exported from DataPower.

The name of the .zip file must start with the name of the user-defined policy (as defined in the policy YAML file). If the implementation requires certificate and key files, these files must be added to the implementation directory.

Note: Your package .zip file must contain a .zip policy implementation file.

2. Use the developer toolkit command-line tool to import your policy into a Catalog or Space. When successfully imported, the policy appears on the policy palette of the API Manager assembly editor.

- a. Log in to the command-line tool, for example:

```
apic login --username userid --password password --server mgmt_endpoint_url --realm mode/realm
```

where

- *userid* is your user name (you must have permissions to be able to update resources in the organization and Catalog where the policy is being imported).
- *password* is your password.
- *mgmt\_endpoint\_url* the platform API endpoint URL, for example `platform-api.myserver.com`.
- *mode/realm* is your authentication scope. *mode* is the context of your login, and can be one of two options:
  - **admin** - use this option when you want to log in as an administrator.
  - **provider** - use this option when you want to log in as a Provider organization.

*realm* is your identity provider, and this can be an external provider such as Google, or the identity provider that is configured in the Cloud Manager.

For example, if you are signing in as an administrator by using Google, then the realm option is `--realm`

`admin/google`. If you are signing in as a Provider organization by using an LDAP user registry called *ibm-ldap*, then the realm option is `--realm provider/ibm-ldap`.

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

For more information about user registries, see [Managing Authentication and Security](#).

- b. Run the following command:

```
apic policies:create --catalog catalog --configured-gateway-service gateway --org organization --server mgmthost.com --scope scope [--space space] mypolicy.zip
```

where

- *catalog* is the Catalog name or ID that you want to import your policy into.

- `gateway` is the Configured Gateway Service name or ID.
- `organization` is the Provider Organization name or ID.
- `mgmthost.com` is the address of the management server endpoint, for example `example.server.dev.ciondemand.com`.
- `scope` has one of the following values:
  - `catalog` if the Catalog does not have Spaces enabled.
  - `space` if the Catalog has Spaces enabled. If you specify `space` for the `--scope` parameter you must also supply the `--space` parameter.
- (optional) `space` is the name of the Space. The `--space` parameter is required if the Catalog has Spaces enabled, in which case you must also include `--scope space` in the command.
- `mypolicy` is the name of your policy .zip file.

Note:

- You must import your user-defined policy into every Catalog in API Manager that you require your policy to run in.
- For the policy to be displayed on the palette in the [API assembly editor](#), you must import the user-defined policy into the Sandbox Catalog.
- If you have more than one Gateway service enabled in your Catalog, you must repeat the import operation for each Gateway service.
- If Spaces are enabled in a Catalog, a user-defined policy that you import into one Space is imported into **all** Spaces; you cannot import a user-defined policy into an individual Space in the Catalog. Any subsequent updates are also applied to all Spaces.

## Results

---

The user-defined policy is now imported into a Catalog or Space, and is shown in the list of available policies in the Policy Assembly tab of the API Editor in the API Manager. Policies are listed by their name and version number, and multiple versions of the same policy are grouped under a single heading.

## Related concepts

---

- [Authoring policies](#)

## Related information

---

- [Overview of the command-line tool](#)
- [apic policies:create](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using syndication in API Connect

With the IBM® API Connect syndication feature, you can partition your Catalogs into *Spaces*. Each Space is used by a different API provider development team and has its own set of management capabilities relating specifically to the APIs that the associated team publishes to that Space, enabling each team to manage their APIs independently.

When you stage or publish an API to a Catalog that has Spaces enabled, you specify the Space within that Catalog that you want to stage or publish to. However, application developers that access the Developer Portal for the Catalog are unaware of the Space partitioning of the Catalog and see the APIs as a coordinated offering.

Each Space has its own Product lifecycle management, subscription approvals, and analytics data. You use Space specific access control to restrict user access to each Space; for example, a developer in the Flights team is able to stage APIs only to the Flights Space.

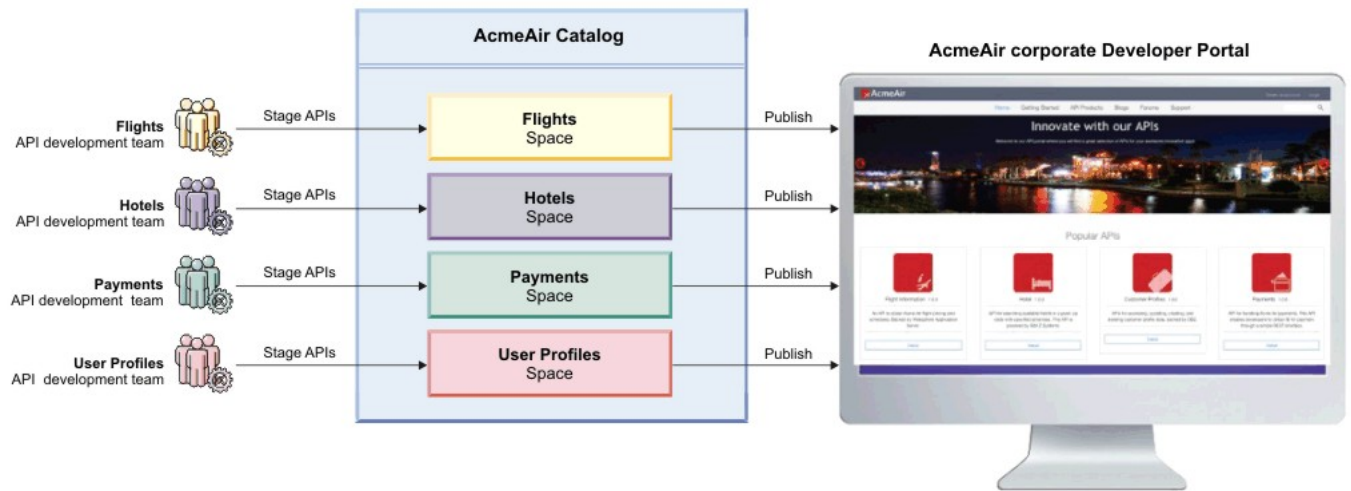
## Example

---

A travel company, AcmeAir, has separate API provider development teams for each of the following areas of their business:

- Flights
- Hotels
- Payments
- User Profiles

They have an AcmeAir corporate Catalog, with its associated corporate Developer Portal. They partition their Catalog into separate Spaces, one for each development team.



Spaces are described in detail in the following sections:

- [Enabling Spaces in a Catalog](#)  
To use the syndication feature in IBM API Connect, you must enable Spaces in any Catalog in which you require syndication capabilities.
- [Creating, modifying, and deleting Spaces](#)  
You use the API Manager user interface to create a new Space in a Catalog, modify the summary details, and delete a Space from a Catalog.
- [Working with Spaces](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can select a specific Space to work with in the API Manager user interface. This enables you to manage the Products and APIs that are specific to that Space, and control user access to the Space.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Enabling Spaces in a Catalog

To use the syndication feature in IBM® API Connect, you must enable Spaces in any Catalog in which you require syndication capabilities.

### Before you begin

If Spaces are enabled for a Catalog, Products (and their associated APIs) can be published only to a Space within that Catalog. For information about publishing, see [Staging a Product](#) for publishing by using the API Manager, and see [Publishing APIs](#) for publishing by using the developer toolkit.

Note: You cannot enable Spaces in a Catalog to which one or more Products have already been staged or published. If the Catalog does contain staged or published Products, first complete the following steps for each Product in the Catalog:

1. Remove all staged Products.
2. Retire, then remove, all published Products.

After enabling Spaces, re-stage or re-publish Products, and re-create application subscriptions, as required.


### About this task

By default, Spaces are disabled in a Catalog. You enable Spaces by modifying the Catalog settings. After Spaces are enabled for a Catalog, a default Space is automatically created that inherits all of the configuration settings from the Catalog.

Note: You can also enable Spaces by using the developer toolkit CLI; for details, see [apic spaces](#).

### Procedure

To enable Spaces in a Catalog, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Settings.
3. Set the Spaces slider control to the On position.
4. In the Enable Spaces window, click Enable.

### Results

Spaces are enabled for your Catalog, and a default Space is created that automatically inherits all of the configuration settings from the Catalog. The Catalog owner is also the owner of the default Space.

## What to do next

---

You can modify Space settings, and create more Spaces; see [Creating, modifying, and deleting Spaces](#) and [Working with Spaces](#) for more information.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



## Creating, modifying, and deleting Spaces

You use the API Manager user interface to create a new Space in a Catalog, modify the summary details, and delete a Space from a Catalog.

### Before you begin

---

Navigate to the Spaces page of the Catalog containing the Space you want to work with, by completing the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.  
The Spaces option is available only if Spaces are enabled in the Catalog; see [Enabling Spaces in a Catalog](#).

Note: You can also enable Spaces by using the developer toolkit CLI; for details, see [apic spaces](#).

### Procedure

---

- To create a new Space, complete the following steps:
  1. Click Add, Add a new Space.
  2. In the Space Details window, enter the Title of the Space and, optionally, a descriptive summary. A Name is entered automatically.  
Note: The value in the Name field is a single string that is used to identify the Space in developer toolkit CLI commands. The Title is used for display. To view the CLI commands to manage Spaces, see [apic spaces](#).

3. Click Create to create the Space.

Note:

- As the user who creates the Space, you are automatically the Space owner and have all Space permissions.
- When you create a new Space, it does not automatically inherit all of the configuration settings from the Catalog. So you should check and modify the settings as necessary.


- To modify the details of a Space, complete the following steps:
  1. On the Spaces page, select the Space whose details you want to modify.
  2. Use the options in the navigation pane to modify the details as required.

The following types of resource can be enabled in a Space:

- Gateway services
- API user registries
- OAuth providers
- TLS client profiles

Note:

- If you enable a resource of any of the following types in one Space in a Catalog, it is deployed to the gateway services across all Spaces in that Catalog:
  - API user registries
  - OAuth providers
  - TLS client profiles
- If you deploy any of the following types of resource to a gateway in a Space, it is actually deployed to the gateway at Catalog scope:
  - Custom policy
  - Global policy
  - Global policy pre-hook
  - Global policy post-hook
  - Gateway extension


- To delete a Space, complete the following steps:
  1. On the Spaces page, click the Manage icon  for the Space that you want to delete.
  2. Click Delete, then click Confirm to confirm deletion.

Note:

- You cannot delete a Space to which one or more Products have been staged or published.
- If there is only one Space in a Catalog, you cannot delete it, but you can disable Spaces for the Catalog. However, you cannot disable Spaces in a Catalog to which one or more Products have already been staged or published. If any of the Spaces in the Catalog do contain staged or published Products, first complete the following steps for the Products in each of the Spaces:

1. Remove all staged Products.
2. Retire, then remove, all published Products.

Then, to disable Spaces for the Catalog, complete the following steps:

1. Click  Settings in the Catalog.
2. Set the Spaces slider control to the Off position.

After disabling Spaces, re-stage or re-publish Products, and re-create application subscriptions, as required.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Working with Spaces

If Spaces are enabled in an IBM® API Connect Catalog, you can select a specific Space to work with in the API Manager user interface. This enables you to manage the Products and APIs that are specific to that Space, and control user access to the Space.

### Procedure

---

To work with a Space, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.

### What to do next

---

For details of the Space management tasks, see the following subtopics:

- [Managing Products in a Space](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can use the API Manager user interface to separately manage the Products that are staged or published to each Space.
- [Managing subscription requests in a Space](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can approve, or deny, individual requests to subscribe to Plans in Products that are published to the Space.
- [Managing application subscriptions in a Space](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can use the API Manager user interface to separately manage the application subscriptions to Plans in the Products that are published to a Space.
- [Managing Space membership](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can manage the members within the Space. You manage Space membership by adding new users to the Space and assigning roles to the users.
- [Managing user access in a Space](#)  
If Spaces are enabled in an IBM API Connect Catalog, you can manage the access that users have within the Space. You manage access by specifying the permissions that are assigned to user roles.
- [Managing Gateways in a Space](#)  
If Spaces are enabled in a Catalog, you can separately control which Gateway services are used by each of the Spaces in the IBM API Connect Catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing Products in a Space

If Spaces are enabled in an IBM® API Connect Catalog, you can use the API Manager user interface to separately manage the Products that are staged or published to each Space.

### Before you begin

---

To manage Products in a Space in the API Manager UI, a user must be assigned a role that has the Products\_>\_Manage permission. For more information on assigning Space permissions to a role, see [Managing user access in a Space](#).

### Procedure

---

To manage the Products that are staged or published to a specific Space, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.

### What to do next

---

For details of the management tasks that you can perform on the Products in the Space, see [Working with Products in the API Manager](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing subscription requests in a Space

If Spaces are enabled in an IBM® API Connect Catalog, you can approve, or deny, individual requests to subscribe to Plans in Products that are published to the Space.

## Before you begin




---

To manage Space subscription requests in the API Manager UI, a user must be assigned a role that has the Subscription Approvals, Manage permission. For more information on assigning Space permissions to a role, see [Managing user access in a Space](#).

## Procedure

---

To manage the subscription requests in a Space, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.
4. Click  Tasks in the API Manager UI navigation pane.  
The Tasks page lists only the subscription requests for the selected Space.

## What to do next

---

For details on how to manage approvals, see [Approving Product lifecycle and subscription requests](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing application subscriptions in a Space

If Spaces are enabled in an IBM® API Connect Catalog, you can use the API Manager user interface to separately manage the application subscriptions to Plans in the Products that are published to a Space.




## Before you begin

---

To manage application subscriptions in the API Manager UI, a user must be assigned a role that has the Subscriptions, Manage permission. For more information on assigning Space permissions to a role, see [Managing user access in a Space](#).

## Procedure

---

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.
4. In the navigation pane of the API Manager UI, click  Applications.  
All applications that have been registered in the Developer Portal associated with this Space are listed.
5. Expand the application whose subscriptions you want to manage.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing Space membership

If Spaces are enabled in an IBM® API Connect Catalog, you can manage the members within the Space. You manage Space membership by adding new users to the Space and assigning roles to the users.

## Before you begin

---

To manage Space members in the API Manager UI, a user must be assigned a role that has the Child, Manage permission for the Catalog that contains the Space. For more information on assigning Space permissions to a role, see [Managing user access in a Space](#).




## About this task

---

Note: You can add the same user to two or more Spaces and assign different roles in each, allowing a user to have differing levels of access in different Spaces.

## Procedure

---

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.
4. In the navigation pane of the API Manager UI, click  Members.
5. To add an existing user, click Add\_>\_Add member. To add a new user, click Add\_>\_Invite member.  
If you are using the Add member option, you can add any user who is already a member of another Space, or of a Catalog, in the provider organization, and is neither currently a member of this Space, nor the Space owner.
6. Enter the email address of a new user, or search for, and select, an existing user.
7. Select the roles that you want to assign to the user.  
For details of the roles and the default permissions assigned to them, see [API Connect user roles](#). For details on how to create your own roles, see [Creating custom roles](#).

If a user was originally added either to the provider organization, or to the Catalog that contains the Space, rather than to the Space itself, the following conditions apply:

- Any role assigned to the user at the provider organization or Catalog level is assigned automatically to the user in all Spaces, and cannot be removed at the Space level.

For details on adding a user to a provider organization, see [Adding provider organization users and assigning roles](#).

For details on adding a user to a Catalog, see [Managing Catalog membership](#).

- In the Space, the user has the permissions that are configured for the role at the Space level.  
For details on configuring role permissions for a Space, see [Managing user access in a Space](#).

- Any role that hasn't been assigned to the user at the provider organization or Catalog level can be assigned at the Space level

Note: You can subsequently change the roles assigned to a user by selecting or clearing the appropriate check boxes alongside that user on the Members page.

8. If you are adding an existing user, click Create. If you are adding a new user, click Invite.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---





## Managing user access in a Space

If Spaces are enabled in an IBM® API Connect Catalog, you can manage the access that users have within the Space. You manage access by specifying the permissions that are assigned to user roles.

### Before you begin

To manage Space permissions in the API Manager UI, a user must be assigned a role that has the Child\_>\_Manage permission for the Catalog that contains the Space. For information on adding users to a Space and assigning user roles, see [Managing Space membership](#).

### Procedure

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.
4. In the navigation pane of the API Manager UI, click  Settings.
5. Click Roles, click the options icon  alongside the role whose permissions you want to modify, then click Edit.
6. Select the required permissions, then click Save.

### Related tasks

- [Creating custom roles](#)

### Related information

- [API Connect user roles](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing Gateways in a Space

If Spaces are enabled in a Catalog, you can separately control which Gateway services are used by each of the Spaces in the IBM® API Connect Catalog.



## Before you begin

---

To manage Space Gateway settings in the API Manager UI, a user must be assigned a role that has the Child\_>Manage permission for the Catalog that contains the Space. For information on adding users to a Space and assigning user roles, see [Managing Space membership](#).

## About this task




---

Note: If Spaces are enabled in a Catalog then you configure the Gateway settings for each Space separately, **not** for the Catalog as a whole.

## Procedure

---

To manage the Gateway settings for a Space, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. In the navigation pane of the API Manager UI, click  Spaces.
3. Select the Space that you want to work with.
4. In the navigation pane of the API Manager UI, click  Settings.
5. Click Gateway Services.  
The Gateway services that are currently enabled for the Space are listed.
6. To change the Gateway service configuration for the Space, click Edit, then select which Gateway services you want the Space to use.
7. Click Save to save your changes.

## Related information

---

- [Configuring the initial Gateway service](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Administering Consumer organizations

---

Manage the Consumer organizations that access your APIs and Plans when their users sign up to use the IBM® API Connect Developer Portal.

## About this task

---

Application developers in Consumer organizations access the Developer Portal and sign up to use the Plans in the Products you create in the API Manager. You can create more than one Consumer organization and each organization has one owner. Application developers are added through the Developer Portal UI.

You can also create groups of Consumer organizations. A Consumer organization group is an efficient way of grouping application developers together to publish Products to, making it easier to control who has access to the Plans in the Products.

Use the following tasks to administer your Consumer organizations and Consumer organization groups:

- [Creating a Consumer organization](#)  
If you have permission to manage developers, you can create new Consumer organizations in a Catalog. Consumer organizations contain the application developers that subscribe to use the APIs that are published to the Catalog
- [Editing a Consumer organization](#)  
You can change the title of an IBM API Connect Consumer organization.
- [Deleting a Consumer organization](#)  
When you delete a Consumer organization, all of the resources in that organization are removed from the cloud. However, the account of the organization owner remains in IBM API Connect.
- [Working with Consumer organization groups](#)  
A *Consumer organization group* is a collection of Consumer organizations, and provides an efficient way of controlling who can see, and subscribe to, the APIs in your Products. By using a Consumer organization group, you can define this access for all the developers in the organizations in that group in a single operation, rather than having to define access for the organizations separately.
- [Managing developer applications](#)  
If an application is behaving suspiciously, exceeding rate limits, or has been compromised, you can disable it to block it from accessing your APIs.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Consumer organization

---

If you have permission to manage developers, you can create new Consumer organizations in a Catalog. Consumer organizations contain the application developers that subscribe to use the APIs that are published to the Catalog

## Before you begin

---

Your IBM® API Connect Developer Portal must be enabled to perform this task. For more information, see [Creating and configuring Catalogs](#). To complete this task, you must be defined as a member in a Catalog or Space (if Spaces are enabled), and you must be assigned a role that has the Consumer-Org\_ Manage permission. For more information, see [Adding provider organization users and assigning roles](#) and [Managing Catalog membership](#).

## About this task

---



You can create a consumer organization in either of the following ways:

- Invite a user to be a consumer organization owner, in which case they receive an invitation email with an activation link, and can complete the creation of the consumer organization and specify the title.
- Create the consumer organization yourself, specifying the owner and title. There are two ways you can specify the owner:
  - Specify the user ID of an existing user.
  - If you are using a Local User Registry, specify a new user ID; the user is added to the registry and becomes the owner of the consumer organization.


If Spaces are enabled in your Catalog, when you create a Consumer organization, it is added to the Catalog and its Spaces. For more information about enabling Spaces, see [Using syndication in API Connect](#).

## Procedure

---

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.

For more information about enabling Spaces, see [Using syndication in API Connect](#).

3. In the navigation pane of the API Manager UI, click  Consumer Organizations.
4. To invite a user to be a consumer organization owner, complete the following steps:
  - a. Click Add\_ Invite organization owner.
  - b. Enter the email address of the organization owner.
  - c. Click Invite.

The new Consumer organization is added to the list, and an email invitation is sent to the owner; by using the activation link, the owner specifies the title of the organization. The status is shown as `Pending` until the recipient of the email clicks the link in the email to complete the creation of their Developer Portal account, after which the status changes to `Enabled`.

5. To create the consumer organization yourself, complete the following steps:
  - a. Click Add\_ Create organization.
  - b. Enter the organization Title. A Name is entered automatically.

The value in the Name field is a single string that is used to identify the consumer organization in developer toolkit CLI commands. The title is used for display.

To view the CLI commands to manage Consumer organizations, see [apic consumer-orgs](#).
  - c. Select the required user registry.
  - d. Specify the organization owner; complete one or other of the following steps, either to specify an existing user, or to create a new user:
    - To specify the user ID of an existing API Connect user who is in the selected user registry, complete the following steps:
      - For the Type of user, select Existing.
      - In the Username field, enter the user ID of the organization ownerAfter the consumer organization is created, the specified user can immediately log in to the Developer Portal associated with the Catalog.
    - If you selected a Local User Registry, then to specify a new user ID, complete the following steps:
      - For the Type of user, select New User.
      - In the Username field, enter the new user ID.
      - Enter the Email, First Name, Last Name, and Password.After the consumer organization is created, the new user will have been added to the selected registry and can immediately log in to the Developer Portal associated with the Catalog, by using the specified user ID and password.
  - e. Click Create to complete the creation of the consumer organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Editing a Consumer organization

You can change the title of an IBM® API Connect Consumer organization.

## Before you begin

---

To complete this task, you must be defined as a member in a Catalog or Space (if Spaces are enabled), and you must be assigned a role that has the Consumer-Org\_ Manage permission. For more information, see [Adding provider organization users and assigning roles](#) and [Managing Catalog membership](#).

## About this task





---

If Spaces are enabled in your Catalog, when you edit the details of a Consumer organization, the updates are applied to the Catalog and its Spaces. For more information about enabling Spaces, see [Using syndication in API Connect](#).

## Procedure

---

To change the title of a Consumer organization, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.For more information about enabling Spaces, see [Using syndication in API Connect](#).
3. In the navigation pane of the API Manager UI, click  Consumer Organizations.
4. Click the options icon  for the Consumer organization that you want to work with, then click Edit.
5. Update the Title, then click Save.

## Results

---

The title of the Consumer organization is changed. All application developers that are members of that Consumer organization will see the title change reflected in the Developer Portal user interface.

## Related tasks

---

- [Publishing a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deleting a Consumer organization

---

When you delete a Consumer organization, all of the resources in that organization are removed from the cloud. However, the account of the organization owner remains in IBM® API Connect.

## Before you begin

---

To complete this task, you must be defined as a member in a Catalog or Space (if Spaces are enabled), and you must be assigned a role that has the Consumer-Org\_> Manage permission. For more information, see [Adding provider organization users and assigning roles](#) and [Managing Catalog membership](#).

Important: If you delete a Consumer organization, client IDs and secrets associated with applications that were registered by users in the Consumer organization can no longer be used to call APIs. However, the account of the organization owner remains in API Connect.

## About this task





---

When you delete a Consumer organization, it is removed permanently and users can no longer access that organization.  
If Spaces are enabled in your Catalog, when you delete a Consumer organization, it is deleted from the Catalog and its Spaces. For more information about enabling Spaces, see [Using syndication in API Connect](#).

## Procedure

---

To delete a Consumer organization, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.For more information about enabling Spaces, see [Using syndication in API Connect](#).
3. In the navigation pane of the API Manager UI, click  Consumer Organizations.
4. Click the options icon  alongside the Consumer organization you want to work with, then click Delete.
5. Click Confirm to delete the Consumer organization.

Note: If the consumer organization is the only one specified in the custom visibility settings for a published Product, the deletion operation will fail. For more information, see [Changing the availability of a Product](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Working with Consumer organization groups

A *Consumer organization group* is a collection of Consumer organizations, and provides an efficient way of controlling who can see, and subscribe to, the APIs in your Products. By using a Consumer organization group, you can define this access for all the developers in the organizations in that group in a single operation, rather than having to define access for the organizations separately.

### About this task

---

A Consumer organization group might represent particular business partners, internal organizations, or other groups of application developers.

When you create a Product, you can define which application developers can access the APIs in that Product. You can configure the following types of access:




- Visibility, which specifies the application developers who can see the APIs in the Developer Portal.
- Subscribability, which specifies the application developers who can create subscribe to use the APIs.



You configure this access by selecting the required Consumer organizations, and Consumer organization groups. You can also change the visibility and subscribability settings for a Product after it has been published. For more information, see [Creating a draft Product](#) and [Changing the availability of a Product](#).

### Procedure

---

To work with Consumer organization groups, you first navigate to the Consumer organization groups page. You then create new groups, and edit or delete existing groups.

- Navigating to the required Consumer organization groups page.
  1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
  2. If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
    - a. In the navigation pane of the API Manager UI, click  Spaces.
    - b. Select the Space that you want to work with.
  3. Click  Consumer Organizations, then click Manage Groups.
- Creating a Consumer organization group.
  1. Click Add.
  2. Enter a Title for the new group and, optionally, a Summary. A Name is entered automatically. The title is used for display.
  3. Use the Type to add organizations/groups field to search for, and select, the consumer organizations that you want to add to the group.

Note: To prevent excessive search results lists, only the first 10 matches are displayed when you begin typing in the field; lengthen your search string to shorten the list for locating the required consumer organization.
  4. Click Save to create the group.
- Editing a Consumer organization group.
  1. Click the options icon  for the Consumer organization group that you want to work with, then click Edit.
  2. Make the required updates, then click Save.
- Deleting a Consumer organization group.
  1. Click the options icon  for the Consumer organization group that you want to work with, then click Delete.
  2. Click Confirm to delete the group.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing developer applications

If an application is behaving suspiciously, exceeding rate limits, or has been compromised, you can disable it to block it from accessing your APIs.

### Before you begin

---

To complete this task, you must be assigned a role that has the Applications\_>\_Manage permission. For more information, see [Adding provider organization users and assigning roles](#) and [Managing Catalog membership](#).

### About this task

---

You might need to disable an application for any of the following reasons:






- The application is trying to maliciously hack your APIs.
- You have soft rate limits enabled on your Plan and the application is consistently exceeding the quota.
- You become aware that the client secret has been compromised and that the developer has not reset it.

Note: If a disabled application calls an API, the API gateway returns error code 403.

### Procedure

---

To disable an application, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
2. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.For more information about enabling Spaces, see [Using syndication in API Connect](#).
3. In the navigation pane of the API Manager UI, click  Applications.  
All applications that have been registered in the Developer Portal associated with this Catalog are listed.
4. Click the options icon  for the application that you want to disable, then click Disable.  
The value in the STATE column for the application changes to Disabled. If Spaces are enabled in the Catalog, the application is disabled in the Catalog and its Spaces.
5. Optional: To enable an application, click the options icon  for the application that you want to enable, then click Enable.  
If Spaces are enabled in the Catalog, the application is reactivated in the Catalog and its Spaces.

## Results

---

When you disable or enable an application, the application developer is notified by email and in their Developer Portal activity feed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Security and authentication

---

In API Manager, you can use TLS profiles to secure the transmission of data between the management server and other API Connect subsystems and external services, and also configure user registries to securely authenticate your Catalogs and APIs.

For details of authentication in IBM API Connect, see the following subtopics:

- [Creating a TLS client profile](#)  
In the IBM API Connect API Manager interface, TLS profiles are used to secure transmission of data between the management server and other API Connect subsystems and external services. TLS and SSL certificates guarantee that information you submit will not be stolen or tampered with. In this topic, you learn how to create a TLS profile in API Manager.
- [Authenticating by using your enterprise user registry](#)  
IBM API Connect supports a variety of user registry types for authenticating users and securing APIs.
- [Configuring a native OAuth provider](#)  
Native OAuth providers are configured and managed by you within your cloud.
- [Configuring a third-party OAuth provider](#)  
Enter the secure endpoints to provide OAuth authentication from a third party.
- [OAuth concepts for API Connect](#)  
OAuth is a token-based authorization protocol that allows third-party websites or applications to access user data without requiring the user to share personal information.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a TLS client profile

---

In the IBM® API Connect API Manager interface, TLS profiles are used to secure transmission of data between the management server and other API Connect subsystems and external services. TLS and SSL certificates guarantee that information you submit will not be stolen or tampered with. In this topic, you learn how to create a TLS profile in API Manager.

### Before you begin

---

One of the following roles is required to configure TLS Profiles:

- Organization Administrator
- Owner
- Custom role with the `Settings: Manage permissions`

### About this task

---


API Connect supports the use of TLS and SSL certificates, but does not itself produce strong encryption keys or manage your encryption keys. Encryption keys should be created and managed according to your own procedures. For more information, see [Viewing certificate details and adding certificates to a keystore or truststore](#) and [Generating a PKCS#12 file for Certificate Authority](#).

Note: If you update a TLS profile that is associated with a Gateway service, the updates are not automatically propagated to Gateway servers.

For instructions on how to configure the toolkit command-line tool to use TLS certificates when connecting to API Manager, see [Configuring the command-line tool to use TLS certificates](#).

## Procedure

To create a TLS profile, complete the following steps:

1. In the API Manager, click  Resources.
2. Select TLS.
3. Click Create in the TLS Client Profile table.
4. Enter the fields to configure the TLS Client Profile:

Field	Description
Title (required)	Enter a Title for the profile. The title is displayed on the screen.
Name (required)	<p>The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage a TLS Client Profile, see <a href="#">apic tls-client-profiles</a>.</p> <p>Important: The name of the TLS Client Profile as saved on the DataPower® Gateway, depending on the gateway type, is as follows:</p> <ul style="list-style-type: none"> <li>• DataPower API Gateway: <code>provider-org-name_catalog-name_tlsp-tls-profile-nameV1.0.0</code></li> <li>• DataPower Gateway (v5 compatible): <code>provider-org-name-tls-profile-nameV1.0.0</code></li> </ul> <p>where</p> <ul style="list-style-type: none"> <li>• <code>tls-profile-name</code> is the value of the auto-generated Name field for the TLS client profile in API Connect.</li> <li>• <code>provider-org-name</code> is the name of the provider organization containing the TLS client profile.</li> <li>• <code>catalog-name</code> is the name of the Catalog, in that provider organization, containing the TLS client profile.</li> </ul>
Version (required)	Assign a version number for the profile. Using version numbers allows you to create multiple server profiles with the same name and different configurations, for example, <i>MyProfile 1.0</i> and <i>MyProfile 1.1</i> .
Summary (optional)	Enter a description of the profile.
Protocols (required)	Select one or more supported TLS protocol versions. The default is 1.2.
Server Connection (optional)	<p>Specify whether to support weak or insecure credentials.</p> <ul style="list-style-type: none"> <li>• Allow insecure server connections - Insecure server connections may result from self-signed certificates, expired or corrupted certificates, or certificates from an unknown or untrusted source. Check this box to allow the connection to proceed with an insecure connection. The default is to not allow insecure server connections.</li> <li>• Support Server Name Indication (SNI) - Check this box to enable SNI. SNI allows support for multiple certificates presented on the same IP address using different host names. The client profile sends the name of a virtual domain as part of the TLS negotiation. The default is to enable SNI.</li> </ul>
Keystore (optional)	A Keystore is a repository containing public and private key pairs. Select the keystore where you will store the certificates for the profile. Default keystores are provided, and you can also create your own.
Truststore (optional)	A Truststore is a repository containing verified public keys, which are usually obtained from a third-party certificate authority. Truststores provide secure identification for peer systems. A truststore is usually used when mutual authentication is enabled. Select a truststore for the profile. Default truststores are provided, and you can also create your own.
Ciphers (required)	Cipher suites are encryption/decryption algorithms used to secure HTTPs communication within the API Connect ecosystem. All ciphers are enabled by default. You can unselect one or more ciphers if they are not needed for the profile.

5. Click Save.

- [Creating a Keystore](#)  
Keystores contain matched pairs of public certificates and private keys used to confirm identity and encrypt/decrypt data transmission over HTTPS.
- [Creating a Truststore](#)  
Truststores are repositories containing trusted certificates with verified public keys. The certificates in the truststore are usually obtained from a third-party certificate authority (CA).
- [Generating a PKCS#12 file for Certificate Authority](#)  
PKCS#12 (P12) files define an archive file format for storing cryptographic objects as a single file. API Connect supports the P12 file format for uploading a keystore and truststore. The keystore should contain both a private and public key along with intermediate CA certificates.
- [Generating a self-signed certificate using OpenSSL](#)  
OpenSSL is an open source implementation of the SSL and TLS protocols. It provides the transport layer security over the normal communications layer, allowing it to be intertwined with many network applications and services.
- [Viewing certificate details and adding certificates to a keystore or truststore](#)  
You can view details for the certificates in an existing keystore or truststore and add additional certificates. You might need add certificates when a certificate or PKCS #12 (P12) file expires.
- [Defining elliptic curve cryptographic schemes for a TLS client profile](#)  
You define the elliptic curve cryptographic schemes for a TLS client profile by using the developer toolkit CLI.

## Related tasks

- [Working with user registries](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Keystore

Keystores contain matched pairs of public certificates and private keys used to confirm identity and encrypt/decrypt data transmission over HTTPS.

### Before you begin

---

API Manager supports and uses TLS certificates, but does not produce strong encryption keys or manage your encryption keys. Encryption keys are generated and managed according to your own procedures. For more information, see [Generating a PKCS#12 file for Certificate Authority](#) and [Generating a self-signed certificate using OpenSSL](#).

One of the following roles is required to configure Keystores:

- Organization Administrator
- Owner
- Custom role with the `Settings: Manage permissions`

### About this task


---

API Connect includes pre-configured Keystores which may be used for testing purposes. For production environments, we suggest creating a new, secure Keystore.

### Procedure

---

Perform the following steps to create a TLS Client profile:

1. In the API Manager, click  Resources.
2. Select TLS.
3. Click Create in the Keystore table.

Field	Description
Title (required)	Enter a Title for the Keystore. The title is displayed on the screen.
Name (required)	The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage keystores, see <a href="#">apic keystores</a> .
Summary (optional)	Enter a brief description.
Private Key & Public Key: Step 1: Upload private key	Upload the file containing the private key certificate. If necessary, you can click Browse to locate the file. If the file contains both the private and public keys, upload it in Step 1. Private and public keys are always uploaded in pairs, either in a single file or separate files.
Private key password (optional)	Enter the password for the private key if it has a password.
Private Key & Public Key: Step 2: Upload public key	If the public key is contained in a separate file, upload it in Step 2. Private and Public keys are always uploaded in pairs, either in a single file or separate files.

4. Click Save.  
Note: After they have been uploaded, private keys cannot be downloaded from API Connect.

### Related tasks

---

- [Creating a TLS client profile](#)
- [Creating a Truststore](#)
- [Generating a PKCS#12 file for Certificate Authority](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a Truststore

Truststores are repositories containing trusted certificates with verified public keys. The certificates in the truststore are usually obtained from a third-party certificate authority (CA).

### Before you begin

---

One of the following roles is required to configure truststores:

- Organization Administrator
- Owner
- Custom role with the `Settings: Manage permissions`


### About this task

---

API Manager supports and uses TLS certificates, but does not produce strong encryption keys or manage your encryption keys. Encryption keys are generated and managed according to your own procedures. For more information, see [Generating a PKCS#12 file for Certificate Authority](#) and [Generating a self-signed certificate using](#)

API Connect includes pre-configured Truststores which may be used for testing purposes. For production environments, we suggest creating a new, secure Truststore.

## Procedure

1. In the API Manager, click  Resources.
2. Select TLS.
3. Click Create in the Truststore table.

Field	
Title (required)	Enter a Title for the Truststore. The title is displayed on the screen.
Name (required)	The Name is auto-generated. The value in the Name field is a single string that can be used in developer toolkit CLI commands. To view the CLI commands to manage truststores, see <a href="#">apic truststores</a> .
Summary (optional)	Enter a brief description.
Public Keys	Upload the file containing the public key certificate. If necessary you can click Browse to locate the file.

4. Click Save.

## Related tasks

- [Creating a TLS client profile](#)
- [Creating a Keystore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Generating a PKCS#12 file for Certificate Authority

PKCS#12 (P12) files define an archive file format for storing cryptographic objects as a single file. API Connect supports the P12 file format for uploading a keystore and truststore. The keystore should contain both a private and public key along with intermediate CA certificates.

## Before you begin

One of the following roles is required to add a key to a keystore or truststore:

- Organization Administrator
- Owner
- Custom role with the `Settings: Manage permissions`

Before you can generate a P12 file, you must have a private key (for example: key.pem), a signed certificate by a Certificate Authority (for example certificate.pem) and one or more certificates from the CA authority (known as *intermediate CA certificates*).

Note: If your certificate file contains more than one certificate, you must manually split the file and create a single file for each entry. Each entry must be bound by the following markers:

```
-----BEGIN CERTIFICATE-----  
-----END CERTIFICATE-----
```

## Procedure

1. If you have intermediate certificates from your CA, concatenate them into a single .pem file to build your **caChain**. Be sure to enter a new line following each certificate's data.

```
cat ca1.pem ca2.pem ca3.pem > caChain.pem  
cat caChain.pem  
-----BEGIN CERTIFICATE-----  
MIIEPjCCA46gAwIBAgIQEod26KZabjd+BQMGlDwl6jANBgkqhkiG9w0BAQUFADCBA  
...  
lQX7CkTJn61AJUusyEa8H/gjVQnHp4VOLFR/dKgeVcCRvZF7Tt5AuiyHY  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
MIIEPDCCAySgAwIBAgIQSEus8arH1xND0aJ0NUmXJTANBgkqhkiG9w0BAQUFADBVB  
...  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
MIIEAjCCA66gAwIBAgIBATANBgkqhkiG9w0BAQUFADBvMQswCQYDVQGEwJTRTEU  
...  
-----END CERTIFICATE-----
```


2. Create the P12 file including the private key, the signed certificate and the CA file you created in step 1, if applicable. Omit the **-CAfile** option if you don't have CA certificates to include.

The following command uses OpenSSL, an open source implementation of the SSL and TLS protocols.

```
openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12 -CAfile caChain.pem -chain
```

Once the certificate file is created, it can be uploaded to a keystore.



3. Once the certificate file is created, it can be uploaded to a Truststore. In the API Manager, click .
4. Select TLS.
5. Click Create in the Keystore table.
6. Create a Keystore and upload the certificate file following the instructions at [Creating a Keystore](#).

Note:

- API Connect supports only the P12 (PKCS12) format file for the present certificate.
- Your P12 file must contain the private key, the public certificate from the Certificate Authority, and all intermediate certificates used for signing.
- Your P12 file can contain a maximum of 10 intermediate certificates.

7. Click Save.

## Results

---

Your P12 file is generated and ready to upload.

## What to do next

---

Upload your P12 file to IBM® API Connect. For more information, see [Creating a TLS client profile](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Generating a self-signed certificate using OpenSSL

OpenSSL is an open source implementation of the SSL and TLS protocols. It provides the transport layer security over the normal communications layer, allowing it to be intertwined with many network applications and services.

## Before you begin

---

One of the following roles is required to complete this task:

- Organization Administrator
- Owner
- Custom role with the `Settings`: `Manage permissions`

## About this task

---

This topic tells you how to generate a self-signed SSL certificate request using the OpenSSL toolkit to enable HTTPS connections.

## Procedure

---

To generate a self-signed SSL certificate using the OpenSSL, complete the following steps:

1. Write down the Common Name (CN) for your SSL Certificate. The CN is the fully qualified name for the system that uses the certificate. For static DNS, use the hostname or IP address set in your Gateway Cluster (for example, `192.16.183.131` or `dp1.acme.com`).
2. Run the following OpenSSL command to generate your private key and public certificate. Answer the questions and enter the Common Name when prompted.

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem
```

3. Review the created certificate:

```
openssl x509 -text -noout -in certificate.pem
```


4. Combine your key and certificate in a PKCS#12 (P12) bundle:

```
openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12
```

5. Validate your P2 file.

```
openssl pkcs12 -in certificate.p12 -noout -info
```

Once the certificate file is created, it can be uploaded to a keystore.

6. In the API Manager, click  Resources.
7. Select TLS.
8. Click Create in the Keystore table.
9. Create a Keystore and upload the certificate file following the instructions at [Creating a Keystore](#).

Note:

- API Connect supports only the P12 (PKCS12) format file for the present certificate.
- Your P12 file must contain the private key, the public certificate from the Certificate Authority, and all intermediate certificates used for signing.
- Your P12 file can contain a maximum of 10 intermediate certificates.

10. Click Save.

## Related tasks

---

- [Creating a TLS client profile](#)
- [Creating a Keystore](#)
- [Creating a Truststore](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Viewing certificate details and adding certificates to a keystore or truststore

You can view details for the certificates in an existing keystore or truststore and add additional certificates. You might need add certificates when a certificate or PKCS #12 (P12) file expires.

### Before you begin

---

One of the following roles is required to edit Keystores and Truststores:

- Organization Administrator
- Owner
- Custom role with the `Settings: Manage permissions`

### About this task


---

API Connect includes pre-configured Keystores and Truststores which can be used for testing purposes. For production environments, we suggest creating a new, secure Keystore or Truststore. From an existing Keystore or Truststore, you can edit the title, view the details of the certificates, and add new certificates.

### Procedure

---

To work with existing Keystores and Truststores:

1. In the API Manager, click  Resources.
2. Select TLS.
3. Click the name of the keystore or truststore.
4. Edit the name and summary if needed.
5. The Subject, Finger Print, and Expiration date is shown for each certificate in the keystore or truststore. Click > to view the certificate details.
6. Add additional private and/or public keys if needed.
7. Click Save.

Note: After they have been uploaded, private keys cannot be downloaded from API Connect.

### Related tasks

---

- [Creating a Truststore](#)
- [Creating a Keystore](#)
- [Generating a PKCS#12 file for Certificate Authority](#)
- [Generating a self-signed certificate using OpenSSL](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Defining elliptic curve cryptographic schemes for a TLS client profile

You define the elliptic curve cryptographic schemes for a TLS client profile by using the developer toolkit CLI.

### About this task

---

To define elliptic curve cryptographic schemes for a TLS client profile, you include `elliptic_curve_auto_negotiation` and `elliptic_curve` properties in a YAML file definition for the TLS client profile. The `elliptic_curve` property lists the required elliptic curve cryptographic schemes. For example:

```
elliptic_curve_auto_negotiation: false
elliptic_curve:
- secp521r1
- secp384r1
- prime256v1
```

You then use the developer toolkit CLI to create the TLS client profile in API Connect.

When `elliptic_curve_auto_negotiation` is set to `true`, the system negotiates the Elliptic-curve Diffie-Hellman (ECDH) key agreement automatically with its peer, and any `elliptic_curve` property settings are ignored.

The following example shows a complete YAML file for a TLS client profile:

```
type: tls_client_profile
name: my-tls-client-profile
version: 1.0.0
title: My TLS client profile
protocols:
- tls_v1.2
ciphers:
- ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
elliptic_curve_auto_negotiation: false
elliptic_curve:
- sect163k1
insecure_server_connections: false
server_name_indication: true
```

Note:

- The `elliptic_curve_auto_negotiation` option is not supported by any of the API Connect gateway types. If the TLS client profile is targeted for an API Connect gateway; this setting is ignored by the gateway.
- The elliptic curve cryptographic schemes shown in each row of the following table are equivalent. However, API Connect recognizes only one or the other depending on how the TLS profile is used, as indicated in the table.

Table 1.

API enforcement on the gateway	API Connect server access security
secp192r1	prime192v1
secp256r1	prime256v1

Therefore, if you want to use either of these schemes and are unsure whether you are targeting the TLS client profile to the API Connect gateway for API enforcement, whether you are using it to secure user access to the API Connect servers, or whether it will be used for both purposes, specify both equivalent schemes; API Connect will simply ignore the non-relevant scheme. For example:

```
elliptic_curve:
.
.
- secp256r1
- prime256v1
.
.
```

## Procedure

To create a TLS client profile with elliptic curve cryptographic schemes defined, complete the following steps:

1. Create a YAML file definition for your TLS client profile, with the required `elliptic_curve` property.
2. Log in to the management server from the developer toolkit CLI. Log in either as a member of the cloud administration organization or as a member of a provider organization, depending on where you want to create the TLS client profile. For details, see [Logging in to a management server](#).
3. Create the TLS client profile by using the following command:

```
apic tls-client-profiles:create --server mgmt_endpoint_url --org organization_name tls_client_profile_yaml_file
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL, and is the same as that which was used when you logged in at step 2.
- `organization_name` is either `admin`, for the cloud administration organization, or the name of your provider organization, and is the same as that which was used when you logged in at step 2.
- `tls_client_profile_yaml_file` is the name of the YAML file that contains the definition for your TLS client profile.

Note: When you install IBM® API Connect, the API Connect gateway has a pre-supplied default TLS client profile that is used for API enforcement if you do not configure a TLS client profile; you cannot configure this default TLS client profile on the gateway.

For reference details of all the `apic tls-client-profiles` commands, see [apic tls-client-profiles](#).

You can also complete the operations described in this topic by using the API Connect REST APIs; see the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authenticating by using your enterprise user registry

IBM® API Connect supports a variety of user registry types for authenticating users and securing APIs.

You can use your enterprise user registry for authentication in API Connect if it is of one of the following types:

LDAP directory

If your user registry uses Lightweight Directory Access Protocol (LDAP), you can use it in API Connect for both user authentication and API security.

Authentication URL User Registry

You can configure a non-LDAP user registry by using an authentication URL user registry. An authentication URL user registry enables integration with third-party authentication providers. You can use an authentication URL user registry in API Connect for both user authentication and API security.

Local User Registry

You can authenticate users with a local user registry. A local user registry is an internal registry stored within API Connect.

OpenID Connect

You can authenticate users through an identity provider that supports the OpenID Connect (OIDC) authentication protocol.

- [Working with user registries](#)  
To secure the APIs that are published to your IBM API Connect Catalogs, you authenticate with user registries.
- [Modifying the configuration details for a user registry](#)  
You modify the configuration details for a user registry by using the User Registries page in the API Manager user interface of IBM API Connect.
- [Password lockout criteria](#)  
You can be locked out of your account if you attempt to log in and fail consecutively.

## Related tasks

---

- [Working with user registries](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---


## Working with user registries

To secure the APIs that are published to your IBM® API Connect Catalogs, you authenticate with user registries.

### Procedure

---

To configure a user registry, complete the following steps:

1. In the API Manager user interface, select  Resources.
2. On the Resources page, select User Registries > Create.
3. Select the tile for the user registry type you want, and continue to the instructions for configuring user authentication for your selection:

Authentication URL User Registry

For more information, see [Creating an Authentication URL user registry](#).

LDAP User Registry

For more information, see [Creating an LDAP user registry in API Manager](#).

Local User Registry

For more information, see [Creating a Local User Registry](#).

### Results

---

The user registry information is added to API Manager.

### What to do next

---

You can now use API Manager to chart and manage your Catalogs.

- [Creating an LDAP user registry in API Manager](#)  
You can use the API Manager UI to configure an organization-specific LDAP user registry to provide user authentication and onboarding for the Developer Portal. APIs can also be secured with an LDAP user registry.
- [Creating an Authentication URL user registry](#)  
An Authentication URL user registry provides a simple mechanism for authenticating users by referencing a custom identity provider.
- [Creating a Local User Registry](#)  
A Local User Registry (LUR) can be created to provide user authentication for API Manager.
- [Creating an OIDC user registry](#)  
Create an organization-specific OIDC user registry when multi-factor authentication (MFA) is required.

## Related tasks

---

- [Modifying the configuration details for a user registry](#)

## Related information

---

- [Developer Portal: Socialize your APIs](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating an LDAP user registry in API Manager

You can use the API Manager UI to configure an organization-specific LDAP user registry to provide user authentication and onboarding for the Developer Portal. APIs can also be secured with an LDAP user registry.

## Before you begin

To configure an LDAP user registry as a resource in API Manager, the LDAP directory must be created and populated for use with your API Connect ecosystem.

LDAP registries can be used to secure APIs, or for securing a Catalog to authenticate Developer Portal users.

Important: If you are using an LDAP registry to secure APIs, note the following limitations:

- Authentication methods Compose DN and Compose UPN are not supported with the DataPower® API Gateway.
- The STARTTLS protocol, which upgrades an insecure protocol to a secure one by applying TLS security, is not supported with an LDAP user registry

One of the following roles is required to configure an LDAP user registry:

- Organization Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

## About this task

You can create an LDAP user registry that is specific to a provider organization, or one that can be shared and available to all of the provider organizations in your API Connect environment. An organization-specific LDAP user registry can be used for authenticating Developer Portal users in a specific provider organization. While a shared LDAP user registry can be used across the Cloud Manager, the API Manager, and the Developer Portal components in your environment.

This topic describes how to configure an organization-specific LDAP user registry. If you want to create a shared registry, see [Configuring an LDAP user registry in the Cloud Manager](#) for more information.

Note:


- You can also create and manage LDAP user registries by using the developer toolkit CLI. For more information, see [Using the CLI to create an organization-specific LDAP user registry](#).
- The API Connect REST APIs can also be used to create and manage LDAP user registries; see the [API Connect REST API documentation](#).

You create an LDAP user registry by configuring a set of properties in the API Manager UI. If you want to enable writable LDAP, you must complete the Attribute Mapping section by selecting the User Managed checkbox, and providing the mapping of your source LDAP attribute names to the target API Connect values. You can also change a registry to be read-only again by clearing the User Managed checkbox. To make the registry available to the Developer Portal, you must define the registry for consumer onboarding in the associated Catalog. To secure APIs with an LDAP registry, you must configure security definitions.

For general information about authenticating with LDAP, see [LDAP authentication](#).

## Procedure

Follow these steps to configure a new LDAP user registry as a Resource in the API Manager UI.

1. In the API Manager, click  Resources.
2. Click Create in the User Registries section.
3. Select LDAP User Registry for the user registry type, and enter the following information:

Field	Description
Title	Enter a descriptive name to display on the screen.
Name	The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .
Display Name (required)	The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> .  Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> .
Summary (optional)	Enter a brief description.
Address	Enter the IP address or host name of the LDAP server.
Port	Enter the Port number that API Connect can use to communicate with the LDAP registry. For example, 389.
Select a TLS Client Profile (optional)	Select the TLS Client Profile the LDAP server requires.
Select a protocol version	Select the version number for the LDAP protocol that you are using.
Case-sensitivity	To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend LDAP server: <ul style="list-style-type: none"> <li>• <b>Only</b> select Case sensitive if your backend LDAP server supports case-sensitivity.</li> <li>• <b>Do not</b> select Case sensitive if your backend LDAP server does not support case-sensitivity.</li> </ul> Note: The Developer Portal does not support case sensitive usernames.  Note: After at least one user has been onboarded into the registry, you cannot change this setting.

4. Click Next and enter the authentication information, which will vary depending on the selected Authentication Method. The choices are:
  - Compose DN - Select this format if you can compose the user LDAP Distinguished Name (DN) from the user name. For example, `uid=<username>,ou=People,dc=company,dc=com` is a DN format that can be composed from the user name. If you are unsure whether Compose (DN) is the

correct option, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, Compose DN is not supported with the DataPower API Gateway.

- Compose UPN - Select this format if your LDAP directory supports binding with User Principal Names such as `john@acme.com`. The Microsoft Active Directory is an example of an LDAP directory that supports Compose UPN authentication. If you are unsure whether your LDAP directory supports binding with UPNs, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, Compose UPN is not supported with the DataPower API Gateway.
- Search DN - Select this format if you cannot compose the user LDAP Distinguished Name from the user name; for example, if the base DN of the users are different. This format might require an administrator DN and password to search for users in the LDAP directory. If your LDAP directory permits anonymous binds, you can omit the admin DN and password. If you are unsure if your LDAP directory permits anonymous binds, contact your LDAP administrator.

For all of the authentication methods:

If you are creating an LDAP registry to authenticate users of an API, you can specify an LDAP authorization group to restrict API access. To be able to call an API that is secured by the LDAP registry, a user must successfully authenticate with their LDAP user ID and password **and** they must be a member of the specified authorization group. The authorization group can be a Static Group or Dynamic Group. A static group is one in which the individual members of the group are explicitly listed. A dynamic group is one which is defined according to the set of attributes that the group members share in common.

5. For authentication method Compose DN, enter the following:

Field	Description
Bind Method	Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind.
Admin DN	For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> .
Admin Password	For Authenticated Bind, enter the user password for the Admin DN.
Prefix	Specify the prefix to the DN. For example <code>(uid=</code> .
Suffix	Specify the suffix to the DN. For example <code>)</code> .
Base DN (optional)	Enter a base DN in the Base DN field, or click Get Base DN to populate the field with a retrieved base DN.
Use group authentication (optional)	Static or Dynamic. For Static Group, enter the Group Based DN, Prefix, and Suffix. For Dynamic Group, enter the Filter condition for the group.

6. For authentication method Compose UPN, enter the following:

Field	Description
Bind Method	Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind.
Admin DN	For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> .
Admin Password	For Authenticated Bind, enter the user password for the Admin DN.
Suffix	Enter the domain part of the user principal name. For example, <code>@acme.com</code> .
Use group authentication (optional)	Enter the Filter condition for the group.

7. For authentication method Search DN, enter the following:

Field	Description
Bind Method	Anonymous or Authenticated. If specific permissions are not needed to search the registry, select Anonymous Bind. Or, if specific permissions are necessary, select Authenticated Bind.
Admin DN	For Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory. For example <code>cn=admin,dc=company,dc=com</code> .
Admin Password	For Authenticated Bind, enter the user password for the Admin DN.
Prefix	Specify the prefix to the DN. For example <code>(uid=</code> .
Suffix	Specify the suffix to the DN. For example <code>)</code> .
Base DN (optional)	Enter a base DN in the Base DN field, or click Get Base DN to populate the field with a retrieved base DN.
Use group authentication (optional)	Static or Dynamic. For Static Group, enter the Group Based DN, Prefix, and Suffix. For Dynamic Group, enter the Filter condition for the group.

8. Optional: Click Test configuration to test the settings for your LDAP user registry. Enter valid credentials to ensure that you can access the LDAP database.

9. Optional: If you want to make your LDAP user registry writable, select the User Managed checkbox in the Attribute Mapping section, and provide the mapping of your source LDAP attribute names to the target API Connect values. Click Add to add each name/value pair, specified as follows:

- LDAP ATTRIBUTE NAME - is the name of the source LDAP attribute.
- API CONNECT VALUE - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.

The default user profile properties that API Connect requires during user registration are `username`, `first_name`, `last_name`, `email`, and `password`, as shown in the following example:

LDAP ATTRIBUTE NAME	API CONNECT VALUE
<code>dn</code>	<code>uid=[username],ou=users,dc=company,dc=com</code>
<code>cn</code>	<code>{first_name} {last_name}</code>
<code>sn</code>	<code>{last_name}</code>
<code>mail</code>	<code>{email}</code>
<code>userPassword</code>	<code>{password}</code>

You must ensure that you enter the correct attribute mapping values for your LDAP configuration, to enable API Connect to access the LDAP database. Note that a writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.

10. Click Create.

Your new LDAP registry is shown in the list of User Registries on the Resources page.

## What to do next

If you want to make the LDAP user registry available for authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. Click the relevant Catalog, then click Settings > Onboarding. In the Catalog User Registries section, click Edit, select the user registry, and click Save. For more information, see [Creating and configuring Catalogs](#).

If you want to use the LDAP user registry to secure APIs, see the following information:

- To use for basic authentication in the security definition for an API, see [Creating a basic authentication security definition](#).
- To use for authentication in the User Security configuration for a native OAuth provider, see [Configuring user security for a native OAuth provider](#).
- [Using the CLI to create an organization-specific LDAP user registry](#)  
You can use the developer toolkit CLI to configure an organization-specific LDAP user registry to provide user authentication for the Developer Portal. APIs can also be secured with an LDAP user registry.

## Related information

---

- [Command-line tool reference for the developer toolkit](#)
- [Securing your API Connect Cloud with LDAP \(series of developer articles\)](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using the CLI to create an organization-specific LDAP user registry

You can use the developer toolkit CLI to configure an organization-specific LDAP user registry to provide user authentication for the Developer Portal. APIs can also be secured with an LDAP user registry.

### Before you begin

---

To configure an LDAP user registry as a resource in API Manager, the LDAP directory must be created and available to use with your API Connect ecosystem.

LDAP registries can be used to secure APIs, or for securing a Catalog to authenticate Developer Portal users.

Important: If you are using an LDAP registry to secure APIs, note the following limitations:

- Authentication methods `compose_dn` and `compose_upn` are not supported with the DataPower® API Gateway.
- The STARTTLS protocol, which upgrades an insecure protocol to a secure one by applying TLS security, is not supported with an LDAP user registry

One of the following roles is required to configure an LDAP user registry:

- Administrator
- Owner
- Topology Administrator
- Custom role with the `Settings: Manage permissions`

### About this task

---

You can create an LDAP user registry that is specific to a provider organization, or one that can be shared and available to all of the provider organizations in your API Connect environment. An organization-specific LDAP user registry can be used for onboarding and authenticating Developer Portal users in a specific provider organization. While a shared LDAP user registry can be used for authenticating Cloud Manager, API Manager, and Developer Portal users.

This topic describes how to configure an organization-specific LDAP user registry. If you want to create a shared registry, see [Using the CLI to configure a shared LDAP user registry](#) for more information.

Note:

- You can also create organization-specific LDAP user registries by using the API Manager UI; for more information see [Creating an LDAP user registry in API Manager](#).
- In addition, you can create and manage LDAP user registries by using the API Connect REST APIs; see the [API Connect REST API documentation](#).

You create an LDAP user registry by first defining the registry details in a configuration file. You then use a developer toolkit CLI command to create the registry, passing the configuration file as a parameter. To make the registry available to the Developer Portal, you must enable the registry in the associated Catalog. To secure APIs with an LDAP registry, you must configure security definitions. You can use the following instructions to create a writable or a read-only LDAP user registry.

For general information about authenticating with LDAP, see [LDAP authentication](#).

### Logging in to the management server CLI

---

Before you can define the LDAP user registry configuration, you must log in to your management server from the developer toolkit CLI as a member of a provider organization. Use the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the login command, see [Logging in to a management server](#).

For more information about how to use the CLI, see [Installing the toolkit](#), and [Overview of the command-line tool](#).

### Defining your LDAP configuration

---

You define the configuration of your LDAP user registry in an `ldap_config_file.yaml` file, as shown in the following example. Note that the actual contents of your YAML file will vary depending on the authentication method of your LDAP server, and this is explained in the following tables.

```
name: registry_name
title: "display_title"
```

```

integration_url: LDAP_integration_url
user_managed: true_or_false
user_registry_managed: false
case_sensitive: true_or_false
identity_providers:
- name: provider_name
  title: provider_title
endpoint:
  endpoint: "ldap_server_url_and_port"
configuration:
  authentication_method: authentication_method
  authenticated_bind: "true_or_false"
  admin_dn: "admin_dn"
  admin_password: admin_password
  search_dn_base: "search_dn_base"
  search_dn_scope: search_dn_scope
  search_dn_filter_prefix: prefix
  search_dn_filter_suffix: suffix
  attribute_mapping:
    dn: "distinguished_name"
    cn: "common_name"
    sn: "last_name"
    mail: "email_address"
    userPassword: "password"

```

The registry properties that are common to each authentication method are described in the following table:

Property	Description
<b>name</b>	The name of the registry. This name is used in CLI commands.
<b>title</b>	A descriptive name to display in a graphical user interface.
<b>integration_url</b>	The LDAP integration URL in your API Connect configuration. You can determine the LDAP integration URL by using the following CLI command: <code>apic integrations:list --server mgmt_endpoint_url --subcollection user-registry</code>
<b>user_managed</b>	Determines whether your user registry is writable or not. Must be set to <b>true</b> for writable LDAP. You can change this setting to <b>false</b> if you don't want the registry to be writable; see the Switching your LDAP registry between writable and read-only section at the end of this topic for details. Note that a writable LDAP user registry cannot be used to authenticate Cloud Manager and API Manager users.
<b>user_registry_managed</b>	Must be set to <b>false</b> for LDAP. Determines whether API Connect manages your user registry. Only LUR registries are managed by API Connect.
<b>case_sensitive</b>	Determines whether your user registry is case-sensitive. Valid values are: <ul style="list-style-type: none"> <li><b>true</b></li> <li><b>false</b></li> </ul> <p>To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend LDAP server:</p> <ul style="list-style-type: none"> <li><b>Only</b> set <b>case_sensitive</b> to <b>true</b> if your backend LDAP server supports case-sensitivity.</li> <li>Set <b>case_sensitive</b> to <b>false</b> if your backend LDAP server does not support case-sensitivity.</li> </ul> <p>Note: After at least one user has been onboarded into the registry, you cannot change this setting.</p>
<b>identity_providers</b>	An array containing the details of your LDAP server, where: <ul style="list-style-type: none"> <li><b>name</b> - is the name of the LDAP server and is the name that is used in CLI commands</li> <li><b>title</b> - is the display name of the LDAP server</li> </ul>
<b>endpoint</b>	The endpoint of your LDAP server, made up of the url and port, for example: <code>"ldap://server.com:389"</code>
<b>tls_profile</b>	Optionally set the TLS Client Profile that the LDAP server requires.
<b>protocol_version</b>	Optionally set the version number for the LDAP protocol that you are using. Valid values are: <ul style="list-style-type: none"> <li>2</li> <li>3</li> </ul> <p>Defaults to 3 if not explicitly set.</p>

The properties in the configuration section will vary depending on the selected authentication method. The three authentication methods are:

- compose\_dn** - Set this format if you can compose the user LDAP Distinguished Name (DN) from the user name. For example, `uid=<username>,ou=People,dc=company,dc=com` is a DN format that can be composed from the user name. If you are unsure whether Compose (DN) is the correct option, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, **compose\_dn** is not supported with the DataPower API Gateway.
- compose\_upn** - Set this format if your LDAP directory supports binding with User Principal Names such as `john@acme.com`. The Microsoft Active Directory is an example of an LDAP directory that supports Compose UPN authentication. If you are unsure whether your LDAP directory supports binding with UPNs, contact your LDAP administrator. If you are using an LDAP registry to secure APIs, **compose\_upn** is not supported with the DataPower API Gateway.
- search\_dn** - Select this format if you cannot compose the user LDAP Distinguished Name from the user name; for example, if the base DN of the users are different. This format might require an administrator DN and password to search for users in the LDAP directory. If your LDAP directory permits anonymous binds, you can omit the admin DN and password. If you are unsure if your LDAP directory permits anonymous binds, contact your LDAP administrator.

For authentication method **compose\_dn**, set the following configuration properties:

Properties	Description
<b>authentication_method</b>	<b>compose_dn</b>
<b>authenticated_bind</b>	The bind method. Valid values are: <ul style="list-style-type: none"> <li><b>"true"</b> - authenticated bind</li> <li><b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p>



Properties	Description
<b>admin_dn</b>	If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example:  "cn=admin,dc=company,dc=com"
<b>admin_password</b>	If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the user password for the <b>admin_dn</b> .
<b>search_dn_base</b>	Optionally set a base DN, for example:  "dc=company,dc=com"
<b>bind_prefix</b>	Set the prefix to the DN, for example:  (uid=
<b>bind_suffix</b>	Set the suffix to the DN, for example:  )
<b>attribute_mapping</b>	If <b>user_managed</b> is set to <b>true</b> , provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows:  <b>ldap_registry_attribute_name: "apic_ldap_attribute_value"</b>  Where: <ul style="list-style-type: none"> <li><b>ldap_registry_attribute_name</b> - is the name of the source LDAP attribute</li> <li><b>apic_ldap_attribute_value</b> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.</li> </ul> <p>The user profile properties that API Connect requires during user registration are <b>username</b>, <b>first_name</b>, <b>last_name</b>, <b>email</b>, and <b>password</b>. The following extract shows an example of an attribute mapping:</p> <pre>attribute_mapping: dn: "uid=[username],ou=users,dc=company,dc=com" cn: "[first_name] [last_name]" sn: "[last_name]" mail: "[email]" userPassword: "[password]"</pre>

For authentication method **compose\_upn**, set the following configuration properties:

Properties	Description
<b>authentication_method</b>	<b>compose_upn</b>
<b>authenticated_bind</b>	The bind method. Valid values are: <ul style="list-style-type: none"> <li><b>"true"</b> - authenticated bind</li> <li><b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p>
<b>admin_dn</b>	If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example:  "cn=admin,dc=company,dc=com"
<b>admin_password</b>	If <b>authenticated_bind</b> is set to <b>"true"</b> , enter the user password for the <b>admin_dn</b> .
<b>bind_suffix</b>	Enter the domain part of the user principal name. For example:  @acme.com
<b>attribute_mapping</b>	If <b>user_managed</b> is set to <b>true</b> , provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows:  <b>ldap_registry_attribute_name: "apic_ldap_attribute_value"</b>  Where: <ul style="list-style-type: none"> <li><b>ldap_registry_attribute_name</b> - is the name of the source LDAP attribute</li> <li><b>apic_ldap_attribute_value</b> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in [ ] with the value that the user supplies when signing up.</li> </ul> <p>The user profile properties that API Connect requires during user registration are <b>username</b>, <b>first_name</b>, <b>last_name</b>, <b>email</b>, and <b>password</b>. The following extract shows an example of an attribute mapping:</p> <pre>attribute_mapping: dn: "uid=[username],ou=users,dc=company,dc=com" cn: "[first_name] [last_name]" sn: "[last_name]" mail: "[email]" userPassword: "[password]"</pre>

For authentication method **search\_dn**, set the following configuration properties:

Property	Description
<b>authentication_method</b>	<b>search_dn</b>
<b>authenticated_bind</b>	The bind method. Valid values are: <ul style="list-style-type: none"> <li><b>"true"</b> - authenticated bind</li> <li><b>"false"</b> - anonymous bind</li> </ul> <p>If specific permissions are not needed to search the registry, select <b>"false"</b>. If specific permissions are necessary, select <b>"true"</b>.</p>

Property	Description
<code>admin_dn</code>	If <code>authenticated_bind</code> is set to <code>true</code> , enter the Distinguished Name (DN) of a user authorized to perform searches in the LDAP directory. For example: <code>"cn=admin,dc=company,dc=com"</code>
<code>admin_password</code>	If <code>authenticated_bind</code> is set to <code>true</code> , enter the user password for the <code>admin_dn</code> .
<code>search_dn_base</code>	Optionally set a base DN, for example: <code>"dc=company,dc=com"</code>
<code>search_dn_scope</code>	Optionally set the search DN scope. The scope determines which part of the directory information tree is examined. Possible values are: <ul style="list-style-type: none"> <li><code>base</code></li> <li><code>one</code></li> <li><code>sub</code></li> </ul>
<code>search_dn_filter_prefix</code>	Set the prefix to the DN, for example: <code>(uid=</code>
<code>search_dn_filter_suffix</code>	Set the suffix to the DN, for example: <code>)</code>
<code>attribute_mapping</code>	If <code>user_managed</code> is set to <code>true</code> , provide the mapping of your source LDAP attribute names to the target API Connect values. This mapping is configured as a name/value pair, specified as follows: <code>ldap_registry_attribute_name: "apic_ldap_attribute_value"</code> Where: <ul style="list-style-type: none"> <li><code>ldap_registry_attribute_name</code> - is the name of the source LDAP attribute</li> <li><code>apic_ldap_attribute_value</code> - is a string that represents the value that API Connect will populate the LDAP attribute with, by replacing the content contained in <code>[ ]</code> with the value that the user supplies when signing up.</li> </ul> The user profile properties that API Connect requires during user registration are <code>username</code> , <code>first_name</code> , <code>last_name</code> , <code>email</code> , and <code>password</code> . The following extract shows an example of an attribute mapping: <pre>attribute_mapping:   dn: "uid=[username],ou=users,dc=company,dc=com"   cn: "[first_name] [last_name]"   sn: "[last_name]"   mail: "[email]"   userPassword: "[password]"</pre>

Save your `ldap_config_file.yaml` so it can be accessed by the `user-registries:create` command in the following section. See the [Example](#) section for an example configuration file.

## Creating your LDAP user registry

To create your organization-specific LDAP user registry, run the following CLI command:

```
apic user-registries:create --server mgmt_endpoint_url --org organization_name ldap_config_file.yaml
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL.
- `organization_name` is the value of the `name` property of your provider organization.
- `ldap_config_file` is the name of the YAML file that defines the configuration of your LDAP user registry.

On completion of the registry creation, the command displays the following summary details:

```
registry_name registry_url
```

The `registry_name` is derived from the `name` property in the configuration YAML file. The `registry_url` is the URL with which the registry resource can be accessed. Your LDAP user registry is now created; see the following section for instructions on how to make the registry available in the Developer Portal.

## Configuring your LDAP registry in a Catalog

If you want to make your LDAP registry available for authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. Complete the following steps:

- Determine the URL of your LDAP user registry by using the following command (or you can copy and paste from the summary of the registry creation):

```
apic user-registries:list --server mgmt_endpoint_url --org organization_name
```

- Log in to the management server as a member of a provider organization; enter the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

- Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic configured-catalog-user-registries:create --server mgmt_endpoint_url --org organization_name --catalog catalog_name -
```

where `catalog_name` is the value of the `name` property of the required Catalog. The command returns

```
Reading CONFIGURED_CATALOG_USER_REGISTRY_FILE arg from stdin
```

- Enter the following data, followed by a new line:

```
user_registry_url: ldap_registry_url
```

- where `ldap_registry_url` is the URL of your LDAP registry, obtained in step 1.
5. Press **CTRL D** to terminate the input.

## Switching your LDAP registry between writable and read-only

After an LDAP user registry has been created, it can be switched between writable and read-only by updating the `user_managed` property in the registry configuration. Complete the following steps.

1. Determine the name or ID of the LDAP user registry that you want to update, by running the following command (or you can use the summary from the registry creation):

```
apic user-registries:list --server mgmt_endpoint_url --org organization_name
```

The command returns a list of all the user registries for that organization, shown by name followed by their registry URL. The registry ID is located at the end of the URL, for example `https://company.com/api/user-registries/x-x-x-x-x/registry_id`.

2. Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic user-registries:update --server mgmt_endpoint_url --org organization_name registry_name_or_id -
```

where `registry_name_or_id` is the name or ID of the LDAP user registry that you want to update (as determined in the previous step). The command returns:

```
Reading USER_REGISTRY_FILE arg from stdin
```

3. Enter the following data, followed by a new line:

```
user_managed: true_or_false
```

where `true` makes the registry writable, and `false` makes the registry read-only.

4. Press **CTRL D** to terminate the input.

Note that if you are changing your registry from read-only to writable, you must also set the `attribute_mapping` configuration, as described in the previous registry property tables.

## Using an LDAP user registry to secure APIs

If you want to use the LDAP user registry to secure APIs, see the following information:

- To use for basic authentication in the security definition for an API, see [Creating a basic authentication security definition](#).
- To use for authentication in the User Security configuration for a native OAuth provider, see [Configuring user security for a native OAuth provider](#).

For details of all the `apic user-registries` and `apic configured-catalog-user-registries` commands, see [apic user-registries](#) and [apic configured-catalog-user-registries](#).

## Example

The following example shows a configuration file that uses the Search DN authentication method for setting up writable LDAP:

```
name: sdn-ldap
title: "SDN LDAP User Registry"
integration_url: https://mycompany.com/api/cloud/integrations/user-registry/xxx-xxx-xxx
user_managed: true
user_registry_managed: false
case_sensitive: false
identity_providers:
- name: ldap
  title: "SDN LDAP Identity Provider"
endpoint:
  endpoint: "ldap://mycompany.com:389"
configuration:
  authentication_method: search_dn
  authenticated_bind: "true"
  admin_dn: "cn=admin,dc=company,dc=com"
  admin_password: xxxx
  search_dn_base: "dc=company,dc=com"
  search_dn_scope: sub
  search_dn_filter_prefix: (uid=
  search_dn_filter_suffix: )
  attribute_mapping:
    dn: "uid=[username],ou=users,dc=company,dc=com"
    cn: "[first_name] [last_name]"
    sn: "[last_name]"
    mail: "[email]"
    userPassword: "[password]"
```

## Related information

- [Command-line tool reference for the developer toolkit](#)
- [Securing your API Connect Cloud with LDAP \(series of developer articles\)](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Creating an Authentication URL user registry

An Authentication URL user registry provides a simple mechanism for authenticating users by referencing a custom identity provider.

## About this task

This topic describes how to create a new Authentication URL user registry as a Resource in your organization. After the user registry is created, the user registry must be added to the Sandbox catalog.

One of the following roles is required to configure user registries:

- Administrator
- Organization Owner
- Custom role with the `Settings: Manage permissions`

Note: When a user is presented with the form for completing their API Connect user registration, which fields are pre-populated depends on which fields are returned in the response from the Authentication URL identity provider. If any of the following fields are returned, they will be pre-populated in the registration form:

- `username`
- `email`
- `first_name`
- `last_name`

If the `username` field is not returned, the registration form displays the user name that was provided by the user. The pre-population capability requires that the response from the Authentication URL identity provider satisfies the following conditions:


- The `Content-Type` must be `application/json`.
- The response body format must be JSON.

A sample response is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "username": "myuser",
  "email": "myuser@example.com",
  "first_name": "My",
  "last_name": "User"
}
```

## Procedure

1. In the API Manager, click  Resources.
2. Select User Registries to see the list of current user registries in your organization.
3. Click Create in the User Registries section.
4. Select Authentication URL User Registry and enter the following parameters:

Field	Description
Title (required)	Enter a descriptive name to use on the screen.
Name (required)	The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .
Summary (optional)	Enter a brief description.
Display Name (required)	The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> . Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> .
URL (required)	Enter the URL for the authentication service. When establishing authentication, API Connect makes a GET call to the URL. The call includes the user name and password it has collected from the user in its authorization header. Either 200 OK or 401 Unauthorized will be returned.
TLS Client Profile (optional)	Select a TLS Client Profile to allow secure authentication with a specific web server.
Case sensitive	To ensure proper handling of user name capitalization, you <b>must</b> ensure that your case-sensitivity setting here matches the setting on your backend Authentication URL server: <ul style="list-style-type: none"><li>• <b>Only</b> select Case sensitive if your backend server supports case-sensitivity.</li><li>• Do <b>not</b> select Case sensitive if your backend server does not support case-sensitivity.</li></ul> Note: The Developer Portal does not support case sensitive usernames. Note: After at least one user has been onboarded into the registry, you cannot change this setting.

5. Click Save.
6. Add the user registry to the Sandbox catalog. See [Creating and configuring Catalogs](#).

## Results

The user registry can be used for Basic Authentication in the Security Definition for an API. For more information, see [Creating a basic authentication security definition](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a Local User Registry

A Local User Registry (LUR) can be created to provide user authentication for API Manager.

### About this task

---

Local User Registries (LURs) are the default user registries included in API Connect. LURs are local databases included with API Connect. Two default LURs are installed and configured during installation of API Connect. They cannot be deleted. The default Admin user account is stored in the Provider LUR.

You can use API Manager to create additional Local User Registries for use with your provider organization.


One of the following roles is required to configure user registries:

- Administrator
- Organization Owner
- Custom role with the `Settings: Manage permissions`

### Procedure

---

Follow these steps to configure a new LUR:

1. In the API Manager, click  Resources.
2. Select User Registries to see the list of current user registries in your organization.
3. Click Create in the User Registries section.
4. Select Local User Registry as the type for the user registry and enter the following information:

Field	Description
Title (required)	Enter a descriptive name for use on the screen.
Name (required)	The name that is used in CLI commands. The name is auto-generated. For details of the CLI commands for managing user registries, see <a href="#">apic user-registries</a> .
Display Name (required)	The name that is displayed for selection by the user when logging in to a user interface, or activating their API Manager account. For details of user interface log in, and account activation, see <a href="#">Accessing the Cloud Manager user interface</a> , <a href="#">Accessing the API Manager user interface</a> , and <a href="#">Activating your API Manager user account</a> . Note: The Developer Portal uses the <b>Title</b> of the User Registries when rendering them at the login page, rather than the <b>Display Name</b> .
Summary (optional)	Enter a brief description.
Case sensitive	Select this setting if user names are case-sensitive. Note: The Developer Portal does not support case sensitive usernames. Note: After at least one user has been onboarded into the registry, you cannot change this setting.

5. Click Save.
6. Add the user registry to the Sandbox Catalog. See [Creating and configuring Catalogs](#).

### Results

---

The user registry can be used for Basic Authentication in the Security Definition for an API. For more information, see [Creating a basic authentication security definition](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating an OIDC user registry

Create an organization-specific OIDC user registry when multi-factor authentication (MFA) is required.

You can create an OIDC user registry that is specific to a provider organization, or that is shared and available to all the provider organizations in your API Connect environment. An organization-specific OIDC user registry is used for onboarding and authenticating Developer Portal users, while a shared OIDC user registry can be used for onboarding and authenticating Cloud Manager, API Manager, and Developer Portal users.

This topic describes how to create an organization-specific registry. For information on how to create a shared registry, see [Configuring an OIDC user registry](#).


API Connect provides two methods for creating an OIDC user registry in API Manager, as described in the following sections:

- [Using the UI to create an OIDC user registry](#)
- [Using the CLI to create an OIDC user registry](#)

Note: Refresh tokens are not supported for a user in an OIDC user registry when accessing the Cloud Manager or API Manager user interfaces.

## Using the UI to create an OIDC user registry

Use the API Manager user interface to create an organization-specific OIDC user registry when multi-factor authentication (MFA) is required.

1. In the API Manager navigation pane, click  Resources.
2. Click User Registries.
3. Click Create and select OIDC User Registry.
4. On the Create OIDC User Registry page, use the fields in each of the following sections to configure the registry settings, and then click Create. Many of the registry settings are preconfigured to simplify the configuration steps.

### Provider Type

Use the settings in Table 1 to define the provider type.

Table 1. Provider Type settings

Field	Description
Provider Type	OIDC provider. Select one of the following supported OIDC providers: <ul style="list-style-type: none"> <li>• Facebook</li> <li>• GitHub</li> <li>• Google</li> <li>• LinkedIn</li> <li>• Slack</li> <li>• Twitter</li> <li>• Windows Live</li> <li>• Standard OIDC (default value allows you to specify another provider)</li> </ul>
Title	Provide a descriptive name for display purposes.
Name	Automatically generated. This name is used in CLI commands to reference the registry. For details of the CLI commands for managing user registries, see the <a href="#">apic user-registries</a> topic in the Command Line tool reference section of this documentation.
Summary	Provide a brief description of the new registry.

### Provider Endpoint

Automatically generated for most supported providers. In the Authorization Endpoint field, type the URL of the provider's authorization endpoint.

### Token Endpoint

Fill in the settings as described in Table 2.

Table 2. Token Endpoint settings

Field	Description
URL	Preconfigured for most of the supported OIDC providers. Type the URL of the provider's token endpoint.
TLS	Select the TLS Client Profile for the token endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile.

### UserInfo Endpoint

Fill in the settings as described in Table 3.

Table 3. UserInfo Endpoint settings

Field	Description
URL	Preconfigured for most of the supported OIDC providers. Type the URL of the provider's userinfo endpoint.
TLS	Select the TLS Client Profile for the userinfo endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile.

### JWKS Endpoint

Fill in the settings as described in Table 4.

Table 4. JWKS Endpoint settings

Field	Description
URL	Type the URL of the read-only endpoint that contains the public keys' information in JWKS format.
TLS	Select the TLS Client Profile for the userinfo endpoint (OIDC must be configured to use TLS). Default is Default TLS Client Profile.

### Client Information

Fill in the settings as described in Table 5.

Table 5. Client Information settings

Field	Description
Client ID	Provide the client ID of the application that is registered with the selected OIDC provider.
Client Secret	Provide the client secret of the application that is registered with the selected OIDC provider.
Response Type	Preconfigured for most of the supported OIDC providers. Specify the data type of the response that will be received from the OIDC provider.
Scopes	Preconfigured for most of the supported OIDC providers. Specify the access scope for the OIDC provider.
Client Authentication Method	Preconfigured for most of the supported OIDC providers. Select the authentication method to be used with the OIDC provider. Options are: <ul style="list-style-type: none"> <li>• Http basic authentication schema</li> <li>• Data encoded form body</li> </ul>

### Additional Support

Optional. Select the additional security parameters described in Table 6.

Table 6. Additional security options

Security parameter	Description
NONCE	Enable the NONCE extension to prevent compromised requests from being used again (replayed).
Proof Key for Code Exchange (PKCE)	Enable the PKCE extension to allow public clients to mitigate the threat of having the authorization code intercepted.

### Advanced Features

Optional. Select the advanced features described in Table 7.

Table 7. Advanced features

Feature	Description
---------	-------------

Feature	Description
Auto onboard	Allow users to execute calls to APIs without logging in first, provided they present a valid token issued by the OIDC provider.
Always use the userinfo endpoint	Configures the OIDC user registry to always fetch user data from the userinfo endpoint, if populated.
Return third-party access token	Include the third-party OIDC access token in the response.
Return third-party id_token	Include the third-party OIDC id_token in the response.

#### User Mapping

Fill in the settings as described in Table 8.

Note:



The User Mapping fields are preconfigured for most of the supported OIDC providers to minimize potential errors; use care when changing the settings. For the Standard OIDC option, contact your OIDC provider to obtain the details of the fields.

Table 8. User mapping settings

Field	Description
Username	The name of the field in the response token that contains the user's user name. Note: The username field must be unique for this OIDC registry, because it identifies the user in the system and cannot be changed.
Email	The name of the field in the response token that contains the user's email address.
First name	The name of the field in the response token that contains the user's given name.
Last name	The name of the field in the response token that contains the user's surname.

### Enabling the OIDC user registry

Complete the following steps to enable the new user registry for a specific catalog in the Developer Portal. Repeat this procedure for each catalog that will use the new registry.

1. On the navigation pane, click  Manage.
2. Select the catalog for which you will enable the new OIDC user registry.
3. On the navigation pane, click  Settings.
4. On the Settings page, click Onboarding.
5. On the Onboarding page, click Edit in the Catalog User Registries section.
6. On the Edit User Registry page, select the new OIDC user registry to enable for your catalog.
7. Click Save.

## Using the CLI to create an OIDC user registry

Use the developer toolkit CLI to create an organization-specific OIDC user registry when multi-factor authentication (MFA) is required.

You configure an OIDC user registry by first defining the registry details in a configuration file. You then use a CLI command to create the registry, passing the configuration file as parameter. To make the registry available to the Developer Portal, you must enable the registry in the associated Catalog.

Note:

- An OIDC registry, in common with a Local User Registry, cannot be used to secure APIs on the gateway.
- Because the interaction with the third party OIDC provider is handled by the Management server, the Management server is the application from the point of view of the third party OIDC provider. Your OIDC redirection endpoint, which is used by authorization server to send the token to the Management server, must be accessible to the OIDC provider through your firewall. When you register your application with the third party OIDC provider, you are required to supply the associated OIDC redirect URI, `https://consumer.mycompany.com/consumer-api/oauth2/redirect` for example. However, this information is not available until you have created your OIDC user registry in API Connect. You must therefore first register your application without this information, then update it later, as detailed in the instructions on this page.

### Logging in to the Management server

Before you can create an OIDC user registry, you must log in to your management server from the CLI. Use the following command:

```
apic login --server mgmt_endpoint_url --username user_id --password password --realm provider/identity_provider
```

For full details of the `apic login` command, see [Logging in to a management server](#).

### Defining your OIDC registry configuration

You define the configuration of your OIDC user registry in a YAML file. As a minimum, the YAML file must have the following content:

```
title: registry_title
integration_url: oidc_integration_url
case_sensitive: case_sensitivity_setting
configuration:
  client_id: 'app_client_id'
  client_secret: 'my-client-secret'
  provider_type: oidc_provider_type
```

where:

- `registry_title` is your chosen descriptive title for the user registry.
- `oidc_integration_url` is the OIDC integration URL in your API Connect configuration. You can determine the OIDC integration URL by using the following CLI command:

```
apic integrations:list --server mgmt_endpoint_url --subcollection user-registry
```

- `case_sensitivity_setting` determines whether your user registry is case-sensitive. Valid values are:
  - `true`
  - `false`

To ensure proper handling of user name capitalization, you **must** ensure that your case-sensitivity setting here matches the setting on the backend OIDC provider:

- **Only** set `case_sensitive` to `true` if the backend OIDC provider supports case-sensitivity.
- Set `case_sensitive` to `false` if the backend OIDC provider does not support case-sensitivity.

Note: After at least one user has been onboarded into the registry, you cannot change this setting.

- `app_client_id` is the client ID of the application that is registered with the OIDC server, and must be in string format.
- `my-client-secret` is the client secret of the application that is registered with the OIDC server, and must be in string format.
- `oidc_provider_type` is the type of OIDC provider; specify one of the following values:
  - `facebook`
  - `github`
  - `google`
  - `linkedin`
  - `slack`
  - `twitter`
  - `windows_live`
  - `standard`

Use the `standard` provider type for any OIDC provider that is compliant with the OIDC standard.

Note: If the provider type is `standard`, you must include the following additional properties in the `configuration` section of your YAML file:

```
authorization_endpoint: 'oidc_auth_endpoint'
token_endpoint:
  endpoint: 'oidc_token_endpoint'
```

where:

- `oidc_auth_endpoint` is the authorization endpoint on the OIDC server, and must be in string format.
- `oidc_token_endpoint` is the token endpoint on the OIDC server, and must be in string format.

## Default OIDC configurations

For each OIDC provider type, API Connect assumes a default configuration, but you can override the default configuration properties in your YAML file. The default configurations are as follows:

- Facebook

```
authorization_endpoint: 'https://www.facebook.com/v3.1/dialog/oauth'
token_endpoint:
  endpoint: 'https://graph.facebook.com/v3.1/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://graph.facebook.com/me'
scope: email public_profile
field_mapping:
  username: email
  email: email
  last_name: last_name
  first_name: first_name
```

- Github

```
authorization_endpoint: 'https://github.com/login/oauth/authorize'
token_endpoint:
  endpoint: 'https://github.com/login/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://api.github.com/user'
scope: 'read:user user:email'
field_mapping:
  username: login
  email: email
  last_name: name
  first_name: name
```

- Google

```
authorization_endpoint: 'https://accounts.google.com/o/oauth2/v2/auth'
token_endpoint:
  endpoint: 'https://www.googleapis.com/oauth2/v4/token'
scope: openid profile email
field_mapping:
  username: email
  email: email
  last_name: family_name
  first_name: given_name
```

- LinkedIn

```
authorization_endpoint: 'https://www.linkedin.com/oauth/v2/authorization'
token_endpoint:
  endpoint: 'https://www.linkedin.com/oauth/v2/accessToken'
userinfo_endpoint:
  endpoint: 'https://api.linkedin.com/v1/people/~:(id,first-name,last-name,picture-url,public-profile-url,email-address)?
format=json'
scope: r_basicprofile r_emailaddress
field_mapping:
  username: emailAddress
  email: emailAddress
  last_name: lastName
  first_name: firstName
credential_location: form_body
```

- Slack

```
authorization_endpoint: 'https://slack.com/oauth/authorize'
token_endpoint:
```



```

endpoint: 'https://slack.com/api/oauth.access'
userinfo_endpoint:
  endpoint: 'https://slack.com/api/users.identity'
  scope: identity.basic identity.email
field_mapping:
  username: user.email
  email: user.email
  last_name: user.name
  first_name: user.name

```

- Twitter:

```

request_endpoint: https://api.twitter.com/oauth/request_token'
authorization_endpoint: https://api.twitter.com/oauth/authenticate'
token_endpoint:
  endpoint: 'https://api.twitter.com/oauth/access_token'
userinfo_endpoint:
  endpoint: 'https://api.twitter.com/1.1/account/verify_credentials.json'
oauth_signature_method: 'HMAC-SHA1'
field_mapping:
  email: email
  first_name: name
  last_name: name
  username: screen_name

```

- WindowsLive

```

authorization_endpoint: 'https://login.microsoftonline.com/common/oauth2/v2.0/authorize'
token_endpoint:
  endpoint: 'https://login.microsoftonline.com/common/oauth2/v2.0/token'
scope: openid offline_access profile email
field_mapping:
  username: preferred_username
  email: email
  last_name: last_name
  first_name: first_name

```

- Standard

```

response_type: code
scope: openid
field_mapping:
  username: sub
  email: email
  last_name: family_name
  first_name: given_name
credential_location: auth_header

```

Although this is the default configuration for the **standard** provider type, you should contact your OIDC provider to obtain the details of the fields that you need to define.

## Creating your OIDC user registry

To create your OIDC user registry, use the following CLI command:

```
apic user-registries:create --server mgmt_endpoint_url --org organization_name oidc_config_file
```

where:

- *mgmt\_endpoint\_url* is the platform API endpoint URL.
- *organization\_name* is the value of the **name** property of your provider organization.
- *oidc\_config\_file* is the name of the YAML file that defines the configuration of your OIDC user registry.

On completion of the registry creation, the command displays the following summary details:

```
registry_name registry_url
```

By default, the *registry\_name* is derived from the **title** property in the configuration YAML file but you can override this by including a **name** property in the file. The *registry\_url* is an internal URL that API Connect assigns to the registry.

After you have created your OIDC user registry, you must update your application registration with the third party OIDC provider to include the OIDC redirect URI; you can obtain this information by using the following command, which displays the details of the registry in the command window:

```
apic user-registries:get --server mgmt_endpoint_url --org organization_name registry_name --output -
```

The required **oidc\_redirect\_uri** value is in the **consumer:** section; for example:

```

consumer:
  oidc_redirect_uri: https://consumer.mycompany.com/consumer-api/oauth2/redirect

```

## Enabling your OIDC registry in a Catalog

To make your OIDC registry available for onboarding and authenticating Developer Portal users, you must enable it in the Catalog that is associated with that Developer Portal. Complete the following steps:

1. Determine the URL of your OIDC user registry by using the following command:

```
apic user-registries:list --server mgmt_endpoint_url --org organization_name
```

2. Enter the following command (the terminating hyphen character means that the command takes input from the command line):

```
apic configured-catalog-user-registries:create --server mgmt_endpoint_url --org organization_name --catalog catalog_name -
```

where *catalog\_name* is the value of the **name** property of the required Catalog. The command returns

Reading CONFIGURED\_CATALOG\_USER\_REGISTRY\_FILE arg from stdin

3. Enter the following data, followed by a new line:

```
user_registry_url: oidc_registry_url
```

where `oidc_registry_url` is the URL of your OIDC registry, obtained in step 1.

4. Press **CTRL D** to terminate the input.

For details of all the `apic user-registries` and `apic`

`configured-catalog-user-registries` commands, see [apic user-registries](#) and [apic configured-catalog-user-registries](#).

You can also complete the operations described in this topic by using the API Connect REST APIs; see the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Modifying the configuration details for a user registry

You modify the configuration details for a user registry by using the User Registries page in the API Manager user interface of IBM® API Connect.

### About this task



If you want to authenticate an API Connect Catalog with a user registry that is not yet defined in API Connect, you define the registry when you configure the Catalog; for details, see [Working with user registries](#).

Note: User registries shared from the Cloud Manager cannot be edited.

To modify the configuration details for a user registry that is already defined, you use the User Registries page.

### Procedure

To modify the configuration details for a user registry, complete the following steps:

1. In the API Manager, click  Resources, and then select User Registries.
2. On the User Registries page, click the options icon  for the user registry that you want to edit, then click Edit.  
The details of the registry are displayed.
3. Modify the configuration details as required, then click Save to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Password lockout criteria

You can be locked out of your account if you attempt to log in and fail consecutively.

Note: Account lock out applies only for local user registries.

The length of time that you are locked out of using the account is based on the number of consecutive attempts that you fail. The length of time increases as the number of consecutive failed attempts increases.

For example, you can be locked out for 15 seconds if you have five consecutive failed attempts, or 32 minutes for 12 consecutive failed attempts.

Note: External user registries, such as LDAP, might enforce their own lockout criteria.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).


---

## Configuring a native OAuth provider

Native OAuth providers are configured and managed by you within your cloud.

### About this task

A native OAuth provider object provides settings for OAuth processing operations such as generating and validating OAuth tokens. An OAuth provider object is referenced by an OAuth security definition to protect an API. When a native OAuth provider is used, the OAuth operations are performed natively by API Connect.

Every OAuth provider object has a backing API. Your configuration here automatically updates the OpenAPI document of the API. You can edit the OpenAPI document directly by navigating to the  Resources > OAuth Providers page, selecting your OAuth provider, then clicking API Editor.

Note: Take care when modifying the code directly on the Source tab of the API Editor because validation is limited. For example:

- If you change the name of auto generated assembly actions in the source code, the assembly will be prevented from updating dynamically when the OAuth provider settings are modified.
- You must ensure that the OAuth provider name matches the value specified in the `oauth-provider-settings-ref` field in each OAuth assembly action.

When a published API references an OAuth provider object, the backing API is automatically made available in the gateway.


One of the following roles is required to configure a native OAuth provider:

- Organization Administrator
- Owner
- Custom role with the Settings > Manage permissions

Note:

- The OAuth provider logs Analytics data for failure cases, but does not log successful cases. Activity log policies that call for logging of Analytics data upon success do not apply for the OAuth provider.
- You must ensure the OAuth Provider is configured in the Sandbox Catalog before using the OAuth Provider in a non-Sandbox Catalog.

## Procedure

1. In the API Manager, click  Resources.

2. Click OAuth Providers > Add > Native OAuth provider.

a. Complete the following parameters for the first screen, then click Next.

Field	Description
Title	Enter a title for the native OAuth provider.
Name	This field is auto-populated by the system.
Description (optional)	Enter a brief description.
Base path (optional)	The base path is the URL segment of the API that is shared by all operations in the API. It does not include the host name or any additional segments for paths or operations. The base path must be unique for a given catalog. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.
Gateway Type	Select the gateway type, either DataPower® Gateway (v5 compatible) or DataPower API Gateway. For information about types of gateways, see <a href="#">API Connect gateway types</a> .  OAuth Providers apply to one gateway type.

b. In the next screen, enter the following additional configuration parameters, then click Next.

Field	Description
Authorize Path	<code>/oauth2/authorize/</code> is the standard OAuth endpoint to login to account
Token Path	<code>/oauth2/token/</code> is the standard OAuth endpoint to exchange code for access token.
Supported grant types	<ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type "Client Credentials." Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type "Authorization Code."</li> <li>• Resource owner - Password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> <li>• <span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Resource owner - JWT - A verified signed JSON Web Token is exchanged directly for an access token.</li> </ul>
Supported client types	<ul style="list-style-type: none"> <li>• Confidential - Client can maintain secure credentials on a secure server</li> <li>• Public - Client credentials are not secure. (Public is not available for DataPower API Gateway at this time.)</li> </ul>

c. Enter the scopes in the next screen. A scope becomes an option in the request and response for an access token. Click Add to add additional fields for scopes. Click Next when done.

Field	Description
sample_scope_1	Scope for token
additional scopes	Scope for token

d. Enter the parameters for User Security in the next screen. Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization. User Security is not required for the Application grant type. Click Next when done.

Field	Description
Identity Extraction	<p>Determines how the user credential is extracted:</p> <ul style="list-style-type: none"> <li>• Basic Authentication - HTTP basic authentication (requires no additional configuration)</li> <li>• Default HTML Form - Use default login form for user name and password</li> <li>• <span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Context variable - Specify which variable contains the user name and password. API Connect OAuth context variables as listed here <a href="#">API Connect context variables</a> Note: DataPower API Gateway only. Not available for DataPower Gateway (v5 compatible).</li> <li>• Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML login form for user security</a>.</li> <li>• Redirect - Enter an endpoint to redirect to a third-party identity provider. For more information, see <a href="#">Authenticating and authorizing through a redirect URL</a>.</li> <li>• <span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Disabled - do not collect the user credential</li> </ul> <p>Note: If you use either the Default HTML Form or Redirect identity extraction methods, the response from the redirect endpoint <b>must</b> maintain the order of the query parameters before the <code>state_nonce</code> query parameter, otherwise the authorization fails.</p>

Field	Description
Authentication	Authenticate application users with a user registry. Select an LDAP or Authentication URL user registry or create the SampleAuthURL User Registry. For a DataPower API Gateway, you have the option to disable authentication with a user registry.
Authorization	The following methods for extracting the user credential are available: <ul style="list-style-type: none"> <li>Authenticated - Authorize authenticated users automatically.</li> <li>Default HTML Form - Use default HTML form to authorize.</li> <li>Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML authorization form for user security</a>.</li> <li><span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Disabled - Disable authorization.</li> </ul>

- Review the Summary for the native OAuth provider configuration.
- Click Back to make changes.
- Click Finish to save the basic configurations and to proceed to the Advanced Parameters for a native OAuth Provider.

## Results

You can use the OAuth Provider to secure the APIs in catalog.

- [Configuring basic settings for a native OAuth provider](#)  
You can update the identification details and basic configuration settings for a native OAuth provider.
- [Configuring scopes for a native OAuth provider](#)  
Access tokens contain authorization for specific scopes.
- [Configuring user security for a native OAuth provider](#)  
Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization.
- [Configuring tokens for a native OAuth provider](#)  
Set time to live for access tokens and refresh tokens, and a time period for maximum consent for all tokens.
- [Configuring token management and revocation for a native OAuth provider](#)  
Select whether to use a native gateway (DataPower) or third party endpoint for token revocation.
- [Configuring introspection for a native OAuth provider](#)  
Define an introspection path to allow the metadata for an access token to be examined.
- [Configuring metadata for a native OAuth provider](#)  
Use Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.
- [Configuring the OIDC parameters for a native OAuth provider](#)  
Open ID Connect (OIDC) provides an additional authentication protocol based on OAuth 2.0. OIDC provides user information encoded in a JSON Web Token, or JWT.
- [Editing a native OAuth provider by using the API Editor](#)  
You can edit the source and assembly policies for the Native OAuth Provider using the API editor.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring basic settings for a native OAuth provider


You can update the identification details and basic configuration settings for a native OAuth provider.

### About this task

One of the following roles is required to configure the basic settings for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the Settings\_>Manage permissions

You can select the basis settings pages for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the basic settings for an existing native OAuth provider. If you want to update the basic settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

- Click  Resources > OAuth Providers.
- Select the required native OAuth provider.

### Procedure

- To modify the identification details, click Info in the sidebar menu, then update the following fields as required:

Field	Description
Title	Enter a title for the native OAuth provider.
Name	This field is auto-populated by the system.
Description (optional)	Enter a brief description.
Base path (optional)	The base path is the URL segment of the API that is shared by all operations in the API. It does not include the host name or any additional segments for paths or operations. The base path must be unique for a given catalog. The base path cannot include special characters and must begin with a "/" character even if it is otherwise empty.

2. To modify the basic configuration settings, click Configuration in the sidebar menu, then update the following fields as required:

Field	Description
Authorize Path	/oauth2/authorize/ is the standard OAuth endpoint to login to account
Token Path	/oauth2/token/ is the standard OAuth endpoint to exchange code for access token.
Supported grant types	<ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type "Client Credentials." Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type "Authorization Code."</li> <li>• Resource owner - Password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> <li>• <span style="border: 1px solid green; padding: 2px;">API Gateway only</span> Resource owner - JWT - A verified signed JSON Web Token is exchanged directly for an access token.</li> </ul> <p>Note: If you plan to configure OpenID Connect (OIDC) for a native OAuth provider, you must include at least one of the following grant types: Implicit, Access code.</p>
Supported client types	<ul style="list-style-type: none"> <li>• Confidential - Client can maintain secure credentials on a secure server</li> <li>• Public - Client credentials are not secure. (Public is not available for DataPower® API Gateway at this time.)</li> </ul>

DataPower Gateway (v5 compatible) only Note: If the gateway type is DataPower Gateway (v5 compatible) and, when the native OAuth provider was created, only the Application grant type was selected, you cannot add further grant types until you configure the user security settings. In particular, you must specify the user registry for authenticating application users. To configure the user security settings, complete the following steps:

- a. Click User Security in the sidebar menu, then click Edit.
- b. Update the user security settings as required; for more details, see [Configuring user security for a native OAuth provider](#).
- c. Click Save when done.

3. Click Save when done.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring scopes for a native OAuth provider

Access tokens contain authorization for specific scopes.

### About this task

The client applications can request only the scopes or a subset of the scopes that you define here. The scopes are included in the access tokens that are generated from the provider. When an OAuth protected API is invoked, the gateway checks the scopes carried in the access tokens against the list of allowed scopes in the security definition for the API to determine whether to grant access.


In addition, you can enforce advanced scope checks. The advanced scope check URLs are invoked after application authentication or after user authentication based on which URLs are configured. The final scope permission that is granted by the access token is the result of all scope checks.

Per IETF RFC 6749, the value of the scope parameter is a list of space-delimited, case-sensitive strings. For more information, see [The OAuth 2.0 Authorization Framework](#).

One of the following roles is required to configure scopes for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the Settings\_>Manage permissions

You can select the scope settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the scope settings for an existing native OAuth provider. If you want to update the scope settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

### Procedure

1. Click Scopes in the sidebar menu. The currently configured scopes are listed. Review and update the scopes as required.

Field	Description
sample_scope_1	Scope for token
additional scopes	Scope for token

In the Default scopes section, select the default scopes to be used if the API request doesn't contain any scopes.

If the user authorization method is set to Default HTML Form in the User Security settings, all scopes specified here are added automatically to the authorization consent form.

2. Advanced scope check before token generation. This setting specifies the scope check endpoint where additional scope verification is performed in addition to the basic scopes. The advanced scope check URLs are invoked after application authentication or after owner authentication based on which URLs are configured. The scopes are included in the token and will overwrite any previous scopes.

Field	Description
Application scope check	Allow extra verification by running a scope check from an endpoint. Enter the endpoint and an optional TLS Profile to use for an application scope check.

Field	Description
Owner scope check	Further refine the scope with an additional check. Enter the endpoint and an optional TLS Profile to use for an owner scope check.

For more information about scope, see [Scope](#)

- Advanced scope check after token generation. This setting specifies an additional scope check at the API consumer level to verify compliance with the scope requirements of the API.

Field	Description
Enabled	Select the check box to enable the advanced scope check after token validation. Enter an optional default validator endpoint.
Use endpoint from API	Select the check box to use the endpoint from the API, or clear the check box to override the endpoint from the API.

For more information about scope, see [Scope](#)

- Click Save when done.

## Results

You can use the OAuth Provider with these scopes to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring user security for a native OAuth provider

Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization.


### About this task

User security authenticates the user. It is required for the Implicit and Access Code (Authorization code) grant types.

One of the following roles is required to configure user security for a native OAuth Provider:




- Organization Administrator
- Owner
- Custom role with the Settings, Manage permissions

You can select the user security settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the user security settings for an existing native OAuth provider. If you want to update the user security settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

- Click  Resources, OAuth Providers.
- Select the required native OAuth provider.

### Procedure

- Click User Security in the sidebar menu.
- Specify the following parameters for User Security. Define the settings to use to extract the application users' credentials, authenticate their identities, and grant authorization. User Security is not required for the Client Credentials (Application) grant type. Click Next when done.

Field	Description
Identity Extraction	<p>Determines how the user credential is extracted:</p> <ul style="list-style-type: none"> <li>Basic Authentication - HTTP basic authentication (requires no additional configuration)</li> <li>Default HTML Form - Use default login form for user name and password</li> <li> Context variable - Specify which variable contains the user name and password. API Connect OAuth context variables as listed here <a href="#">API Connect context variables</a></li> <li>Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form. For instructions on creating a custom form, see <a href="#">Creating a custom HTML login form for user security</a>.</li> <li>Redirect - Enter an endpoint to redirect to a third-party identity provider. For more information, see <a href="#">Authenticating and authorizing through a redirect URL</a>.</li> <li> Disabled - do not collect the user credential</li> </ul> <p>Note: If you use either the Default HTML Form or Redirect identity extraction methods, the response from the redirect endpoint <b>must</b> maintain the order of the query parameters before the <b>state_nonce</b> query parameter, otherwise the authorization fails.</p>
Authentication	Authenticate application users with a user registry. Select an LDAP or Authentication URL user registry or create the SampleAuthURL User Registry.
Authorization	<p>Various methods may be used to authorize application users. For a DataPower® API Gateway, the following methods for extracting the user credential are available:</p> <ul style="list-style-type: none"> <li>Authenticated - Authorize authenticated users automatically.</li> <li>Default HTML Form - Use default HTML form to authorize. If you select the Default HTML Form method, all scopes that are specified in the Scopes settings are added automatically to the authorization consent form.</li> <li>Custom HTML Form - Enter the endpoint and select an optional TLS profile for a custom HTML form.</li> <li> Disabled - Disable authorization.</li> </ul>

- Click Save when done.

## Results

You can use the OAuth Provider to secure the APIs in catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring tokens for a native OAuth provider

Set time to live for access tokens and refresh tokens, and a time period for maximum consent for all tokens.


### About this task

Access tokens are granted to the client application to allow the application to access resources on behalf of the application user. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or more narrow scope. You can also specify how long the consent given by the combination of any number of access and refresh token remains valid.

One of the following roles is required to configure tokens for a native OAuth Provider:




- Organization Administrator
- Owner
- Custom role with the Settings > Manage permissions

You can select the token settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the token settings for an existing native OAuth provider. If you want to update the token settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

### Procedure

1. Click Tokens in the sidebar menu.
2. Define the settings to configure tokens.

Field	Description
Access tokens time to live	Enter the expiration time period in seconds for access tokens.
 One time use access token	Click the check box to enable one time use for the access token. Access tokens are multiple use by default which allows them to be used for multiple requests. When one time use is enabled, the access token will be consumed after one use. The OAuth flow will need to be repeated to obtain another access token. Note: If you select this option, you must also enable token management; see one of the following topics, depending on the user interface you are using: <ul style="list-style-type: none"><li>• <a href="#">Configuring token management and revocation for a native OAuth provider</a> (Cloud Manager)</li><li>• <a href="#">Configuring token management and revocation for a native OAuth provider</a> (API Manager)</li></ul>
Refresh tokens	Click the check box to enable Refresh tokens. Set the Count to limit the number of times a refresh token can be issued. Set the Refresh Token Time to Live value to determine the time to live, or expiration time period, for each refresh token in seconds.
One time use refresh token	Clear the check box to disable one time use for the refresh tokens. Refresh tokens are one time use by default which allows them to be used one time only to generate an access token and a new refresh token. When refresh token one time use is disabled then the refresh token count is limited to one and the refresh token can be used multiple times to generate new access tokens, however, another refresh token will not be generated unless the initial OAuth flow (Authorization Code or Password) is repeated. Note: If you select this option, you must also enable token management; see one of the following topics, depending on the user interface you are using: <ul style="list-style-type: none"><li>• <a href="#">Configuring token management and revocation for a native OAuth provider</a> (Cloud Manager)</li><li>• <a href="#">Configuring token management and revocation for a native OAuth provider</a> (API Manager)</li></ul>
Maximum consent	Click the check box to enable Maximum consent and enter the Maximum Consent Time to Live value in seconds. This is the time to live, or expiration time period, for all tokens, both access and refresh.
 Token secret	Click the check box to select the Shared Secret which was configured for the gateway. If no Shared Secret was entered in the Gateway Configuration, then enter an key name and key value to use as the token secret.
 Proof Key for Code Exchange	Proof Key for Code Exchange (PKCE) is a method to protect OAuth 2.0 public clients from an authorization code interception attack when they use Authorization Code grant requests. You can enable this extension when deploying with the DataPower® API Gateway. For more information, see <a href="#">RFC 7636</a> . Select the options for your OAuth Providers: <ul style="list-style-type: none"><li>• Enable proof key for code exchange If selected, enforces PKCE when submitted in Authorization Code grant requests.</li><li>• Always required If selected, requires PKCE in all Authorization Code grant requests.</li><li>• Allow plain Select this check box to allow the plain challenge method in Authorization Code grant requests.</li></ul>

3. Click Save when done.

## Results

---

You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring token management and revocation for a native OAuth provider

Select whether to use a native gateway (DataPower) or third party endpoint for token revocation.

### About this task

---


Token management enables you to prevent replay attacks by configuring token revocation. API Connect supports token revocation using a native gateway (DataPower) or a third party endpoint. For a native gateway, quota enforcement is used to manage tokens. For a third party endpoint, a URL to an external service is used to manage tokens.

For more information, see the IETF RFC 7009 [OAuth 2.0 Token Revocation](#).

One of the following roles is required to configure token management and revocation for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the Settings > Manage permissions

You can select the token management settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the token management settings for an existing native OAuth provider. If you want to update the token management settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

### Procedure

---

1. Select Token Management in the sidebar menu.
2. Enable Token Management by selecting the check box.
3. From the Type dropdown menu, select either Native or Third party. Native points to DataPower as the token storage location; Third party points to a revocation URL for token storage. For DataPower® API Gateway, only Native is supported.
4. For Native, select one or both of the Resource owner revocation path and Client revocation path.
  - Resource owner revocation path - Uses the standard OAuth revocation path to allow the resource owner (end user) to revoke the application permission.
  - Client revocation path - Uses the standard OAuth revocation path to allow the client (application) to revoke a single token when the application closes.For more information about managing tokens with the Native DataPower Gateway, see [Token management with the native DataPower Gateway](#).
5. For Third party, specify the Endpoint and TLS Client Profile.
  - Endpoint - Enter the URL to an external web server that contains information about access or refresh tokens. API Connect calls the URL to determine if the associated token can be trusted. The token server then checks a token *blacklist* (a data store of inactive tokens) to ensure that the token is still valid. If the token is still valid, API Connect continues the processing. For more information see [Token revocation](#).
  - TLS Client Profile - Select a TLS profile to verify the external endpoint.
6. Click Save when done.

## Results

---

You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring introspection for a native OAuth provider

Define an introspection path to allow the metadata for an access token to be examined.

### About this task

---


Token introspection allows an authorized holder of an access token to examine the contents of tokens using an introspection path. The access token to introspect must be obtained through the native OAuth provider. Introspection provides context for the token by allowing an authorized protected resource to query the authorization server to determine the set of metadata for a given token. The metadata includes whether or not the token is currently active, the scopes assigned to the token, and the authorization context in which the token was granted (including who authorized the token and which client it was issued to). API Connect token introspection conforms to IETF RFC 7662. See [OAuth 2.0 Token Introspection](#).

One of the following roles is required to enable introspection for a native OAuth Provider:



- Organization Administrator
- Owner
- Custom role with the Settings > Manage permissions

You can select the introspection settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the introspection settings for an existing native OAuth provider. If you want to update the introspection settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

1. Click Introspection in the sidebar menu.
2. Select the check box to enable Introspection. The OAuth standard path for introspection, /oauth2/introspect is automatically entered. This path will be used when another entity inspects the token contents.
3. Click Save when done.

## Results

Tokens will be queried using the /oauth2/introspect path. You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring metadata for a native OAuth provider

Use Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.

### About this task

Configure an Authentication URL or an External URL from which custom metadata is collected for inclusion in the token. The metadata is either stored inside the access token or it is sent along with the access token to the client application. For more information about how the metadata is collected, see [OAuth external URL and authentication URL](#).


Following are examples of metadata that can be included with the access token:

- Metadata about the authenticated resource owner
- Grant type that was used to obtain the token
- A confirmation code to be provided to the client application



One of the following roles is required to configure metadata collection for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the Settings > Manage permissions

You can select the metadata settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the metadata settings for an existing native OAuth provider. If you want to update the metadata settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

1. Click Metadata in the sidebar menu.
2. Select Collect metadata to enable metadata collection.
3. The Authentication URL user registry is selected by default and is required. For more information about the Authentication URL, see [Authentication URL user registry](#).
4. Select the External URL to collect metadata from an external URL. Enter the endpoint and an option TLS Client Profile.
5.  If required, override the default Header name token value. The value of this header, if returned in the response from the OAuth endpoint, is placed in the response payload and indicated as **metadata**.
6.  If required, override the default Header name payload value. The value of this header, if returned in the response from the OAuth endpoint, is placed within the access token and indicated as **miscinfo**.
7. Save the OAuth Provider.
8. Click Save when done.

## Results

You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Configuring the OIDC parameters for a native OAuth provider

Open ID Connect (OIDC) provides an additional authentication protocol based on OAuth 2.0. OIDC provides user information encoded in a JSON Web Token, or JWT.

### About this task


When you enable OpenID connect, a template is provided for generating ID tokens along with access tokens and the required assembly policies are automatically created. You can customize the policies to suit your needs in the API Editor. The sample key provided is for test purposes only and is used to sign the JWT token.

One of the following roles is required to configure an OIDC template for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the System\_>Manage permissions



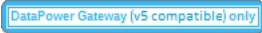


Note: You can configure OIDC parameters only if the selected grant types for the native OAuth provider include at least one of the Implicit or Access code grant types; see [Configuring basic settings for a native OAuth provider](#).

You can select the OIDC settings page for a native OAuth provider immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the OIDC settings for an existing native OAuth provider. If you want to update the OIDC settings for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources > OAuth Providers.
2. Select the required native OAuth provider.

### Procedure

1. Click OpenID Connect in the sidebar menu.
2. Select the initial check box to configure an OIDC Template. Enter the following parameters:

Field	Description
 Support hybrid response types (optional)	Select the response types for the OpenID Connect hybrid flow to be supported by this OAuth provider.
 Auto Generate OIDC API Assembly	Select this option to generate the full OIDC assembly. Leave this option unselected to simply enable OIDC support in the OAuth provider and allow the developer to implement their own assembly.
 ID token issuer	Descriptive text to indicate the source of the key.
 ID token signing key	Specify the JSON Web Key (JWK) to be used to sign the ID token.
 ID token signing algorithm	Select the algorithm used to sign the token.

3. Click Save when done. You can edit the policies by using the API Editor.

### Results

You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Editing a native OAuth provider by using the API Editor

You can edit the source and assembly policies for the Native OAuth Provider using the API editor.

### About this task

If you have configured an OIDC template, you can customize it in API Editor. In the API Editor, the Source tab allows you to edit the code for the configuration using a text editor. The API Assemble tab provides a graphical drag-and-drop editor (identical to the one in API Manager) that allows you to add additional elements to the assembly for the OAuth Provider.


Note: Take care when modifying the code directly on the Source tab of the API Editor because validation is limited. For example:

- If you change the name of auto generated assembly actions in the source code, the assembly will be prevented from updating dynamically when the OAuth provider settings are modified.
- You must ensure that the OAuth provider name matches the value specified in the `oauth-provider-settings-ref` field in each OAuth assembly action.

One of the following roles is required to configure tokens for a native OAuth Provider:

- Organization Administrator
- Owner
- Custom role with the Settings, Manage permissions

You can modify the native OAuth provider configuration by selecting the API Editor page immediately on completion of the creation operation detailed in [Configuring a native OAuth provider](#), or you can update the configuration for an existing native OAuth provider. If you want to update the configuration for an existing native OAuth provider, complete the following steps before following the procedure described in this topic:

1. Click  Resources, OAuth Providers.
2. Select the required native OAuth provider.

## Procedure

1. Click API Editor in the sidebar menu.
2. In the Source tab, view and edit the policies to customize the behavior for the OAuth provider.
3. In the API Assemble tab, use the drag and drop editor to add additional policies to the OIDC behavior.
 

Note: If you add a policy that references a TLS profile, an `invoke` policy for example, then when you publish an API that uses this OAuth provider, you must ensure that the TLS profile is enabled for the Catalog to which you publish the API. For details on how to enable a TLS profile in a Catalog, see [Creating and configuring Catalogs](#).
4. Save the edits.
5. Click Save when done.

## Results

You can use the OAuth Provider to secure the APIs in a catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Configuring a third-party OAuth provider


Enter the secure endpoints to provide OAuth authentication from a third party.

## About this task

One of the following roles is required to configure OAuth Providers:

- Organization Administrator
- Owner
- Custom role with the Settings, Manage permissions

## Procedure

1. In the API Manager, click  Resources.
2. Select OAuth Providers, Add, Third party OAuth Provider.
  - a. Complete the following parameters for the first screen and click Next.

Field	Description
Title	Enter a descriptive title for the gateway service. This title will be displayed on the screen.
Name	This field is auto-populated by the system and used as the internal field name.
Supported grant types	<ul style="list-style-type: none"> <li>• Implicit - An access token is returned immediately without an extra authorization code exchange step.</li> <li>• Application - Application to application. Corresponds to the OAuth grant type Client Credentials. Does not require User Security.</li> <li>• Access code - An authorization code is extracted from a URL and exchanged for an access code. Corresponds to the OAuth grant type Authorization Code.</li> <li>• Resource owner password - The user's username and password are exchanged directly for an access token, so can only be used by first-party clients.</li> </ul>
Gateway type	Select the gateway type, either DataPower® Gateway (v5 compatible) or DataPower API Gateway. For information about types of gateways, see <a href="#">API Connect gateway types</a> .  OAuth Providers apply to one gateway type.

- b. Specify configuration settings for the endpoints.

Field	Description
Authorization URL	An authorization URL where the resource owner grants authorization to the client application to access a protected resource. Example:  <code>https://example.com/oauth2/authorize</code>

Field	Description
Token URL	A token request URL where the client application exchanges an authorization grant for an access token. Example: <code>https://example.com/oauth2/token</code>
Introspect URL	The introspection URL is where the API gateway validates the access tokens that are issued by the third party provider. Example: <code>https://example.com/oauth2/introspect</code>  For more information on integrating third party OAuth providers for introspection, see <a href="#">OAuth introspection for third-party OAuth providers</a> .
TLS Profile (optional)	Select an optional TLS profile for communicating with the third party provider.
Security	Default is Basic Authentication.
<span style="border: 1px solid green; border-radius: 5px; padding: 2px;">DataPower Gateway (v5 compatible) only</span> Basic authentication request header name	The <code>x-introspect-basic-authorization-header</code> is available to provide a user-configured HTTP Basic authorization header.
<span style="border: 1px solid green; border-radius: 5px; padding: 2px;">API Gateway only</span> Basic authentication request header name (optional)	The name of the header that will provide a user-configured HTTP Basic authorization. The default value is <code>x-introspect-basic-authorization-header</code> .
<span style="border: 1px solid green; border-radius: 5px; padding: 2px;">API Gateway only</span> Basic authentication username (optional)	The default user name for HTTP Basic authentication.
<span style="border: 1px solid green; border-radius: 5px; padding: 2px;">API Gateway only</span> Basic authentication password (optional)	The default password for HTTP Basic authentication.
<span style="border: 1px solid green; border-radius: 5px; padding: 2px;">API Gateway only</span> Custom header pattern (optional)	A regular expression for request headers that are to be passed to the third-party provider; for example, <code>x-Introspect-*</code> .

c. Enter the scopes in the third screen. A scope becomes an option in the request and response for an access token. Click Add to add additional fields for scopes. Click Next when done.

Field	Description
sample_scope_1	Scope for token
sample_scope_2	Scope for token
additional scopes	Scope for token

d. Review the settings on the Summary panel.

3. Click Save and Edit to complete the configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## OAuth concepts for API Connect

OAuth is a token-based authorization protocol that allows third-party websites or applications to access user data without requiring the user to share personal information.

Note: In a multi-node cluster, OAuth operations will fail if quorum is lost. Quorum requires that the number of active nodes is greater than 50% of the total number of nodes in the cluster.

- [OAuth user scenario](#)  
Potential users of OAuth with IBM® API Connect have a number of methods to secure their API. The following scenario provides an overview of the available options.
- [OAuth introspection for third-party OAuth providers](#)  
OAuth token validation can be offloaded to the third-party Open Authorization (OAuth)/Open ID Connect (OIDC) provider by using the Introspection URL. Clients can use a third-party OAuth or OIDC provider to obtain a token that is protected by IBM API Connect. IBM API Connect can use this feature along with the mentioned provider to protect access to the API.
- [OAuth external URL and authentication URL](#)  
You can use the Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.
- [Scope](#)  
Per IETF RFC 6749, the value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings.
- [Tokens](#)  
Tokens can be managed using refresh tokens and revocation URLs. You can extend the life of tokens using refresh tokens. You can end the life of a token by specifying a revocation URL.
- [Authentication URL user registry](#)  
You can use an Authentication URL user registry to specify a REST authentication service that manages user authentication, and optionally provides additional metadata to be embedded in the token.
- [Custom forms for user security](#)  
You can create custom forms for the authorization and identity extraction phase of OAuth.
- [Securing an API with a JSON Web Token](#)  
There are two methods to secure your API with a JSON Web Token. You can use the `jwt-generate` command, or you can use a token that has been generated external to IBM API Connect.
- [Troubleshooting OAuth](#)  
You can find the answers to some common questions in the Developer Portal, or in support communities and forums in DeveloperWorks, on GitHub, or on Youtube.

## Related information

---

- [The OAuth 2.0 Framework](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## OAuth user scenario

Potential users of OAuth with IBM® API Connect have a number of methods to secure their API. The following scenario provides an overview of the available options.

### Scenario overview

---

In this scenario, Alice is a user of an application. Alice can grant permission for an application to access specific information about Alice in a third-party system that is using OAuth. Depending on the type of OAuth that is supported by the target service, Alice does not enter her user name and password into the application. Instead, the application receives an access token that represents her credentials (user name and password). The application can now access the information about Alice in the target system.

For example, Alice maintains a list of books that she reads on a service that is provided by mybooks.com. Following the purchase of a smartphone, Alice installs an application on her new phone to display the book details. The phone application wants to call an API provided by mybooks.com, which can access the information. The mybooks.com API is secured by using the OAuth protocol.

To access the book details, the application must complete a two-step process:

1. The application must first obtain permission from Alice.
2. The application then uses that permission to call the target service and obtain the list of books.

In the first step, the application typically directs Alice to the provider of the target service, mybooks.com. Alice provides her user name and password, and gives permission for the application to access her information. It is important that Alice trusts that she is providing her credentials to the provider of the target service and not to an untrusted proxy application. For example, by checking that the security certificate of the website where Alice enters her credentials matches what Alice expects from the provider of the target service.

The result of this step is the access token that the application can use to call the API. The application then generates the appropriately formatted OAuth request. For example, the Authorization header, or HTTP query parameters, which includes the access token, consumer key, and signature method that are required by OAuth. This OAuth request is used to invoke the API proxy operation.

### Scenario within API Connect

---

No changes to the definition of your API operation are required to support this scenario.

1. Alice grants permission for the application to access her information *before* the invocation of the API.
2. When the application provides the Authorization header, or query parameters, containing the OAuth details about the call to the operation endpoint, the header is automatically passed through to the target service without any additional configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

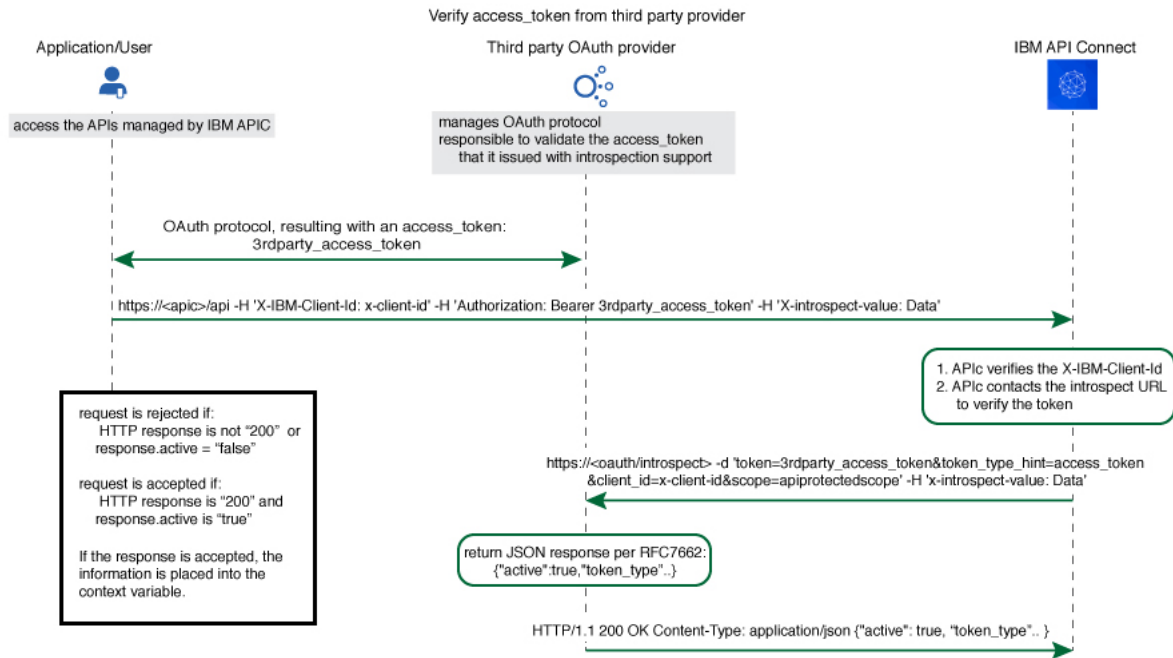
## OAuth introspection for third-party OAuth providers

OAuth token validation can be offloaded to the third-party Open Authorization (OAuth)/Open ID Connect (OIDC) provider by using the Introspection URL. Clients can use a third-party OAuth or OIDC provider to obtain a token that is protected by IBM API Connect. IBM® API Connect can use this feature along with the mentioned provider to protect access to the API.

You can use API Connect to protect an API that is secured by using the third-party OAuth access token in accordance with Introspection specification as defined in [RFC 7662](#). In addition to the specification, the `x-Introspect-` header is provided to pass other content to the third party as you require.

Authentication information can be carried in the request by configuring a basic authentication request header.

The following sequence diagram depicts the overall flow of request and response. The purpose of this diagram is only to provide a general visualization of the nature of the flow; for the precise details, refer to the explanatory information after the diagram.



The Introspect URL is configured in the Third Party OAuth provider configuration. See [Configuring a third-party OAuth provider](#).

When an API is protected by a third-party OAuth provider, API Connect will extract the bearer token and issue an HTTP POST request to the endpoint specified in the Introspect URL field.

The GET request is protected by API Connect with this feature.

You can use a header prefix to pass information to the third-party provider. The header prefix can include a regular expression and specifies the name pattern of the headers to use for sending additional information, such as `x-introspect-*`.

```
GET /petstore/pet/123 HTTP/1.1
Host: apiconnect.com
x-Introspect-type: dog
x-Introspect-name: simon
x-custom-apic: petstore123
Authorization: Bearer tGzv3JOkF0XG5Qx2TlKWIA
x-IBM-Client-Id: xxx-xxx
```

However, if you want to use a different header prefix, specify the required value in the Custom header pattern field in the third party OAuth provider configuration. For third-party OAuth provider configuration details, see [Configuring a third-party OAuth provider](#). The Custom header pattern feature is available only with the DataPower® API Gateway, if you are using the DataPower Gateway (v5 compatible) the header prefix must be `x-Introspect-`.

API Connect will issue this POST request to the introspection endpoint specified in `x-tokenIntrospect`, as illustrated in the code sample:

```
POST /oauth/introspectURL HTTP/1.1
Host: apiconnect.ibm.com
Content-Type: application/x-www-form-urlencoded
x-Introspect-type: dog
x-Introspect-name: simon
x-custom-apic: petstore123

token_type_hint=access_token&token=tGzv3JOkF0XG5Qx2TlKWIA
```

The third party OAuth/OIDC provider will respond with `HTTP 200` indicating the request was successful and that payload contains the token information. API Connect honors the active claim as defined in the RFC specification.

If the value of the active claim is `true`, the token is treated as valid.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "active": true,
  "token type": "bearer",
  "client id": "xxx-xxx",
  "username": "John Smith",
  ...
}
```

If the value of the active claim is `false`, the token is treated as invalid.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
```

```
"active":false
}
```

A response code other than **HTTP 200** indicates failure to process the request.

When the OAuth token is valid and active, context variables are populated with information from the introspect JSON response. For more information, see [API Connect context variables](#).

When contacting the introspection endpoint, API Connect uses `client_id/appId` and `client_secret/appSecret` to construct the HTTP Basic authorization header.

By default, if `x-introspect-basic-authorization-header` exists in the request, the value is used for the HTTP Basic authentication header when the introspection endpoint is contacted. API Connect verifies that the HTTP Basic authentication header value is Base64 encoded before it is sent, and encodes it if necessary as shown in the following example. If the header is already encoded, it is sent without modification.

```
GET /petstore/pet/123 HTTP/1.1
Host: apiconnect.com
x-introspect-basic-authorization-header: user:password
```

API Connect issues the following:

```
POST ..
Authorization: Basic M3JkLXBhcncR5LWNsaWVudF9pZDozcmQtcGFydHl1fy2xpZW50X3NlY3JldA==token_type_hint=access_token&token= ..
```

If you are using the DataPower API Gateway, you can specify your own value for the HTTP Basic authentication header by providing the required value in the Basic authentication request header name field in the third party OAuth provider configuration. The following rules determine what authentication data is passed to the introspection endpoint:

- If there is a basic authentication header in the request, the specified credentials are used. The value must be either a string in the format `user:password`, or the Base64 encoded equivalent; API Connect will Base64 encode the value if necessary before sending the request to the introspection endpoint.
- If there is no basic authentication header in the request but there are values specified in the Basic authentication username and Basic authentication password fields in the third party OAuth provider configuration, those values are used.
- If there is no basic authentication header in the request, and user name and password details are not supplied in the third party OAuth provider configuration, but `client_id` and `client_secret` values are supplied in the body of the request, these are used.
- Otherwise, an error is returned.

For third-party OAuth provider configuration details, see [Configuring a third-party OAuth provider](#).

If either, or both, of `scope` and `scope validate url` are configured, and if the response is an active token with a scope claim from the third-party OAuth Provider introspection endpoint, API Connect would further enforce the scope validation in the following order:

1. If `scope` is configured for the OAuth API protection, verify the third-party scope against the scope that is configured.
2. If `scope validation url` is configured, verify the third-party scope against the scope validation url.

For more information, see [Scope](#).

**DataPower Gateway (v5 compatible) only**

By default the API Connect client ID and scope are sent to the third party OAuth provider. You can suppress this behavior in either of the following ways:

- Supply a `suppress-parameters` header as follows:

```
suppress-parameters: client_id
```

```
suppress-parameters: scope
```

or

```
suppress-parameters: client_id scope
```

depending on which parameters you want to suppress.

- Define an API property called `suppress-parameters` in the API definition itself, with one of the following string values:

```
client_id
```

```
scope
```

or

```
client_id scope
```

depending on which parameters you want to suppress. For information on how to define API properties, see [Setting API properties](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## OAuth external URL and authentication URL

You can use the Authentication URL or External URL parameters to request user-defined content from a remote server and include it in the access token or in the response payload that contains the access token.

### Including custom metadata in a token

In many scenarios, custom metadata needs to be included during the access token generation process. The metadata is either stored inside the access token or it is sent along with the access token to the client application. The client application can then send that access token, or the metadata in the payload, in a subsequent request to

IBM® API Connect where the metadata is retrieved, validated, or sent to the downstream systems as required.

Examples include, but are not limited to:

- When resource owners are authenticated, metadata about the authenticated resource owner needs to be stored within the access token.
- The grant type that was used to obtain the token is another example of metadata stored within the access token.
- A confirmation code that needs to be sent to the client application along with access token is stored as metadata within the access token.

## Configuring External URL or Authentication URL in API Connect to obtain metadata

Metadata can be set by using either or both of the following URLs:

- External URL - When you call the External URL, an HTTP GET request is sent and API Connect expects an HTTP 200 OK along with an optional set of the specified response headers.
- Authentication URL - When you call the Authentication URL, the API Connect gateway sends a GET request with HTTP headers and then processes any HTTP response from the URL. For authentication, a REST authentication service is expected at the Authentication URL.  
See: [Authentication URL](#).

The External URL endpoint is entered in the Metadata section when you configure an OAuth Provider in the Cloud Manager or API Manager UI.

Use the following HTTP headers in the response, depending on the type of gateway you are using:

- DataPower® API Gateway:

**X-API-OAUTH-METADATA-FOR-PAYLOAD**

**X-API-OAUTH-METADATA-FOR-ACCESSTOKEN**

Note: These are the default header names for the DataPower API Gateway but you can override them; see [Configuring metadata for a native OAuth provider](#).

- DataPower Gateway (v5 compatible):

**API-OAUTH-METADATA-FOR-PAYLOAD**

**API-OAUTH-METADATA-FOR-ACCESSTOKEN**

The response header value from **X-API-OAUTH-METADATA-FOR-PAYLOAD** or **API-OAUTH-METADATA-FOR-PAYLOAD** is placed in the response payload and indicated as `metadata`.

The response header value from **X-API-OAUTH-METADATA-FOR-ACCESSTOKEN** or **API-OAUTH-METADATA-FOR-ACCESSTOKEN** is placed within the access token and indicated as `miscinfo`.

The two metadata response headers are case insensitive and you must escape any special characters in the string value content.

An example response payload that contains metadata along with the access token:

```
{
  "token_type": "bearer",
  "access_token": "AAEkNzhjDHYyyYy...cL0Mv6ct137w7ZU",
  "metadata": "m:metadata-for-payload_content",
  "expires_in": 3600,
  "scope": "read",
  "refresh_token": "AAEnj5SynCMYbF...oEZ6JjxYax_HdNg",
}
```

This example output from the token introspection endpoint shows the contents of the access token with `miscinfo` containing the metadata information.

```
{
  "active": true,
  "token_type": "bearer",
  "client_id": "78c2f10f-799a-4e1f-8e0a-098634997a35",
  "username": "Fred Smith",
  "sub": "fred",
  "exp": 1479850049,
  "expstr": "2016-11-22T21:27:29Z",
  "iat": 1479846449,
  "nbf": 1479846449,
  "nbfstr": "2016-11-22T20:27:29Z",
  "scope": "read",
  "miscinfo": "m:metadata-for-accesstoken_content",
  "client_name": "MobileApp"
}
```

## Input to the External URL

The following request headers are sent to the External URL.

Note: Any existing metadata values that were previously sent from the Authentication URL are also sent in the two request headers **x-existing-metadata-for-payload** and **x-existing-metadata-for-access-token**. The Metadata URL can make use of this information to create a new set of metadata values. The two request headers that are sent to the Metadata URL are displayed in bold text.

```
X-existing-metadata-for-payload    payload-from-auth-url
X-existing-metadata-for-access-token    token-from-auth-url
X-URI-in      /org/env/miscinfo/oauth2/token (the URL that was sent to APICoconnect for this particular token request)
X-METHOD-in    POST
X-POST-Body-in  client_id=client_id&grant_type=password&scope=read&username=name&password=password
X-X-Client-IP    IP_address
X-X-Global-Transaction-ID    ID_number
...
```



Note: If you are using the DataPower API Gateway rather than the DataPower Gateway (v5 compatible), the **X-POST-Body-in** header is not supported.

## Retrieving Metadata in API Connect

As described in the previous example scenarios, the metadata can be retrieved from the access token in the Application API and sent to the downstream systems. Retrieval can be done in the API assembly, which is secured to accept tokens in the security definitions.

In the resource API that accepts an access token, the `miscinfo` field can be accessed in the Assembly with the `oauth.miscinfo` context variable, as in the example.

```
apim.setvariable('message.body', apim.getvariable('oauth.miscinfo'));
```

You can also use token introspection to look at the contents of the access token.

## Refresh tokens and metadata

The Authentication URL (if configured for authentication) is invoked first, during authentication of the resource owner. The External URL is invoked as the last step, just before the generation of the access token. The only exception is when access tokens are generated from a refresh token. In cases where refresh tokens are used to generate new access tokens, the External URL is not invoked. The metadata from the refresh token is retained and then copied into the newly generated access token.

## Identifying the source of the metadata

The metadata is prefixed with keywords to indicate whether it originated from the External URL or the Authentication URL.

- Metadata from the External URL is prefixed with `m`:
- Metadata from the Authentication URL is prefixed with `a`:

Note: When revocation is enabled, some internal details are also stored in the `miscinfo` field, in square-brackets within the access token as shown in the following example:

```
"miscinfo": "[tlsprofile@https://api-revoke-url:443/server/revocation-url]m:metadata-for-accesstoken_content"
```

## Maximum size of the metadata

Metadata for the access token cannot exceed 512 bytes.

Metadata for the payload does not have a specific size restriction, except for when you use the Authorization code grant type. These restrictions are described in following sections.

## Characters are not allowed in metadata in certain scenarios

When you use the Authorization code grant type, or when a consent form is used for implicit grant type, there is a temporary state or code where the metadata from the authentication URL is stored. API Connect internally uses two prefixes - `!ma` and `!mp` to differentiate between payload and token metadata received from the Authentication URL and store them internally in the temporary state/code. Hence these specific character sequences - `!ma` and `!mp` should not be used as the metadata itself.

## Grant types and metadata

The OAuth grant types described in the following sections include `authorization code` (`access code`), `implicit`, and `client credentials` (`application`).

Authorization code grant type

- When metadata is included from an Authentication URL for an Authorization code grant type, as it is a three legged flow, both the content and the payload are stored within the `dp-state` and carried on to the authorization code and to the access token. Note that around 10 characters are used internally to differentiate between the metadata for payload and metadata for the access token when stored in the `dp-state`. In addition, if revocation is enabled, that will also be part of the token metadata. Hence the combined size of the token metadata, the payload metadata (including the 10 characters of internal data), and internal revocation details, cannot exceed 512 bytes in total.

If the overall size of the metadata exceeds 512 bytes, then the access token generation succeeds, but the metadata fields contain an error message of "metadata too large" as shown in the example.

```
"metadata": "m:error: metadata too large for AZ code grant type[Authorization Code-metadata-url-payload]"
"miscinfo": "[r:gateway]m:error: metadata too large for AZ code grant type[Authorization Code-metadata-url-token]"
```

This size restriction can be overcome when the metadata is sent from the Metadata URL and not from the Authentication URL, because the metadata is not stored in `dp-state` or in the authorization code.

Example of the `authorization code` (`access code`) grant type:

```
$ curl -k -d
"grant_type=authorization_code&code=$mycode&client_id=$myid&scope=scope_introspect&redirect_uri="
https://9.70.153.90/fei/sb/introspectpl/oauth2/token
{ "token_type": "bearer",
  "access_token": "AAEkOThlZDhhNjYtYTQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ2Z0Zn5Tl_TqYFeIfB7BFf6HFgibEoOjWXXEA84oFsWuE4NY-
RRZVdnGSaXAIYJz7s5vczfk5EV-BIb_6P_1YKm3ahrfrR5kI3sPO0uADEoseIP5-O9anUpEM5yhSayXvZbJ_6VDYz-hyXSJHTNqVj-
PHBialoRkBD5qca6k00fv2M", "metadata": "a:[Authorization Code-Test-auth-url-payload]", "expires_in": 3600,
  "scope": "scope_introspect", "refresh_token": "AAFAG1EVMbwicr_L0fTZ4q6HZ-
RcQygniXFC9zbSKO4wd3hcnlC4RQX21X0fL2c8cnmzCZgws8xxLzNyfjQhUJNG15C1GbTe3dwhXJdiWA0Go-
dduhVtCbG26sJRRXyYrMeRwWnJsy1BETPI8HQEN4a_D7fmxKcTVRZBvq86byg95qe1ZKYERi0Lhxdd_04Nvss" }
```

```
$ curl -k -H "X-IBM-Client-Id: $myid" -H "X-IBM-Client-Secret: $mysecret" -d
"token_type_hint=access_token&token=$mytoken" https://9.70.153.90/fei/sb/introspectp1/oauth2/introspect
{ "active":true, "token_type":"bearer", "client_id":"98ed8a66-a45e-4a33-a47d-fa68fd344746", "username":"anyuser",
"sub":"anyuser", "exp":1484766368, "expstr":"2017-01-18T19:06:08Z", "iat":1484762768, "nbf":1484762768,
"nbfstr":"2017-01-18T18:06:08Z", "scope":"scope_introspect", "miscinfo":{"r:gateway}a:[Authorization Code-Test-auth-
url-token]", "client_name":"oauth_app" }
```

Implicit grant type

- When implicit grant type is used, the access token and the metadata are returned in the `Location` header as a fragment, as you see in the example.

```
< Location:
https://localhost#access_token=AAEkOTh1ZDhhNjYtYTQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ2buS2KfWdq-
nYBJSi4nmPxQBtLael7tKBPRMzwP5BC386nlxpoOTE1G748ZVH6Mq_TJL3GeV3PtXTVIkWOBJi_7t1jiQfnpVfrNkovvZkhUexYmFkcDsmLSdaWxZ6Pe
IMPAC4ojT8qV1sYV-
ChTk36yqOx_NiKimZaDikDk7WTg&expires_in=3600&scope=scope_introspect&token_type=bearer&metadata=a%3A[Implicit-Test-
auth-url-payload]
```

```
$ curl -k -H "X-IBM-Client-Id: $myid" -H "X-IBM-Client-Secret: $mysecret" -d
"token_type_hint=access_token&token=$mytoken" https://9.70.153.90/fei/sb/introspectp1/oauth2/introspect
{ "active":true, "token_type":"bearer", "client_id":"98ed8a66-a45e-4a33-a47d-fa68fd344746",
"username":"anyuser", "sub":"anyuser", "exp":1484768365, "expstr":"2017-01-18T19:39:25Z", "iat":1484764765,
"nbf":1484764765, "nbfstr":"2017-01-18T18:39:25Z", "scope":"scope_introspect", "miscinfo":{"r:gateway}a:[Implicit-
Test-auth-url-token]", "client_name":"oauth_app" }
```

Client credentials grant type

Authentication URL will not be invoked when using client credentials grant type, as there is no resource owner. The metadata from Authentication URL is not available for this grant type. However, content returned from Metadata URL will be included as metadata.

## Behavior when retrieving metadata with both an External URL and an Authentication URL

If an External URL is configured and a connection to the external server is successful, the response headers overwrite any existing metadata obtained from the Authentication URL to become the final value. Therefore, you must carefully examine the incoming request headers and create appropriate response headers from the External URL.

If an External URL is configured, but the connection to the External URL fails, then a failure message of "error on metadata url" is written for metadata in both the payload and the access token.

If an External URL is configured and the connection is successful, but the remote server does not send any of the specified HTTP response headers, a blank value is written for metadata in both the payload and the access token.

Attention: An External URL overwrites existing values from Authentication URL. This includes blank values.

If no External URL is configured, the metadata that is obtained from the Authorization URL is retained as the final value.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Scope

Per IETF RFC 6749, the value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings.

In IBM® API Connect, scopes have no inherent meaning. Instead, scopes are defined in the OAuth Provider so that an application can request an access token that is valid for one or more of the scopes. In the secured API, scopes are listed as requirements for an access token to be considered valid. All scopes that are listed by the security definition for the API must be granted by the access token.

## OAuth provider

To provide more refined support for the OAuth scope handling, API Connect allows the [Authentication URL user registry](#) extension to modify the scope value.

When you define an [OAuth provider](#), the Advanced scope check extensions provide the flexibility to check and override allowed scopes. The optional extensions are Application scope check and Owner scope check.

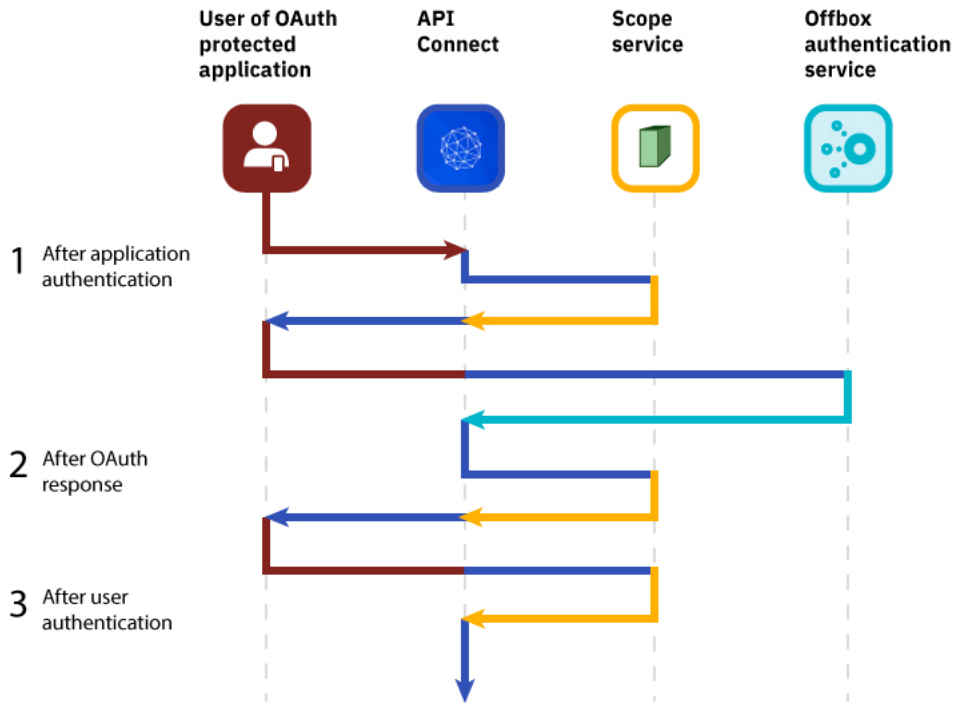
The scope that is eventually received by the application is determined by the interactions that are described in the three following paragraphs. Scope processing follows the sequence of paragraphs 1, 2, and 3 in order, offering three opportunities to override the scope value. Figure 1 provides an overview of the process.

1. After the application successfully authenticates, and if OAuth Native provider `>Advanced scope check>` Application scope check is configured, API Connect makes a call to allow extra verification and uses the contents of `x-selected-scope` to override the scope value that was initially requested by the application. When Application scope check is enabled, the HTTP response header `x-selected-scope` must be present, or the call fails.

2. If OAuth Native provider `>User Security>` Authentication is configured to authenticate application users using an Authentication URL, then API Connect makes a call, as documented in [Authentication URL user registry](#). When the response code is HTTP 200, and the response header `x-selected-scope` is present, the value that is configured in `x-selected-scope` is used as the new scope value, overriding both what the application already requested and what was provided in the Application scope check described in paragraph 1. In the response header, `x-selected-scope` is an optional element.

3. After the user successfully authenticates, and if OAuth Native provider `>Advanced scope check>` Owner scope check is enabled and configured with a valid URL, API Connect makes a call to allow the content of `x-selected-scope` to refine the scope value. When Owner scope check is enabled, the HTTP response header `x-selected-scope` must be present, or the call fails.

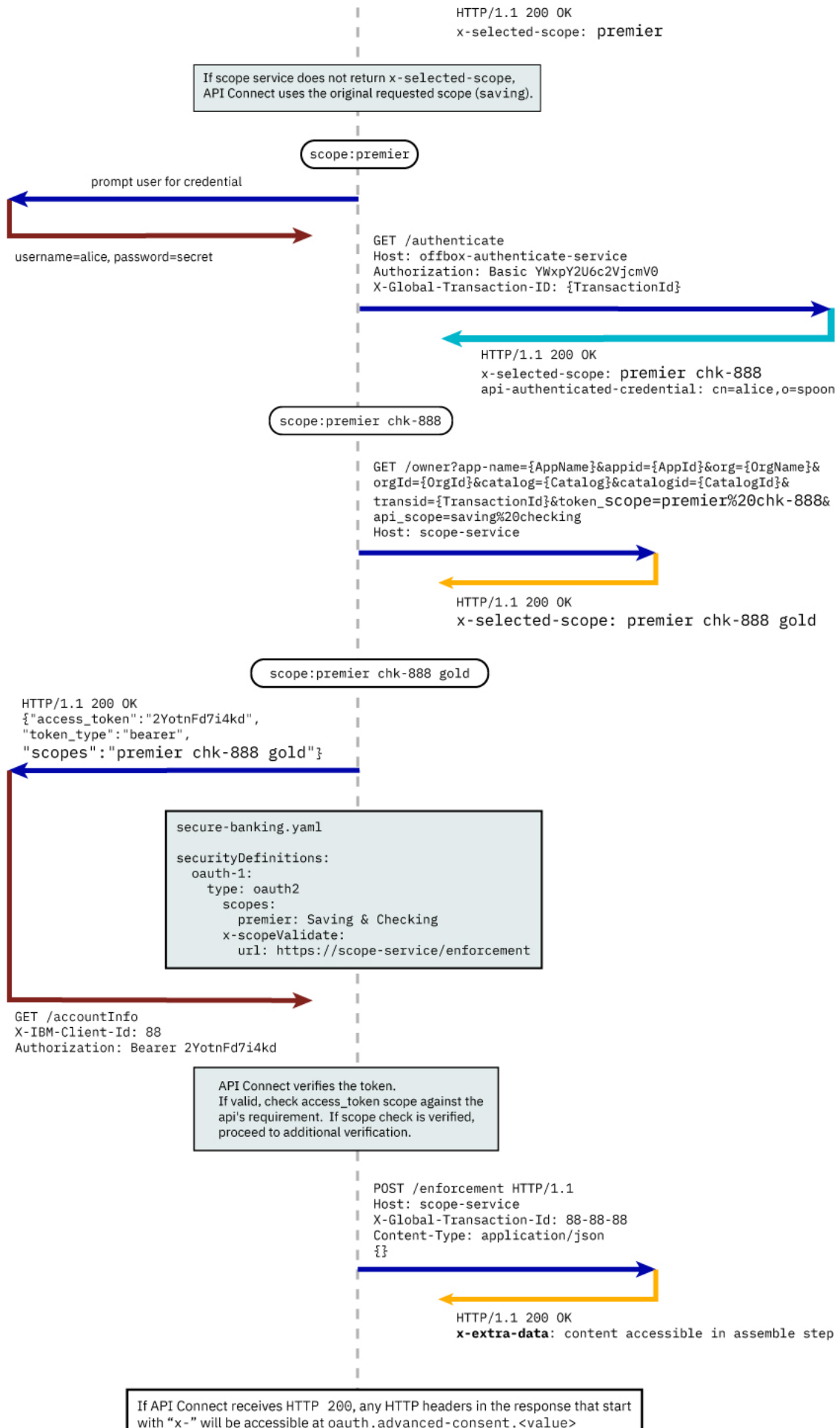
Figure 1. OAuth advanced scope overview



The final scope permission that is granted by the access token is the result of the flow described in paragraphs 1, 2, and 3. [Figure 2](#) shows a more detailed view of the transaction flow with examples that show when `x-selected-scope` provides a new scope value.

Figure 2. OAuth advanced scope detail





For the above example, `x-extra-data` is accessible at `oauth.advanced-consent.x-extra-data`.  
 If other than HTTP 200 is returned, it has the same effect as if the access token does not contain necessary permissions to access the resource.

## Consumer API enforcement

Standard scope validation

To access the API `/getaccount` the application must send a **GET** request with an access token that contains the scope, or scopes, defined in the OAuth provider.

```
GET /getaccount
HTTP/1.1
Host: server.example.com
X-IBM-Client-Id: 8888-8888-8888
Authorization: Bearer AAEkNjVkwOWIyYjgtOWY5ZS00YWQwLWlWYyZktZ
```

The following application OpenAPI file `secure-banking.yaml` defines the scope, or scopes, that must exist in the token to be granted access to the API `/getaccount`.

```
secure-banking.yaml

securityDefinitions:
  scope-only:
    type: oauth2
    description: ''
    flow: implicit
    authorizationUrl: ''
    scopes:
      checking: 'Checking Account'
      saving: 'Saving Account'
      mutual: 'Mutual Fund Account'
  security:
    - scope-only:
      - checking
    - scope-only:
      - saving
      - mutual
```

In the examples, the access token `AAEkNjVkwOWIyYjgtOWY5ZS00YWQwLWlWYyZktZ` is able to access the API because it contains one, or a combination of `scope-only` that is defined in `secure-banking.yaml` such as:

- `checking`
- `saving mutual`
- `checking saving mutual`

Advanced Scope Check

Administrators can enable an additional scope check by configuring the consumer API property Advanced Scope Check URL that becomes `x-scopeValidate` as shown in the following OpenAPI file example.

```
securityDefinitions:
  advanced-scope-only:
    type: oauth2
    description: ''
    flow: implicit
    authorizationUrl: ''
    scopes:
      checking: 'Checking Account'
      saving: 'Saving Account'
      mutual: 'Mutual Fund Account'
  x-scopeValidate:
    url: 'https://advanced-scope-check.bk.com/validate-scope'
    tls-profile: 'ssl-client'
```

After API Connect successfully verifies the access token against any scope requirement, API Connect will make an **HTTP POST** request to the `x-scopeValidate` endpoint similar to the following example. Response code `HTTP 200` from the endpoint indicates a success. Any other response code, or a connection error, is treated as a permission error for the token.

```
POST /validate-scope?app-name=..&appid=..&org=..&orgid=..&catalog=..&catalogid=..&transid=..
HTTP/1.1
Host: advanced-scope-check.bk.com
Content-Type: application/json
```

```
{
  "context-root" : checking,
  "resource" : accountinfo,
  "method" : GET,
  "api-scope-required" : [jointaccount],
  "access_token" : {
    "client_id" : "2cd71759-1003-4a1e-becb-0474d73455f3",
    "not_after" : 174364070,
    "not_after_text" : "2017-07-11T02:27:50Z",
    "not_before" : 174360470,
    "not_before_text" : "2017-07-11T01:27:50Z",
    "grant_type" : "code",
    "consented_on" : 1499736470,
    "consented_on_text" : "2015-07-11T01:27:50Z",
    "resource_owner" : "cn=spoon,email=spoon@poon.com",
    "scope" : "jointaccount mutual",
    "miscinfo" : "[r:gateway]"
  }
}
```

An example of successful response follows.

```
HTTP/1.1 200 OK
Cache-Control: no-store
Pragma: no-cache
X-Custom-For-Assemble-Process: audit
```

Any HTTP response header that begins with "x-" is kept as the context variable `oauth.advanced-consent`. Based on the example successful response, `X-Custom-For-Assemble-Process: audit` becomes `oauth.advanced-consent.x-custom-for-assemble-process`, and can be accessed in the assemble step.

Additional validation options

You can optionally use additional fields for validation:

Request Header

Defines the regular expression to match against **request** headers. Matching headers are included in the request to the advanced scope validation endpoint.

Response Context Variable

Defines the regular expression to match against **response** headers. Matching headers are saved as context variables in the format `oauth.advanced-consent.*`.

## Related information

---

- [IETF RFC 6749](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tokens

Tokens can be managed using refresh tokens and revocation URLs. You can extend the life of tokens using refresh tokens. You can end the life of a token by specifying a revocation URL.

This section contains information regarding token management, including refreshing and revoking tokens, redirecting, and managing tokens with the DataPower Gateway.

- [Refresh tokens](#)  
If you are using OAuth authentication, you can enable refresh tokens. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope.
- [Token revocation](#)  
In IBM® API Connect, you can revoke or refresh tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.
- [Authenticating and authorizing through a redirect URL](#)  
You can use a service that is hosted externally from IBM API Connect to collect authentication and authorization details from your user when an application requests access on that user's behalf.
- [Token management with the DataPower Gateway \(v5 compatible\)](#)  
API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.
- [Token management with the DataPower API Gateway](#)  
API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Refresh tokens

If you are using OAuth authentication, you can enable refresh tokens. Refresh tokens are issued to the client to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope.

When you are using OAuth authentication, API requests must include a valid access token, by using the Authorization HTTP header. Access tokens that are issued by the IBM® API Connect Token Endpoint are valid for 3600 seconds (1 hour) by default, as indicated by the `expires_in` property that is returned on the token request. The following code block shows an example API request with an Authorization header:

```
GET /bankingApi/accountSummary?client_id=32427ce5-bb7c-48a7-9de3-4bb629091103
HTTP/1.1
Accept: application/json
Host: api.ibm.com
Authorization: Bearer AAEFYy1hbGx1hdS5nVX4x6iTL2sb3ymBivQb...
```

After an access token expires, if the option is enabled in the OAuth provider configuration Tokens > Refresh tokens screen, the application uses refresh tokens. Each refresh token is valid for approximately 31 days after it is issued (or for the Time to Live time period specified) and can be used only once to request a new access token. Along with the new access token, a new refresh token is also returned. For details on how to enable refresh tokens, see [Configuring a native OAuth provider](#).

If the access token is expired and the application does not have a refresh token, it must restart the OAuth exchange by using the choice of Grant Type(s) allowed by the OAuth provider.

Note: Refresh tokens are not supported for a user in an OIDC user registry when accessing the Cloud Manager or API Manager user interfaces.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Token revocation

In IBM® API Connect, you can revoke or refresh tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.

---

## Token revocation using an external third party service

This topic describes token revocation using a third party, external service. This option is configured in the Native OAuth provider, using the Token Management, Type = Third party screen. The revocation URL is an endpoint that links to an external service which contains information about access or refresh tokens. API Connect is involved in the initial creation and validation of tokens. When an OAuth revocation URL is present, API Connect calls the URL to determine if the associated token can be trusted. The token server then checks a token *blacklist* (a data store of inactive tokens) to ensure that the token is still valid. If the token is still valid, API Connect continues the processing.

---

## Examples

A number of revocation examples follow. The first shows a sample fetch request and the response from a remote revocation URL.

---

## GET request and response

In this example, an API Gateway server issues a GET request to the Revocation URL and receives a result. It shows that different resource owners (Laura and Emily) can revoke all tokens when using the same application (client ID).

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
GET <revocationURL> HTTP/1.1
User-Agent: IBM-APIManagement/4.0
Accept: application/xml; text/xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>

<!--
Access Tokens and/or Refresh Tokens that are revoked can be individually listed.
To keep this list small, please only include access tokens and refresh tokens that are valid.
For access tokens, any token older than 20 minutes is no longer valid.
For refresh tokens, any token older than 44700 minutes is no longer valid.
-->

<token type="access">AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPScl...</token>
<token type="refresh">fZaRlVbnPSclUGTjCRdq4mPbOosD2+aZIKbJ6bTeW...</token>

<!--
If a resource owner has revoked all tokens issued to a given application, please
list them as shown here.
-->

<resource-owner client-id="760d75a2-44b1-4485-8c6f-0d264fcf7398">laura</resource-owner>
<resource-owner client-id="760d75a2-44b1-4485-8c6f-0d264fcf7398">emily</resource-owner>

</oauth-revocation>
```

---

## POST request and response

The next example shows a post request and response.

Request:

```
POST <revocationURL> HTTP/1.1
User-Agent: IBM-APIManagement/4.0
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<token>
  <token_type>bearer</token_type>
  <access_token>AAENY1hbGwtcmVmcmVzaOfNeQKX8ZeojsBY9v0FI7/OerQvzKHq...</access_token>
  <expires_in>3600</expires_in>
  <scope>3600</scope>
  <resource-owner>alice</resource-owner>
  <client_id>83d9cdcd-ba72-4d00-abae-005da8da5fb1</client_id>
</token>
```

Response:

```
HTTP/1.1 200 OK
```

---

## Provide token information on revocation request

In this example, the application calls an API and passes a bearer token. In response, the Gateway fetches the revocation URL and provides information on the token being verified.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Alternatively, the same process occurs when using a refresh token to issue a new access token. The application sends a refresh request to the token service. The Gateway then fetches the revocation URL, providing information on the refresh token being verified.

Gateway:

```
GET <revocationURL>GET HTTP/1.1
refresh-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Note: If you are using a third party OAuth provider then, for API Calls with bearer tokens, when Introspect URL is enabled on the API, the revocation URL is not applicable. Instead, the third party endpoint must validate the token and also check for revocation before returning a 200 OK response to the Gateway.

## Revoking tokens issued to Alice up to and including a specific date

On May 1st, Alice loses her phone and needs to reset her password. As a result, the token provider wants to revoke every token issued before Alice lost her phone. In this example, the Gateway sends a GET request to the revocation service. The revocation service replies confirming revocation of the specified tokens.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>
  <resource-owner before="2015-05-01T09:30:10Z">alice</resource-owner>
</oauth-revocation>
```

## Revoking all tokens issued up to and including a specific date

Under certain catastrophic conditions, you may need to revoke all tokens issued up to and including a specific date, for example, May 1st. In this example, the Gateway sends a GET request to the revocation service. The revocation service replies confirming revocation of the specified tokens.

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>
  <everytoken before="2015-05-01T09:30:10Z" />
</oauth-revocation>
```

## Putting it all together

The following shows the examples contained in this topic executed in a single action:

Gateway:

```
GET <revocationURL>HTTP/1.1
access-token: AAETb2F1dGgtcmV2b2t1LWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Cache-Control: public, max-age=120
Date: Fri, 08 May 2015 21:49:03 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<oauth-revocation>
  <resource-owner before="2015-04-08T09:30:10Z">mary</resource-owner>
  <resource-owner before="2015-04-12T09:30:10Z">john</resource-owner>
  <resource-owner before="2015-04-13T09:30:10Z">kevin</resource-owner>
  <resource-owner before="2015-04-01T09:30:10Z">alice</resource-owner>
</oauth-revocation>
```



Notes:

- In the previous example, there are no entries older than one month in the response (the maximum life of a refresh token).
- Each response is cached for up to two minutes according to response's directive.
- The **before** attribute uses the **xs:dateTime** syntax.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authenticating and authorizing through a redirect URL

You can use a service that is hosted externally from IBM® API Connect to collect authentication and authorization details from your user when an application requests access on that user's behalf.

### Before you begin

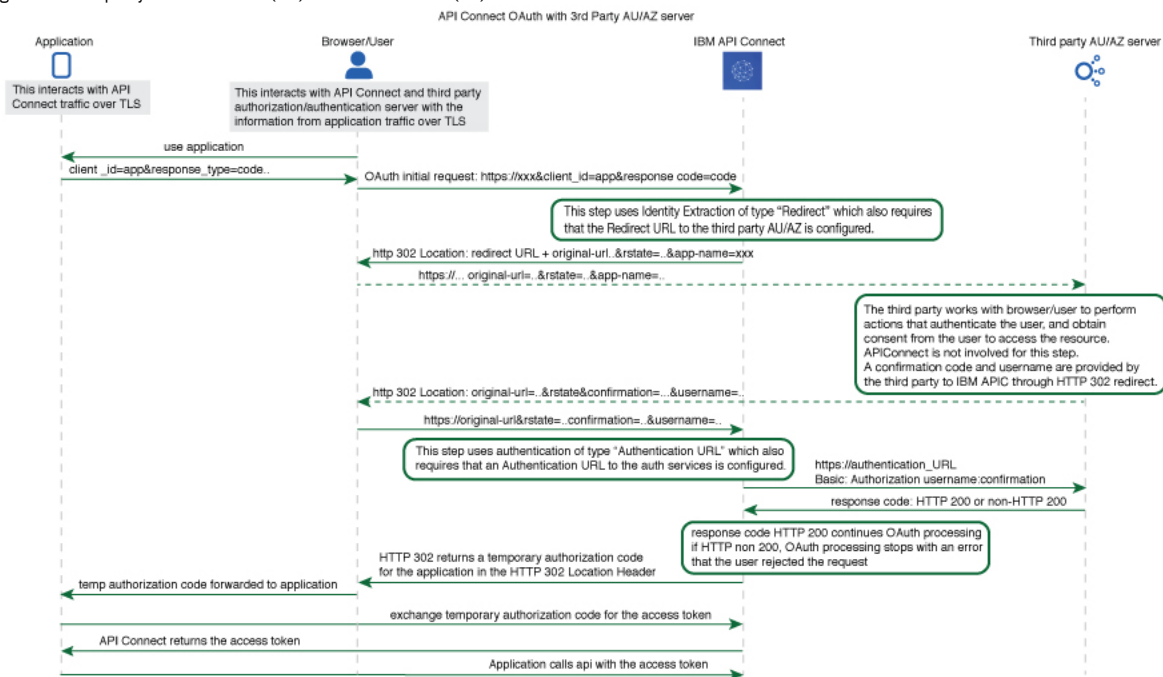
To complete this task, you will need to either create or have created an OAuth security definition that uses Implicit grant type or Access Code (Authorization Code) grant type. For more information, see [Creating an OAuth security definition](#).

### About this task

If you use methods for authentication that are not supported by API Connect, you can redirect users to a suitable URL at which they can authenticate. The user is then returned to the OAuth process after authentication and authorization have been confirmed.

The following illustration indicates the transaction flow for third party authentication.

Figure 1. Third party authentication (AU) and authorization (AZ) transaction flow



1. The application initiates a request to access an API protected with a third-party entity. API Connect redirects the application with an **HTTP 302** redirect based on **identity extraction** -> **redirect** -> **redirect-url**, for user authentication (and optional authorization).
2. The application communicates directly with the third-party entity to gather user identity. API Connect is not involved in this communication. After the third-party entity finishes processing authentication (and optional authorization), it returns an **HTTP 302** redirect that uses the **original-uri** from the request, with the username and confirmation code appended.
3. API Connect receives the request that includes the username and confirmation code, and communicates with the authentication URL, based on **authentication** -> **x-ibm-authentication-url**, to confirm user identity before the request is completed.
4. An **HTTP 200** response from the third-party entity allows API Connect to continue the OAuth 2.0 request process as if the owner is authenticated. The request is then processed according to the **grants** type.
  - - **accessCode** returns a temporary code to the application.
  - - **implicit** returns the access token to the application.

For any response other than HTTP 200, the request fails with a statement added to the error log.

## Procedure

To create an external form, and to indicate the URL to which API Connect will redirect users, complete the following instructions:

1. Create your service for authentication and authorization. You will use the URL of the landing page of this service as your redirect URL.
  - a. To include elements in your form that are provided by API Connect, use the query parameters from the URL that your user is redirected to. When the user is redirected to your page, the URL they are sent to contains the following query parameters:

**app-name**  
The name of the application requesting access, as provided through the Developer Portal.

**appid**  
The id of the application requesting access.

**catalog**  
The name of the catalog where the product is being used by the application.

**catalogid**  
The id of the catalog where the product is being used by the application.

**catalogtitle**  
User-friendly display name for the catalog.

**org**  
The name of the consumer organization that hosts the application.

**orgid**  
The id of the consumer organization that hosts the application.

**orgtitle**  
User-friendly display name for the organization.

**original-url**  
The original URL that the user was directed to by the application, including query parameters from the original URL that are necessary for standard OAuth 2.0 requests. You can include these parameters in your service to provide information to the user. Additionally the `state_nonce` is appended. The `state_nonce` is a hash code generated by API Connect for verification purposes. The URL is URL-encoded and should be decoded before further use, the `state_nonce` should remain unchanged.

**provider**  
The name of the API provider organization.

**providerid**  
The id of the API provider organization.

**providertitle**  
User-friendly display name for the provider organization.

**requested-scope**  
[optional] If [Application Scope check](#) is enabled and replaces the `scope` from the initial application request, this field holds the `scope` value from the initial application request, and the new replacement scope value is put into `original-url`.

**transid**  
transaction id used in the GW for the transaction which trigger this call

The URL to which the user is sent to when they are redirected to your page has the following form:

**`Redirect_URL?original-url=Original_URL&state_nonce=R_State&app-name=Application_Name`**

where all variables are as described previously. The Redirect URL does not have a size limit enforced by API Connect.

- b. Create the stages of authentication, authorization, and any intermediate stages that you require to take place before you allow access to the application. Upon completion of these stages, redirect the user to the `Original_URL` and append a user name, their confirmation code, and the application name to be evaluated for access grant or denial by API Connect. The confirmation code does not have a size limit enforced by API Connect. Original URL requires the following form:

**`Original_URL&username=User_Name&confirmation=Confirmation_Code`**

where all variables are as described previously.

For example:

**`https://your_gateway.com/your_org/your_catalog/your_api/oauth/authorize?response_type=code&redirect_uri=https://example.com/redirect&scope=/your_api&client_id=5af57a4a-6db9-4141-ad08-5709432af66e&state_nonce=HoIbRG+6bZtq1B7LDkq4gj1D3SHKq1CbnYdHs/bMz2Y=&username=spoon&confirmation=12345678`**

- c. To send your own error responses after the authentication and authorization service, redirect the user to the `Original_URL` and append an error code. You can also append a user name. Use the following form:

**`Original_URL&username=User_Name&error=Error_Response`**

where `Error_Response` is the message you wish to send and all other variables are described as previously.

For example:

**`https://your_gateway.com/your_org/your_catalog/your_api/oauth/authorize?response_type=code&redirect_uri=https://example.com/redirect&scope=/your_api&client_id=5af57a4a-6db9-4141-ad08-5709432af66e&state_nonce=HoIbRG+6bZtq1B7LDkq4gj1D3SHKq1CbnYdHs/bMz2Y=&username=User&error=access_denied`**

2. Create a service to validate the confirmation code and user name. API Connect makes a GET call to your authentication URL after the user is redirected back to the authorization URL. When the call is made, it includes in its authorization header the user name and confirmation code you supplied previously. Confirm that these are correct and respond with an HTTP success code such as 200 OK if you want to allow access, or non-200 HTTP response code, such as 401 Unauthorized to deny access.
3. In your OAuth provider configuration, supply the redirect URL that is used in Step 1 and the authentication URL that is used in Step 2. For more information on configuring an OAuth Provider, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Token management with the DataPower Gateway (v5 compatible)

API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

In order to manage tokens with the DataPower® Gateway (v5 compatible), you must set the Token Management Type to Native in your Native OAuth provider configuration. See [Configuring token management and revocation for a native OAuth provider](#) when using Cloud Manager or [Configuring token management and revocation for a native OAuth provider](#) when using API Manager.

Also, the DataPower quota enforcement server must be enabled on the DataPower Gateway to use the distributed cache to manage tokens. See [Quota enforcement](#).

When distributed cache support is enabled, replay protection is provided across the gateway cluster through the quota enforcement server. This support ensures that the same token cannot be reused across the members of the quota enforcement peer group.

Important:

If you have a cluster of DataPower Gateway servers, the OAuth data synchronization behavior across the servers depends on whether or not you enable revocation:

- If you enable revocation, API Connect uses the DataPower quota enforcement server, and OAuth data is synchronized across the servers. If an access token is obtained from one server, the OAuth data synchronization ensures that the same authorization code cannot be used to obtain an access code from another server. You must ensure that the DataPower quota enforcement server is configured.
- If you disable revocation, API Connect does **not** use the DataPower quota enforcement server and OAuth data does not synchronize across the cluster of DataPower Gateway servers. Therefore, to prevent the same authorization code being used to obtain an access code from more than one server you must configure DataPower to synchronize OAuth data across the servers, by using the DataPower Gateway console.

To configure DataPower to synchronize OAuth data, ensure that the DataPower quota enforcement server is configured. For more information, see [Configuring the quota enforcement server](#).

## Resource owner revocation

When Resource owner revocation path is selected in the Token Management screen, the configuration inserts two REST API calls to /oauth2/issued.

- An HTTP GET operation that retrieves a list of all granted permissions for a specific user.
- An HTTP DELETE operation that revokes an application for a specific user.

The setting inserts header-based security definitions of client ID and client secret as shown in the **View permissions example**. The API call to revoke a given application for a given user is shown in the **Revoke permissions example**.

View permissions example

To list all the applications granted by user `cn=spoon,o=ibm` with username `spoon` and password `spoon` using a registered administration application of `5287fe53-8747-438a-8262-681ec75b79c5`.

- Request:

```
GET /oauth2/issued
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
[
  {
    "clientId": "7369ad66-5674-b7d3-4567-de35283421aca",
    "owner": "cn=spoon,o=ibm",
    "clientName": "PetStore Application",
    "scope": "listpet",
    "issuedAt": 1503327054,
    "consentedOn": 1503327054,
    "expiredAt": 1503330654,
    "refreshTokenIssued": false,
    "appId": "d2031f0f27339315333734ab9",
    "org": "PetStoreOrg",
    "orgTitle": "Katie Pet Grooming Inc",
    "orgId": "5887803de4b06e6998c4b2c7",
    "provider": "SuperStore",
    "providerTitle": "Simon SuperStore",
    "providerId": "5887803de4b06e6998c4b2c7",
    "catalog": "publicapi",
    "catalogTitle": "For public",
    "catalogId": "5887803de4b06e6998c4b2d3"
  },
  {
    "clientId": "a8746323-9825-a842-8736-abd8202356ac8",
    "owner": "cn=spoon,o=ibm",
    ...
  }
]
```

Revoke permissions example

To revoke application `a8746323-9825-a842-8736-abd8202356ac8` by owner `cn=spoon,o=ibm`.

- Request

```
DELETE /oauth2/issued?client-id=a8746323-9825-a842-8736-abd8202356ac8
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

## Client revocation

---

When Client revocation path is selected in the Token Management screen, the result is the following:

This option inserts one REST API call to /oauth2/revoke, which supports OAuth 2.0 [IETF RFC 7009](#). An HTTP POST operation that an application can send to this API to revoke either an `access_token`, or `refresh_token` with `token_type_hint` as shown in the following examples:

Revoke `access_token`

- Request:

```
POST /oauth2/revoke
HTTP/1.1

Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=access_token&token=AAIHZGVmYXVsdD1-KqwD0Yc3EDn941SWX14xuR....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

Revoke `refresh_token`

- Request:

```
POST /oauth2/revoke
HTTP/1.1

Host: apic.ibm.com

x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded

token_type_hint=refresh_token&token=.....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache

{ "status": "success" }
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Token management with the DataPower API Gateway

API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.

In order to manage tokens with the DataPower® API Gateway, you must set the Token Management Type to Native in your Native OAuth provider configuration. See [Configuring token management and revocation for a native OAuth provider](#) when using Cloud Manager or [Configuring token management and revocation for a native OAuth provider](#) when using API Manager.

For token management with the DataPower API Gateway, you must configure the API Security Token Manager on the gateway. Complete the following steps:

1. Log in to the DataPower administration console, selecting apiconnect for the domain, and WebGUI for the Graphical Interface.

2. In the search box, enter API Security Token Manager, then click on the API Security Token Manager link that displays in the search results.
3. For the Administrative state, select enabled.
4. Click the + icon alongside the Gateway Peering field to create a new gateway peering object.
5. Provide a Name and a Local address.
6. In the Local port and Monitor port fields, provide values that aren't already in use by services for the Local address.
7. Ensure that Peer group mode is selected.
8. Alongside the Peers field, use the add button and accompanying field to add at least two peers, to ensure that quorum is achieved.
9. If the Enable SSL option is selected, select a value for the Identification credentials.
10. For Persistence location, select a setting other than memory.
11. When done, click Apply.
12. Repeat steps 1 to 11 for each DataPower API Gateway in the peer group.
13. When done, click Apply, then click Save Configuration to save your changes.

For more information, see [Defining the API security token manager](#) and [Creating a gateway peering instance](#).

## Resource owner revocation

When Resource owner revocation path is selected in the Token Management screen, the configuration inserts two REST API calls to /oauth2/issued.

- An HTTP GET operation that retrieves a list of all granted permissions for a specific user.
- An HTTP DELETE operation that revokes an application for a specific user.

The setting inserts header-based security definitions of client ID and client secret as shown in the **View permissions example**. The API call to revoke a given application for a given user is shown in the **Revoke permissions example**.

View permissions example

To list all the applications granted by user `cn=spoon,o=ibm` with username `spoon` and password `spoon` using a registered administration application of `5287fe53-8747-438a-8262-681ec75b79c5`.

- Request:

```
GET /oauth2/issued
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iNluU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
[
  {
    "clientId": "7369ad66-5674-b7d3-4567-de35283421aca",
    "owner": "cn=spoon,o=ibm",
    "clientName": "PetStore Application",
    "scope": "listpet",
    "issuedAt": 1503327054,
    "consentedOn": 1503327054,
    "expiredAt": 1503330654,
    "refreshTokenIssued": false,
    "appId": "d2031f0f27339315333734ab9",
    "org": "PetStoreOrg",
    "orgTitle": "Katie Pet Grooming Inc",
    "orgId": "5887803de4b06e6998c4b2c7",
    "provider": "SuperStore",
    "providerTitle": "Simon SuperStore",
    "providerId": "5887803de4b06e6998c4b2c7",
    "catalog": "publicapi",
    "catalogTitle": "For public",
    "catalogId": "5887803de4b06e6998c4b2d3"
  },
  {
    "clientId": "a8746323-9825-a842-8736-abd8202356ac8",
    "owner": "cn=spoon,o=ibm",
    ...
  }
]
```

Revoke permissions example

To revoke application `a8746323-9825-a842-8736-abd8202356ac8` by owner `cn=spoon,o=ibm`.

- Request

```
DELETE /oauth2/issued?client-id=a8746323-9825-a842-8736-abd8202356ac8
HTTP/1.1
Host: apic.ibm.com
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iNluU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
Authorization: Basic c3Bvb246c3Bvb24=
```

- Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache
```

```
{ "status": "success" }
```

## Client revocation

When Client revocation path is selected in the Token Management screen, the result is the following:

This option inserts one REST API call to `/oauth2/revoke`, which supports OAuth 2.0 [IETF RFC 7009](#). An HTTP POST operation that an application can send to this API to revoke either an `access_token`, or `refresh_token` with `token_type_hint` as shown in the following examples:

Revoke `access_token`

- Request:

```
POST /oauth2/revoked
HTTP/1.1
```

```
Host: apic.ibm.com
```

```
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded
```

```
token_type_hint=access_token&token=AAIHZGVmYXVsdD1-KqwD0Yc3EDn94lSWX14xuR....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache
```

```
{ "status": "success" }
```

Revoke `refresh_token`

- Request:

```
POST /oauth2/revoked
HTTP/1.1
```

```
Host: apic.ibm.com
```

```
x-ibm-client-id: 5287fe53-8747-438a-8262-681ec75b79c5
x-ibm-client-secret: E2qM6mG2bX2uClxT2iN1uU6bt5cV4dN7nW5kM5uP8vL3uF3cT7
Content-Type: application/x-www-form-urlencoded
```

```
token_type_hint=refresh_token&token=.....
```

- Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: private, no-store, no-cache, must-revalidate
Pragma: no-cache
```

```
{ "status": "success" }
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authentication URL user registry

You can use an Authentication URL user registry to specify a REST authentication service that manages user authentication, and optionally provides additional metadata to be embedded in the token.

This support can optionally enable any of the following:

- Providing the authenticated credential to IBM® API Connect. For example, the user logs-in with user name: `spoon`, and password: `fork`. When the user is authenticated, the credential becomes `cn=spoon,o=eatery`. The credential is kept in the OAuth `access_token` to represent the user.
- Providing metadata support. Allow extra metadata to be stored in the `access_token`.
- Overriding the `scope` that the application receives after a successful OAuth protocol processing. By responding with a specific header, the Authentication URL endpoint can replace the `scope` value that the application receives. For example, you can provide a specific resource owner an account number within the `scope` header response for use in future processing steps.

When you call the Authentication URL user registry, the API Connect gateway sends a GET request with HTTP headers and then processes any HTTP response from the URL. For authentication, a REST authentication service is expected at the Authentication URL.

The following response from the REST authentication service indicates that user authentication is successful and that API Connect will use `cn=spoon,o=eatery` as the user identity.

```
HTTP/1.1 200 OK
Server: example.org
X-API-Authenticated-Credential: cn=spoon,o=eatery
```

For information on how to configure a User Security policy in an API assembly for use with an Authentication URL user registry, see [User Security policy](#).

For an example of an OAuth provider configuration that uses an Authentication URL user registry, see [Example - using multiple OAuth policies in an OAuth provider assembly](#).

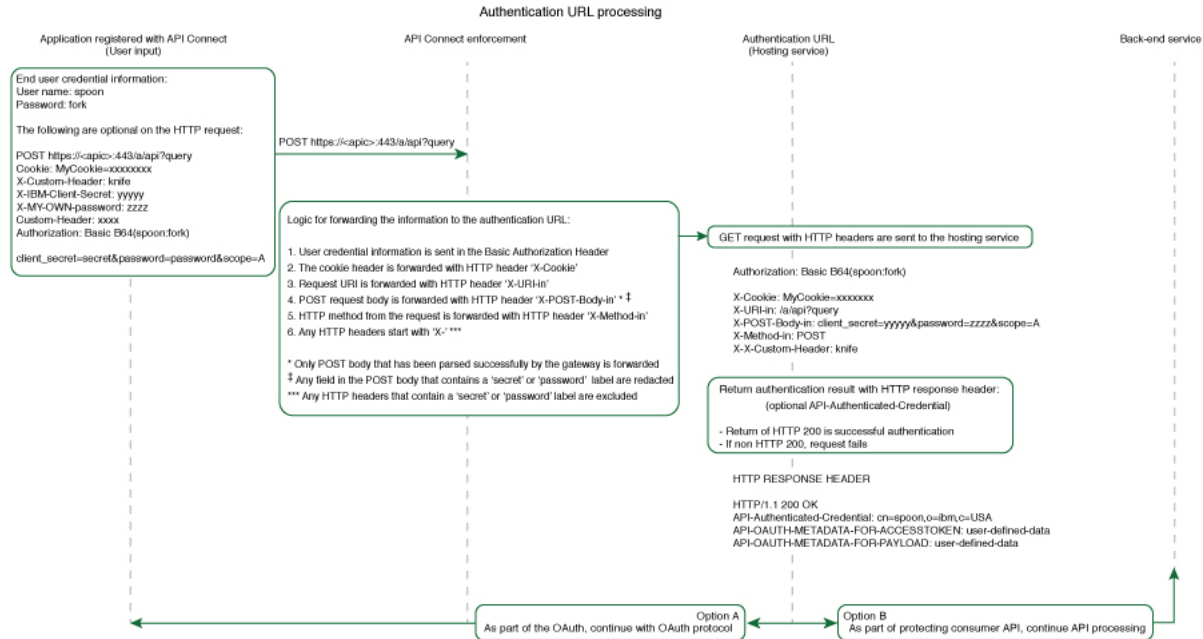
API Connect considers any non-200 HTTP response code a failed user authentication attempt.

When an Authentication URL user registry is invoked, two HTTP response headers are available that include metadata in the access token or the response payload that contains the access token. For more information, see [OAuth external URL and authentication URL](#). The two metadata response headers are:

**API-OAUTH-METADATA-FOR-ACCESSTOKEN**  
**API-OAUTH-METADATA-FOR-PAYLOAD**

When an Authentication URL is invoked, an HTTP response header is available to override the requested **scope** from the application. For more information, see [Scope](#). The response header is:

**x-selected-scope**



If you are using the DataPower® API Gateway rather than the DataPower Gateway (v5 compatible), this diagram is provided for guidance only and is not fully accurate for this release.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Custom forms for user security

You can create custom forms for the authorization and identity extraction phase of OAuth.

### About this task

The Native OAuth provider configuration provides the capability for requiring additional authentication and authorization steps for user security. Custom HTML forms may be created to extract the identity of the user and to authorize the user. This capability applies to Implicit, Resource owner password, and Access code grant types.

- [Creating a custom HTML login form for user security](#)  
Custom HTML forms can be created for user security during the identity extraction stage in OAuth.
- [Creating a custom HTML authorization form for user security](#)  
Custom HTML forms can be created for user security during the authorization stage in OAuth.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a custom HTML login form for user security

Custom HTML forms can be created for user security during the identity extraction stage in OAuth.

## Before you begin

The Native OAuth provider configuration includes Identity Extraction when using the Implicit, Access code, or Resource owner password grant types. You have the option to select how to extract the user credential and one of the choices is Custom HTML Form. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager. For more information, see [Configuring a native OAuth provider](#). This topic describes how to create the Custom HTML form for identity extraction.

## About this task

During three-legged OAuth definitions (Implicit flow, Resource owner password flow, and Access (Authorization) code flow, the user is presented with a form for signing in to the service provided by the API. You can present a custom form or a default form. Your custom form must fulfill certain requirements.

Important: The fields used by IBM® API Connect to inject information into your form have case-sensitive field names.

## Procedure

To create a custom sign-in form for your Native OAuth provider, complete the following steps:

1. Create a well formed XHTML document that will be parsed and transformed by API Connect to inject hidden fields.
2. For your XHTML form, set the method as `POST`, the encoding type as `application/x-www-form-urlencoded`, and the action as `authorize`. Add any other parameters that you require.  
For example:

```
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">
```

3. Create a text input field that is named `username` and create a password input field named `password`.
4. Add the line `<EI-INJECT-HIDDEN-INPUT-FIELDS>`. This third element is a placeholder that API Connect replaces with input fields to complement the user-submitted data.
5. Create a button to initiate the sign-in process.  
For example:

```
<button  
id=" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.apionprem.doc_oauth_custom_login_form_2_login_b  
utton" type="submit" name="login" value="true">Log in</button>
```

6. Optional: Add text that is displayed the first time that the user visits the sign-in page. Use the tag `<EI-LOGINFIRSTTIME>` for the text that you want to display.
7. Optional: Add text that appears when the user is returned to the sign-in page if they fail to authenticate. Use the tag `<EI-LOGINFAILED>` for the text that you want to display.
8. Optional: Have an error message displayed when an error in the custom form prevents it from being displayed to the user correctly. Use the tag `<EI-INTERNAL-CUSTOM-FORM-ERROR/>`; the message text is generated automatically. You should detect such errors during testing to prevent this error message being displayed to the end user.
9. Optional: You can add elements that are loaded from external sources, such as images or JavaScript.
10. Insert spacing and other features as you require. Completing Steps [1](#) through to [8](#) results in a form similar to the following example:

```
<html lang="en" xml:lang="en">  
<head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /></head>  
<body>  
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">  
<h1>Please sign in</h1>  
<p>Username </p>  
<p ><input type="text" name="username" required="required" /> </p>  
<p>Password </p>  
<p ><input type="password" name="password" required="required" /> </p>  
<EI-INJECT-HIDDEN-INPUT-FIELDS/>  
<p > <button  
id=" home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.apionprem.doc_oauth_custom_login_form_2_login_b  
utton" type="submit" name="login" value="true">Log in</button> </p>
```

```
<EI-LOGINFIRSTTIME>  
<p>If you have forgotten your user name or password, contact your system administrator.</p>  
</EI-LOGINFIRSTTIME>
```

```
<EI-LOGINFAILED>  
<p >At least one of your entries does not match our records.  
If you have forgotten your user name or password, contact your system administrator.</p>  
</EI-LOGINFAILED>
```

```
<EI-INTERNAL-CUSTOM-FORM-ERROR/>
```

```
</form>  
</body>  
</html>
```

11. Make your form available at a URL of your choice.
12. If you have not already done so, configure your Native OAuth provider to use a Custom HTML form for identity extraction and provide the URL at which your form is available. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

## Related tasks

- [Creating a custom HTML authorization form for user security](#)

## Related information



- [Configuring API security](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating a custom HTML authorization form for user security

Custom HTML forms can be created for user security during the authorization stage in OAuth.

### Before you begin

The Native OAuth provider configuration includes user authorization when using the Implicit, Access code, or Resource owner password grant types. You have the option to select how to authorize application users and one of the choices is Custom HTML Form. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager. This topic describes how to create the Custom HTML form for authorization.

### About this task

During three-legged OAuth definitions (Implicit flow, Resource owner password flow, and Access (Authorization) code flow, the user is presented with a form through which they grant permission to an application to access their data through the API on their behalf. You can present a custom form or a default form. Your custom form must fulfill certain requirements.

Important: The fields used by IBM® API Connect to inject information into your form have case-sensitive field names.

### Procedure

To create a custom authorization form for your Native OAuth provider, complete the following steps:

1. Create a well-formed XHTML document. This will be parsed and transformed by API Connect to inject hidden fields.
2. For your XHTML form, set the method as POST, the encoding type as `application/x-www-form-urlencoded`, and the action as `authorize`. Add any other parameters that you require.  
For example:

```
<form method="POST" enctype="application/x-www-form-urlencoded" action="authorize">
```

3. Add the line `<AZ-INJECT-HIDDEN-INPUT-FIELDS/>`. This line is a placeholder that API Connect will replace with input fields necessary for the completion of the OAuth process.
4. Create two buttons with the following code so that the user can grant or deny permission. Edit the text to suit your preferences.

```
<button class="cancel" type="submit" name="approve" value="false">No Thanks</button>  
<button class="submit" type="submit" name="approve" value="true">Allow Access</button>
```

5. Optional: Display an error message when an error in the custom form prevents it from being displayed to the user correctly. Use the tag `<AZ-INTERNAL-CUSTOM-FORM-ERROR/>`; the message text is generated automatically. You should detect such errors during testing to prevent this error message being displayed to the end user.
6. Optional: You can add to the form HTML elements that will load features from external sources, such as images or JavaScript.  
For example, `<script src="http://www.example.com/example.js" />`
7. Insert spacing and additional elements as you require. Completing Steps [1](#) through to [6](#) results in a form similar to the following example:

```
<html lang="en" xml:lang="en">  
<head><title>Request for permission</title></head>  
<body class="customconsent">  
  <div>  
    <div>  
      <form method="post" enctype="application/x-www-form-urlencoded" action="authorize">  
        <AZ-INJECT-HIDDEN-INPUT-FIELDS/>  
        <p>Greeting..</p><DISPLAY-RESOURCE-OWNER/>  
        <p>This app </p><OAUTH-APPLICATION-NAME/><p> would like to access your data.</p>  
        <div>  
          <button class="cancel" type="submit" name="approve" value="false">No Thanks</button>  
          <button class="submit" type="submit" name="approve" value="true">Allow Access</button>  
        </div>  
      </form>  
    </div>  
    <AZ-INTERNAL-CUSTOM-FORM-ERROR/>  
  </div>  
</body>  
</html>
```

8. Make your form available at a URL of your choice.
9. If you have not already done so, configure your Native OAuth provider to use a Custom HTML Form for authorization for User Security. Provide the URL as the endpoint at which your form is available. For more information, see [Configuring a native OAuth provider](#) when using API Manager or [Configuring a native OAuth provider](#) when using Cloud Manager.

### Related tasks

- [Creating a custom HTML login form for user security](#)

## Related information

---

- [Configuring API security](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Securing an API with a JSON Web Token

There are two methods to secure your API with a JSON Web Token. You can use the **jwt-generate** command, or you can use a token that has been generated external to IBM® API Connect.

### About this task

---

JSON Web Token (JWT) is an OAuth 2.0 compliant method of authentication that can be useful to secure your API in API Connect.

You can secure your API with a JSON Web Token by using either of the following methods:

- Generate a token through the **jwt-generate** command, and then augment the response payload with your generated token replacing the `id` token.
- Use a token that was generated outside of API Connect and include it into the response payload, by using the metadata URL.

### Procedure

---

Create a **jwt-generate** policy in the assembly.

- In API Manager, open the Assembly tab.
- Add a **jwt-generate** policy to the assembly for the API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting OAuth

You can find the answers to some common questions in the Developer Portal, or in support communities and forums in DeveloperWorks, on GitHub, or on Youtube.

You can use the following links to go to the topics:

- [Enable an extended error description](#)
- [OAuth 2.0 support in the developer portal test tool](#)
- [OAuth requires quorum in a multi-node cluster](#)

### Enable an extended error description

---

The [OAuth 2.0 Authorization Framework](#) governs how IBM® API Connect is defined. According to the specification, OAuth 2.0 error conditions trigger a payload with `error`, and an optional field `error_description`. To prevent information leakage, the IBM DataPower® Gateway default setting returns the error only. However, during the development and testing phase of an application, it is useful to find out why an OAuth 2.0 request is being rejected. To enable this behavior, the caller can pass an HTTP request header in the OAuth request:

```
APIm-Debug: true
```

The presence of the header enables the 'just-in-time' debugging for that OAuth transaction, and `error_description` is returned as part of the error condition. The output helps the caller to determine why an OAuth request is rejected by API Connect as shown in the examples.

Example error output without 'just-in-time' debugging:

```
{ "error": "invalid_request" }
```

Example error output with 'just-in-time' debugging:

```
{ "error": "invalid_request", "error_description": "Multiple OAuth client credentials are provided" }
```

### OAuth 2.0 support in the developer portal test tool

---

OAuth 2.0 support is exposed as an API through the provider OpenAPI definition. You can use the test tool that is included with API Connect to test the OAuth 2.0 configuration. For more information, see [Testing an API using the Developer Portal test tool](#).

### OAuth requires quorum in a multi-node cluster

---

In a multi-node cluster, OAuth operations will fail if quorum is lost. Quorum requires that the number of active nodes is greater than 50% of the total number of nodes in the cluster.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Working with Products in the API Manager

In IBM API Connect, Plans and APIs are grouped together in Products.

Information about managing Products in the API Manager UI can be found in the following topics.

- [The Product lifecycle](#)  
When you manage your Product versions, you move them through a series of lifecycle states. From initially staging a Product version to a Catalog, through to publishing to make the Product version available to your application developers, and to eventual retiring and archiving. The syndication feature in IBM API Connect means that Product lifecycle states can also be managed within Spaces in the associated Catalog.
- [Managing your Products](#)  
You can manage your Products in API Manager by using the Products page of the associated Catalog. In this view, you can move the Products through their lifecycle, display analytics information, and control who can see or subscribe to the Products. The syndication feature in IBM API Connect means that Products can also be managed by using the Products tab of the associated Space.
- [Managing the APIs in a Product](#)  
For a specific Product, you can take an individual API offline. If an API is taken offline then it becomes unavailable and calls to the API will fail.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

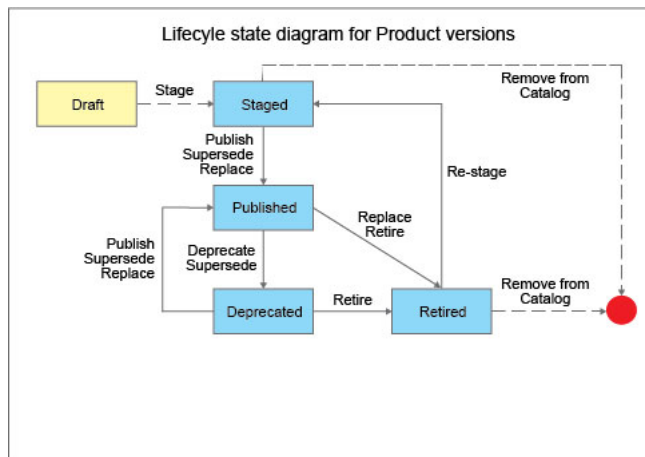
## The Product lifecycle

When you manage your Product versions, you move them through a series of lifecycle states. From initially staging a Product version to a Catalog, through to publishing to make the Product version available to your application developers, and to eventual retiring and archiving. The syndication feature in IBM® API Connect means that Product lifecycle states can also be managed within Spaces in the associated Catalog.

---

## Product lifecycle state diagram

The following diagram shows the possible lifecycle states for a Product version, and the Product management operations that move a Product version from one lifecycle state to another. For example, the Retire operation moves a Product version from the Published to the Retired state.



Note: The same Product lifecycle states apply irrespective of whether your Product is managed within a Catalog, or within a Space in a Catalog. For more information about the syndication feature, see [Using syndication in API Connect](#).

If approval is required for a Product management operation, an approval request is sent and the Product version moves to the pending state. When the request is approved, the operation is completed and the Product version moves to the next lifecycle state. If approval is not required, the operation is completed immediately.

Note: Approval is not required for the following lifecycle state transitions:

- Retired to Staged.
- Deprecated to Published.

For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

The following sections describe the various lifecycle states for a Product version.

Note: All references in this topic to a Catalog, can also be applied to a Space in a Catalog, unless specified otherwise.

---

## Draft

The draft state for a Product or API is when a Product or API definition is not deployed and is not associated with any Catalog.

---

## Staged

When you stage a Product, a copy of the Product version is deployed to the target Catalog. Staged is the initial state when you publish a Product. When a Product is in the staged state, it is not yet visible to, or subscribable by, any developers. For more information about staging a Product, see [Staging a Product](#).

You stage a Product so that the appropriate approvals, internally within the organization, can be given for it to then be published. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#). For information on publishing a Product, see [Publishing a Product](#).

---

## Published

When you publish a Product, a fixed copy of the Product version is deployed to the target Catalog. The Product version is visible to, and subscribable by, the targeted developers or communities. When a Product is published in a Catalog, the visibility and subscription settings can be changed for the published version of that Product. Any further changes require a new version of the Product to be staged and published before they take effect.

If you replace a Published Product with a Staged or Deprecated Product, the replacement Product is published, and the replaced Product is retired.

If you supersede a Published Product with a Staged or Deprecated Product, the superseding Product is published, and the superseded Product is deprecated.

For more information about publishing a Product, see [Publishing a Product](#).

---

## Deprecated

When you deprecate a Product, the Product version is visible only to developers whose applications are currently subscribed. No new subscriptions to the Plans in the Product are possible. For more information about deprecating Products, see [Deprecating a Product](#).

A Product is also deprecated if you supersede it with another Product. For more information, see [Superseding a Product with another Product](#).

---

## Retired

When you retire a Product, the Product version can neither be viewed nor can its Plans be subscribed to, and all of the associated APIs are taken offline. For more information about retiring Products, see [Retiring a Product](#).

A Product is also retired if you replace it with another Product. For more information, see [Replacing a Product with another Product](#).

---

## Related concepts

- [Working with Catalogs](#)

---

## Related tasks

- [Managing your Products](#)
- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing your Products

You can manage your Products in API Manager by using the Products page of the associated Catalog. In this view, you can move the Products through their lifecycle, display analytics information, and control who can see or subscribe to the Products. The syndication feature in IBM® API Connect means that Products can also be managed by using the Products tab of the associated Space.

---

## Before you begin

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## About this task

---

For more information about the Product lifecycle, see [The Product lifecycle](#).






For more information about the syndication feature, see [Using syndication in API Connect](#).

For information on viewing Analytics that can provide insight into Product and API usage and performance, see [API Analytics](#).

## Procedure

---

To manage your Product, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Expand a Product to see details of the APIs and Plans in that Product.  
Manage options are also available, depending on the state of the Product. To view the manage options, click the options icon  and select the options that you require.
4. To manage the lifecycle of a version of a Product, click the options icon  alongside the Product version, and select the required lifecycle action.
5. To set the migration target of a version of a Product, click the options icon  alongside the Product version, and select Set migration target. In the Set migration target window, select the Product that you want to set as the migration target, and select Next. Then map the migration source Plans to the migration target Plans, and click OK.  
Restriction: If the Plan is part of a Product that is contained within a Space in the Catalog, and the migration action is being made at the Space level, the Plan that you are migrating subscriptions from must be located in the same Space as the Plan you are migrating to. If the migration action is being made at the Catalog level, subscriptions can be migrated across Spaces. For more information about the Space feature, see [Using syndication in API Connect](#).

## What to do next

---

For details of the ways in which you can manage your Products, see the following subtopics:

- [Publishing a Product](#)  
APIs become accessible when a Product is published and made visible on the IBM API Connect Developer Portal for use by application developers. A Product can be published to selected communities of application developer organizations, and the Plans within the Product can be used to tailor access and visibility further.
- [Changing the availability of a Product](#)  
You can change the availability of a Product and the associated Plans by using the Products view within a Catalog in API Manager. The syndication feature in IBM API Connect means that you can also use the Products view within a Space in a Catalog to change the availability of a Product.
- [Deprecating a Product](#)  
You can deprecate a published Product by using the Manage Products page within a Catalog in API Manager. When you deprecate a Product, application developers that are already subscribed to the Product can continue to use it, but no new developers can subscribe to the Product. The syndication feature in IBM API Connect means that you can also use the Manage Products page within a Space in a Catalog to deprecate a published Product.
- [Retiring a Product](#)  
You can retire a published or deprecated Product by using the Manage Products page within a Catalog in API Manager. When a Product is retired, all associated APIs are taken offline, and any subscriptions are deleted. The syndication feature in IBM API Connect means that you can also use the Manage Products page within a Space in a Catalog to retire a Product.
- [Re-staging a Product](#)  
If a Product in a Catalog has been retired, you can re-stage the Product so that it can re-enter the Product lifecycle. A re-staged Product moves to the Staged state, from where it can be published. The syndication feature in IBM API Connect means that you can also re-stage a Product within a Space in a Catalog.
- [Updating the gateway services for a Product](#)  
When you stage or publish a draft Product to a Catalog, you specify the gateway services at which the APIs in the Product are to be made available. However, you can later update the selected gateway services after a Product has been stage or published.
- [Removing a Product from a Catalog](#)  
You can remove a Product from a specific Catalog by using the Products view within a Catalog in API Manager. The syndication feature in IBM API Connect means that you can also use the Products view within a Space to remove a Product from a Catalog.
- [Approving Product lifecycle and subscription requests](#)  
To approve or decline requests to change the lifecycle state of a Product, and requests by application developers to subscribe to a Plan, use the Tasks page in the Catalog that contains the associated Product in API Manager. The syndication feature in IBM API Connect means that you can also use the Tasks page within a Space in a Catalog to approve or decline requests.

## Related tasks

---

- [Creating and configuring Catalogs](#)
- [Working with Spaces](#)

## Related information

---

- [Staging a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Publishing a Product

APIs become accessible when a Product is published and made visible on the IBM® API Connect Developer Portal for use by application developers. A Product can be published to selected communities of application developer organizations, and the Plans within the Product can be used to tailor access and visibility further.

### Before you begin

---

You must stage a Product before it can be published. For more information about staging Products, see [Staging a Product](#).  
Note: If you want to publish a LoopBack project, you must publish both the APIs (by publishing the Products that contain the APIs) and the associated applications so the project can be run. For more information about publishing LoopBack applications, see [Publishing APIs and applications](#).  
To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

### About this task

---

A community is a collection of consumer organizations, used to control which organizations have access to Products and Plans without having to assign access on an individual basis. A Product can be published to selected communities, which means that only application developers within those organizations contained within the community can see the Product on the Developer Portal and obtain application keys to access it. A Product can alternatively be published to all communities. Communities are used to restrict the visibility and accessibility of APIs, for example to particular business partners, internal organizations, or other groups of application developers.

### Procedure

---

You can publish a product in any of the following ways:

- Publish a new Product.
- Replace a Product with another Product.
- Supersede a Product with another Product  
For details of the ways in which you can publish a Product, see the following subtopics:
- [Publishing a new Product](#)  
Publish a Product from within its containing Catalog in API Manager. The syndication feature in IBM API Connect means that you can also publish a Product from within its containing Space in a Catalog.
- [Replacing a Product with another Product](#)  
You can replace a published Product in IBM API Connect with another Product, and automatically migrate subscribers to the new Product, by using the API Manager.
- [Superseding a Product with another Product](#)  
You can supersede a published Product with another Product by using the API Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Publishing a new Product

Publish a Product from within its containing Catalog in API Manager. The syndication feature in IBM® API Connect means that you can also publish a Product from within its containing Space in a Catalog.

### Before you begin

---

The Product that you are publishing must be in the Staged or Deprecated state.

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

### About this task

---

You can complete this task either by using either the API Designer UI application, or by using the browser based API Manager UI.

If you publish a Product to a non-development Catalog, the Product that is published is an independent and fixed copy of the version of the Product that you chose to stage. Editing the Product through the Products page will not affect the published Product. For this reason, it is recommended that when you stage a Product, you then create a new version of the Product to edit in future, so as to avoid confusion regarding the properties of the published Product. For more information on creating new versions of your Product, see [Creating a new version of your Product](#).

An exception is that if you publish a Product to a development Catalog, editing it through the Products page will enable you to re-stage and publish the same version of the Product. For information on how to create a development Catalog, see [Creating and configuring Catalogs](#).




Although you can publish to a development Catalog, the development Catalog should be used only for testing purposes. Similarly, a Developer Portal created from a development Catalog must be used for testing purposes only, and not for production use. For more information on Catalogs, see [Working with Catalogs](#).

Note: All references in this topic to a Catalog can also be applied to a Space in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in API Connect](#).

## Procedure

---

To publish a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
  2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
    - a. In the navigation pane of the API Manager UI, click  Spaces.
    - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
  3. Alongside the Product version that you want to work with, click the options icon  and then click Publish.  
The Confirm Visibility Settings page opens.
  4. Specify the following options:
    - The users that the Product is visible to  
You can choose Public users, Authenticated users, or Custom.
    - Custom - You can use the Type to add... field to search for organizations or communities that you want your Product to be visible to.
    - Who can subscribe to the Product  
You can add or remove one or more consumer organizations or communities.
    - Custom - You can use the Type to add... field to search for organizations or communities that you want to allow to subscribe to your Product.
  5. Click Publish, then click Confirm to proceed with the publish operation.  
If approval is required to publish Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is published when the request is approved. If approval is not required, the Product version is published immediately, and moves to the Published state. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).
- Note:
- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
  - Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your Product is in the Published state.

Your Product is published to your Catalog and available to your specified organizations or communities. Application developers within the groups you selected can see and use the APIs within the Product.

Any application developer requests to use your Product are displayed on the Approvals tab in the containing Catalog, where you can decline or accept the request.

## Related tasks

---

- [Deprecating a Product](#)

## Related information

---

- [Staging a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Replacing a Product with another Product

You can replace a published Product in IBM® API Connect with another Product, and automatically migrate subscribers to the new Product, by using the API Manager.

## Before you begin

---

The Product to be replaced must be in the Published state, and the replacement Product must be in the Staged, Published, or Deprecated state.

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

Restriction: If Spaces are enabled in the Catalog, both the Product you are replacing and the replacement Product must be located in the same Space. For more information about the Space feature, see [Using syndication in API Connect](#).

## About this task

---

When you replace a Product with another Product, the following actions are taken:

- The replacement Product is published.
- If the visibility and subscribability settings in the replacement Product are such that access is the same as, or less restrictive than, the original Product, the settings in the replacement Product are used. If the settings in the replacement Product are more restrictive, meaning that fewer consumer organizations can see or subscribe to the Product, the replace operation fails. For more information on visibility and subscribability settings, see [Changing the availability of a Product](#).
- The subscribers to the original Product are migrated to the replacement Product.
- The original Product is moved to the Retired state. Products in the Retired state are removed from the Developer Portal; they are no longer visible to the application developers, and any subscriptions to them are canceled. However, the Product can be staged again later if required.




Although you can publish to a development Catalog, the development Catalog should be used only for testing purposes. Similarly, a Developer Portal created from a development Catalog must be used for testing purposes only, and not for production use. For more information on Catalogs, see [Working with Catalogs](#).

Note: All references in this topic to a Catalog, can also be applied to a Space in a Catalog, unless specified otherwise.

## Procedure

---

To replace a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with. The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with. The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to replace, click the options icon , then click Replace. The Replace window opens.
4. Select the replacement Product, then click Next.
5. From the drop-down list, select which Plans from your new Product correspond to Plans in your old Product.
6. Click Replace.

If approval is required to replace Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is replaced when the request is approved. If approval is not required, the Product is replaced immediately. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your new Product is in the Published state, and the original Product is in the Retired state.

Your new Product is published to your preferred organizations or communities. Application developers within the groups you selected can see and use the APIs within the Product.

Any application developer requests to use your Product are displayed on the Approvals tab in the containing Catalog, where you can decline or accept the request.

## Related concepts

---

- [The Product lifecycle](#)

## Related information

---

- [Staging a Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Superseding a Product with another Product



You can supersede a published Product with another Product by using the API Manager.

## Before you begin

---

The Product to be superseded must be in the Published state. The superseding Product must be in the Staged, Published, or Deprecated state.

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM® API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

Restriction: If Spaces are enabled in the Catalog, both the Product you are superseding and the superseding Product must be located in the same Space. For more information about the Space feature, see [Using syndication in API Connect](#).

## About this task

---

When you supersede a Product with another Product, the following actions are taken:

- The superseding Product is published.
- If the visibility and subscribability settings in the superseding Product are such that access is the same as, or less restrictive than, the original Product, the settings in the superseding Product are used. If the settings in the superseding Product are such that access is more restrictive, meaning that fewer consumer organizations can see or subscribe to the Product, the supersede operation fails. For more information on visibility and subscribability settings, see [Changing the availability of a Product](#).
- The original Product is moved to the Deprecated state.
- The application developers that are already subscribed to the now deprecated Product can continue to use it, but no new developers can subscribe to the Product. In the Developer Portal the application developers will see a Migrate this subscription message, which they can click to upgrade their subscription to the migration target.
- The deprecated product can be published again if required.




Although you can publish to a development Catalog, the development Catalog should be used only for testing purposes. Similarly, a Developer Portal created from a development Catalog must be used for testing purposes only, and not for production use. For more information on Catalogs, see [Working with Catalogs](#).

Note: All references in this topic to a Catalog, can also be applied to a Space in a Catalog, unless specified otherwise.

## Procedure

---

To supersede a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to supersede, click the options icon  and then click Supersede.
4. Select the superseding Product and then click Next.
5. From the drop-down list, select which Plans from your new Product correspond to Plans in your old Product.
6. Click Supersede.

If approval is required to supersede Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is superseded when the request is approved. If approval is not required, the Product is superseded immediately. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your superseding Product is in the Published state.

The Product that was superseded is in the Deprecated state.

Your superseding Product is published to your preferred organizations or communities. Application developers within the groups you selected can see and use the APIs within the Product. The original Product is deprecated.

Any application developer requests to use your Product are displayed on the Approvals tab in the containing Catalog, where you can decline or accept the request.

## Related concepts

---

- [The Product lifecycle](#)

## Related tasks

---

- [Deprecating a Product](#)
- [Publishing a new Product](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing the availability of a Product

You can change the availability of a Product and the associated Plans by using the Products view within a Catalog in API Manager. The syndication feature in IBM® API Connect means that you can also use the Products view within a Space in a Catalog to change the availability of a Product.

### Before you begin

---

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).




The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

**Note:** All references in this topic to a Catalog can also be applied to a Space in a Catalog, unless specified otherwise. For more information about Spaces, see [Using syndication in API Connect](#).

### Procedure

---

To change the availability of a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to work with, click the options icon , then click Edit visibility.  
The Visibility page opens.
4. In the Visibility section, specify the users that you want the Product to be visible to. You can choose Public to make the Product visible to all users, Authenticated to make the Product available to users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that you want the Product to be visible to.
  - If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that you want the Product to be visible to. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).
5. In the Subscribability section, specify the users that can subscribe to the Plans in the Product. You can choose Authenticated to make the Plans in the Product subscribable by users who have successfully authenticated, or Custom to specify the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product.
  - If you select Custom, use the Type to add organizations field to search for the consumer organizations and consumer organization groups that can subscribe to the Plans in the Product. If you selected custom visibility, only the consumer organizations and consumer organization groups selected there are available for adding to the custom subscribability list. For information about how to create and manage consumer organizations and consumer organization groups, see [Administering Consumer organizations](#).
6. Click Save when done.  
The Product version is configured with the modified availability settings.  
**Note:**
  - Changing who can view or subscribe to a Product does not affect the settings of the draft Product. The change applies only in the Catalog that was selected in Step 1.
  - Changing who can view or subscribe to a Product does not affect existing subscriptions.

### Results

---

Your Product remains in the same lifecycle state, now with new availability settings.

### Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deprecating a Product

You can deprecate a published Product by using the Manage Products page within a Catalog in API Manager. When you deprecate a Product, application developers that are already subscribed to the Product can continue to use it, but no new developers can subscribe to the Product. The syndication feature in IBM® API Connect means that you can also use the Manage Products page within a Space in a Catalog to deprecate a published Product.

## Before you begin

---

The Product that you are deprecating must be in the Published state.




To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## Procedure

---

To deprecate a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with. The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with. The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to work with, click the options icon , then click Deprecate. The Deprecate Product window opens.
4. Click Confirm to deprecate the Product.

If approval is required to deprecate Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is deprecated when the request is approved. If approval is not required, the Product is deprecated immediately. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your Product is in the Deprecated state.

## Related concepts

---

- [Using syndication in API Connect](#)

## Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Retiring a Product

You can retire a published or deprecated Product by using the Manage Products page within a Catalog in API Manager. When a Product is retired, all associated APIs are taken offline, and any subscriptions are deleted. The syndication feature in IBM® API Connect means that you can also use the Manage Products page within a Space in a Catalog to retire a Product.

## Before you begin

---

The Product that you are retiring must be in the Published or Deprecated state.




To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## Procedure

---

To retire a Product, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to work with, click the options icon  and then click Retire. The Retire Product window opens.
4. Click Confirm.

If approval is required to retire Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is retired when the request is approved. If approval is not required, the Product is retired immediately. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your Product is in the Retired state. It is also removed from the Developer Portal, is no longer visible to the application developers, and any subscriptions to it are deleted. You can stage the Product again later if required; for details, see [Staging a Product](#).

## Related concepts

---

- [Using syndication in API Connect](#)

## Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Re-staging a Product

If a Product in a Catalog has been retired, you can re-stage the Product so that it can re-enter the Product lifecycle. A re-staged Product moves to the Staged state, from where it can be published. The syndication feature in IBM® API Connect means that you can also re-stage a Product within a Space in a Catalog.

## Before you begin

---

The Product that you are re-staging must be in the Retired state.

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## About this task



---


You stage a Product so that the appropriate approvals, internally within the organization, can be given for it to then be published. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#). For information on publishing a Product, see [Publishing a Product](#).

## Procedure

---

To re-stage a Product, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.The Products page of the Space opens, and all of the Products available in that Space are displayed.

3. Alongside the Product version that you want to work with, click the options icon  and then click Re-stage. The Re-stage Product window opens.
4. Click Confirm.

If approval is required to stage Products in this Catalog, an approval request is sent, and the Product moves to the Pending state; the Product is re-staged when the request is approved. If approval is not required, the Product is re-staged immediately. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#). For information on approving requests, see [Approving Product lifecycle and subscription requests](#).

Note:

- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

## Results

---

Your Product is in the Staged state, from where it can be published; see [Publishing a new Product](#).

## Related concepts

---

- [Using syndication in API Connect](#)

## Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Updating the gateway services for a Product

When you stage or publish a draft Product to a Catalog, you specify the gateway services at which the APIs in the Product are to be made available. However, you can later update the selected gateway services after a Product has been stage or published.

## Before you begin

---

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM® API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## About this task

---

When you stage or publish a draft Product to a Catalog, you can either choose to make the APIs in the Product available at all the gateway services that match the gateway service type of the Product, either DataPower® API Gateway or DataPower Gateway (v5 compatible), or can you select specific gateway services of that type. The APIs in the Product can then be called at those gateway service endpoints.

However, after a Product has been staged or published you can later update the gateway services to change the selections.




For more information on staging and publishing a Product, see [Staging a draft Product](#) and [Publishing a draft Product](#).

For more information on the gateway service types, see [API Connect gateway types](#).

## Procedure

---

To update the gateway services for a Product, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to work with, click the options icon , then click Update gateway services.  
The Select Gateway Services window opens.
4. Select the required gateway services, then click Save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing a Product from a Catalog

You can remove a Product from a specific Catalog by using the Products view within a Catalog in API Manager. The syndication feature in IBM® API Connect means that you can also use the Products view within a Space to remove a Product from a Catalog.

### Before you begin

---

The Product that you are removing from the Catalog must be in the Staged or Retired state. For more information, see [Staging a Product](#) or [Retiring a Product](#).




To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

### Procedure

---

To remove a Product from a Catalog, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Alongside the Product version that you want to work with, click the options icon , then click Delete.  
The Delete Product window opens.
4. Click Confirm to delete the Product.

### Results

---

Your Product is removed from the Catalog. You can stage the Product again if required; for details, see [Staging a Product](#).  
If your Product was contained within a Space in a Catalog, your Product is removed from both the Space and the Catalog.

### Related concepts

---

- [Using syndication in API Connect](#)

### Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Approving Product lifecycle and subscription requests

To approve or decline requests to change the lifecycle state of a Product, and requests by application developers to subscribe to a Plan, use the Tasks page in the Catalog that contains the associated Product in API Manager. The syndication feature in IBM® API Connect means that you can also use the Tasks page within a Space in a Catalog to approve or decline requests.

### Before you begin

---

To see lifecycle change requests for a Product, you must either be the owner of the API provider organization, or you must be assigned a user role that has permission to view or manage Products in the Catalog that contains the Product. For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in API Connect means that Products can be contained within a Space in a Catalog. In this case, to see lifecycle change requests for a Product, you must either be the owner of the API provider organization, or you must be assigned a user role that has permission to view or manage Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

### About this task

---

If approvals for Product lifecycle changes are enabled for a Catalog, then an attempt to change the lifecycle state of a Product results in an approval request being sent. This request is displayed in the Tasks page in the Catalog, from where the request can be approved or declined. The authority to approve Product lifecycle state changes is restricted to users in specified roles. For information on configuring Product lifecycle approvals for a Catalog, see [Creating and configuring Catalogs](#).

Note:




- Approval for Product lifecycle state changes in a Catalog is disabled by default. You must explicitly enable the Product lifecycle state changes that you want to enforce.
- Product lifecycle approvals can be configured only at the Catalog level. This feature is not available at the Space level.

If a Product is set to require approval for subscription by application developers, then an attempt by an application developer to subscribe to the Product results in an approval request being sent. This request is displayed in the Approvals tab in the Catalog that contains the associated Product in API Manager, from where the request can be approved or declined.




## Procedure

---

To work with a Product lifecycle change request, complete the following steps.

1. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
2. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
3. Click  Tasks in the API Manager UI navigation pane, then locate the Product lifecycle change request that you want to deal with.
4. Click the Approve or Decline as required.  
Note: You can approve or decline lifecycle change requests only if you have a user role that has permission to approve Product publish requests in the Catalog in which the changes will take place.  
For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

To work with a subscription request, complete the following steps.

5. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.  
The Products page of the Catalog opens, and all of the Products available in that Catalog are displayed.
6. If the Product that you want to work with is contained within a Space, select the required Space by completing the following steps
  - a. In the navigation pane of the API Manager UI, click  Spaces.
  - b. Select the Space that you want to work with.  
The Products page of the Space opens, and all of the Products available in that Space are displayed.
7. Click  Tasks, then locate the subscription requests that you want to deal with.  
Note: To see subscription approvals, you must either be the owner of the API provider organization, or you must be assigned a user role that has permission to view or approve subscription requests. For information on creating and assigning user roles, see [Administering user access](#).
8. Click the Approve or Decline as required.  
Note: The options to approve or decline a subscription request are available only if you are assigned a user role that has permission to approve subscription requests. For information on configuring permissions for a Catalog, see [Creating and configuring Catalogs](#) For information on assigning user roles, see [Managing Catalog membership](#).

## Results

---

For a subscription request, an email is sent confirming whether you approved or declined the request. For a request to change the lifecycle state of a Product, no email is sent.

## Related concepts

---

- [Using syndication in API Connect](#)

## Related tasks

---

- [Working with Spaces](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Managing the APIs in a Product

For a specific Product, you can take an individual API offline. If an API is taken offline then it becomes unavailable and calls to the API will fail.

## Before you begin

---

The Product must be in the Published or Deprecated state.

To complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Catalog that contains the Product. If you have View permission for Products, you have read-only access to the Product management page.

For information on configuring Product management permissions for a Catalog, see [Creating and configuring Catalogs](#).

The syndication feature in IBM® API Connect means that Products can be contained within a Space in a Catalog. In this case, to complete the Product management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned Manage permission for Products in the Space that contains the Product. For information on configuring Product management permissions for a Space, see [Managing user access in a Space](#).

## About this task

---





After a Product has been published to a Catalog, you can individually take any of its APIs offline so that the API is no longer available to be called. When required, you can make the API available again by putting it back online.


## Procedure


---

To manage the APIs in a Product, complete the following steps:

1. Locate the API that you want to work with:

- a. In the navigation pane of the API Manager UI, click  Manage, then select the Catalog that you want to work with.
- b. If Spaces are enabled in the Catalog, select the Space that you want to work with by completing the following steps:
  - i. In the navigation pane of the API Manager UI, click  Spaces.
  - ii. Select the Space that you want to work with.
- c. If the Products page is not already displayed, click  Products in the API Manager UI navigation pane.
- d. Alongside the Product version that you want to work with, click the options icon , then click Manage APIs. The Product APIs window opens, listing all the APIs in the Product.
- e. Locate the required API in the list.

2. To take an API offline, click the options icon  alongside the required API, then click Take Offline.

3. To make an API available that is currently offline, click the options icon  alongside the required API, then click Online.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## API Analytics

You can use IBM API Connect to filter, sort, and aggregate your API event data. You can present the results within correlated charts, tables, and maps to help you manage service levels, set quotas, establish controls, set up security policies, manage communities, and analyze trends.

API analytics is built on the Kibana open source analytics and visualization platform, which is designed to work with the Elasticsearch real-time distributed search and analytics engine.

Use the following tasks to configure and manage analytics:

- [Catalogs, Spaces, and Analytics](#)  
Understand how syndication and the use of Catalogs and Spaces affects the use of the features in IBM API Connect Analytics.
- [Accessing analytics](#)  
You can view predefined or customized analytics information for your IBM API Connect Catalogs within dashboards. If Spaces are enabled in your Catalogs, you can also view predefined or customized analytics information for your API Connect Spaces within dashboards.
- [The applications landing page](#)  
In IBM API Connect, each Catalog or Space has an applications landing page, which is a launching point for accessing the saved searches, dashboards, and visualizations that are available for that Catalog or Space. For more information about Catalogs and Spaces, see [Working with Catalogs](#) and [Using syndication in API Connect](#).
- [Dashboard application page](#)  
In IBM API Connect, each Catalog has its own default landing page, which lists the dashboards that are available for that Catalog.
- [Visualization application page](#)  
In IBM API Connect, you can use existing visualizations or create your own visualizations to organize the data on your dashboards.
- [Working with searches \(Discover\)](#)  
Use the IBM API Connect Analytics Discover tab to create and run searches against your analytics data.
- [Exporting API event data](#)  
You can obtain analytics and API event data from the API Manager user interface of IBM API Connect or by using REST API calls.
- [Troubleshooting Analytics start-up](#)  
If you receive an error starting the Analytics service after installing, upgrading, or recovering from a catastrophic failure of IBM API Connect, you might be able to resolve it by completing this task.

## Related information

---

-  [Elastic](#)
-  [Elasticsearch](#)
-  [Kibana](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



## Catalogs, Spaces, and Analytics

Understand how syndication and the use of Catalogs and Spaces affects the use of the features in IBM® API Connect Analytics.

The data for Analytics is collated from API events that are logged when API operations are invoked. Analytics data for an API is scoped to the Catalog or Space where that API resides, and is only included in search results (and thus, visualizations and dashboards) for the owning Catalog or Space. For example, you cannot create a single dashboard that includes data from multiple Catalogs.

Access to the analytics data, and to the analytics functions in the API Manager user interface, can be managed through the use of Catalogs and Spaces, and the roles and permissions that are assigned to the users (or *members*) of the provider organization.

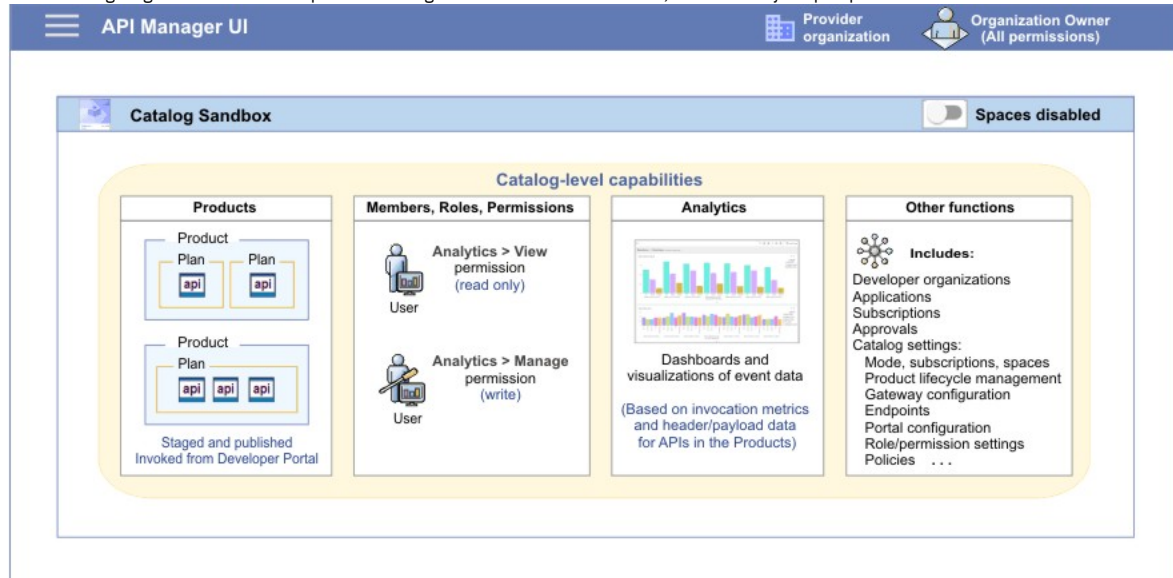
## Catalogs, Spaces, and permissions

Catalogs act as deployment targets through which APIs (in their containing Plans and Products) are staged and published to consumer organizations. API Manager users in the provider organization can be assigned the following access to the Analytics component for a Catalog:

- A role that has the `Analytics.>.View` permission for a Catalog: These users can view the analytics data generated for the APIs in the Catalog within *dashboards*, export dashboard data in its raw format or as event records, and apply filters to the data shown within the dashboards.
- A role that has the `Analytics.>.Manage` permission for a Catalog: These users have an implicit View permission. They can additionally complete the following actions:
  - Create, edit, and delete dashboards.
  - Create, edit, delete, export, and import the charts, tables, and maps.

For more information about creating and configuring Catalogs, and assigning roles and permissions to a Catalog, see [Working with Catalogs](#).

The following diagram shows an example of a Catalog with its associated functions, from an analytics perspective.



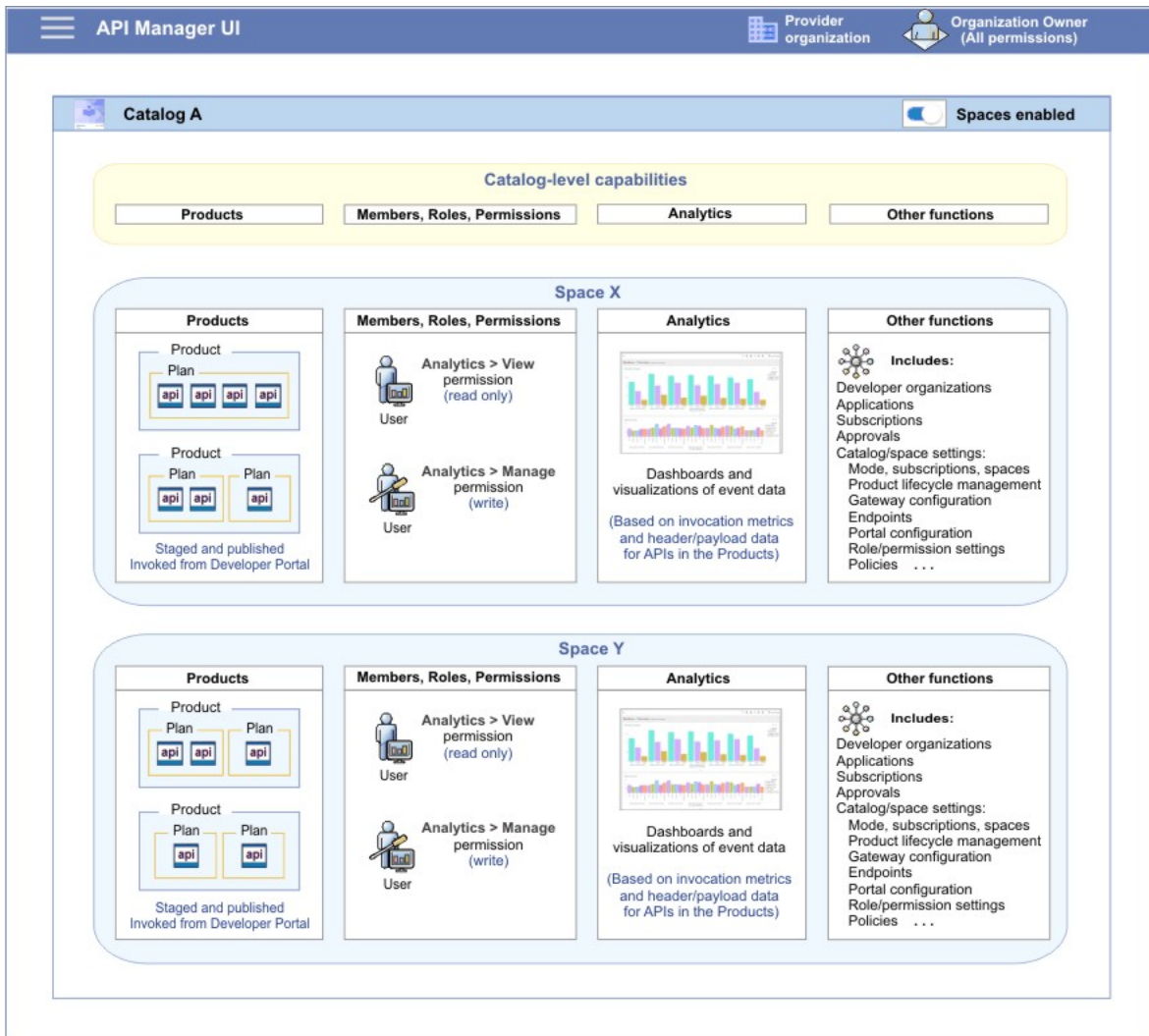
The IBM API Connect syndication feature provides a way for you to partition a Catalog into multiple deployment targets (or *Spaces*) through which separate groupings of APIs (in their containing Plans and Products) can be staged and published. Each Space can be allocated to a separate group of users who need to manage their Products independently, and the analytics data in each Space is scoped to those Products only. API Manager users in the provider organization can be assigned the following access to the Analytics component for a Space:

- A role that has the `Analytics.>.View` permission for a Space: These users can view the analytics data generated for the APIs in the Space within dashboards, export dashboard data in its raw format, and apply filters to the data shown within the dashboards.
- A role that has the `Analytics.>.Manage` permission for a Space: These users have an implicit View permission. They can additionally complete the following actions:
  - Create, edit, and delete dashboards.
  - Create, edit, delete, export, and import the charts, tables, and maps.

For information about enabling Spaces in a Catalog, setting up Spaces, and assigning roles and permissions to a Space, see [Using syndication in API Connect](#).

Note: Users can also be assigned a role with Catalog permissions, which provides a view of all Spaces (including analytics) in a Catalog.

The following diagram shows an example of a Catalog with its associated functions and Space partitions, from an analytics perspective.



For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

For more information about the default API Manager roles and permissions, see [API Connect user roles](#).

## Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces

When you create a Catalog, a set of default API Manager dashboards, visualizations, and searches are added to the Catalog. If you enable Spaces in the Catalog, a similar set of default dashboards, visualizations, and searches are added to each Space. You can also create custom dashboards, visualizations, and searches in the Catalog and its Spaces, if you have the required permission.

A layered implementation is used to define an "inheritance" flow for dashboards, visualizations, and searches in a Catalog and its Spaces. This structure determines whether updates made to the dashboards, visualizations, and searches in a Catalog are reflected in a Space, and affects what you see when you attempt to edit, delete, or restore default dashboards, visualizations, or searches, or when you attempt to create, edit, or delete custom dashboards, visualizations, or searches.

To help you identify the state of a dashboard, visualization, or search as default, custom, or inherited, tags are applied to dashboards and visualizations within the API Manager user interface. These tags are visible on the main Discover, Visualize, and Dashboard pages, which list your saved searches, dashboards, and visualizations.

Searches, dashboards, and visualizations

For the searches, dashboards, and visualizations, the following principles apply for create, edit, or delete operations:

Create

- When you create a search, dashboard, or visualization in a Catalog, it is automatically added to the Spaces in that Catalog.
- When you create a search, dashboard, or visualization in a Space, the search, dashboard, or visualization is added to that Space only.

Edit

- You cannot edit a default search, dashboard, or visualization. You must clone it, and then edit the clone.
- Searches, dashboards, and visualizations that are created in the Catalog are propagated down to the Spaces in that Catalog.
- You cannot modify a Catalog search, dashboard, or visualization in the Space of the Catalog. You must clone it to modify it.
- Searches, dashboards, and visualizations that are created or cloned in a Space are not propagated up to the Catalog.

Delete

- You cannot delete a default search, dashboard, or visualization.

- When you delete a search, dashboard, or visualization from a Catalog, it is also deleted from the Spaces within that Catalog.
- When you delete a search, dashboard, or visualization from a Space, it is not deleted from the Catalog.
- If a search, dashboard, or visualization in a Space was inherited from a Catalog, and has not been edited in the Space, you cannot delete that search, dashboard, or visualization from the Space. You must delete it from the Catalog.

How tags are applied to searches, dashboards, and visualizations in the Discover, Dashboard, and Visualize pages

The Discover, Dashboard, and Visualize pages, display tags to highlight the state of a search, dashboard or visualization as admin, catalog, or space. At the Catalog level, the following tags are applied to searches, dashboards or visualizations in the list to indicate their state:

- Each default search, dashboard or visualization, which has not been customized, is tagged with a **admin** label.
- Each search, dashboard or visualization that was created or cloned in the Catalog is tagged with a **catalog** label.
- Each custom dashboard or visualization that was created in the Space is tagged with a **space** label.

For information about accessing the Manage Saved Objects page from the API Manager UI, see:

- [Editing visualizations](#)
- [Exporting visualizations](#)
- [Importing visualizations](#)
- [Deleting custom visualizations](#)
- [Editing dashboards](#)
- [Exporting dashboards](#)
- [Importing dashboards](#)
- [Deleting custom dashboards](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Accessing analytics






You can view predefined or customized analytics information for your IBM® API Connect Catalogs within dashboards. If Spaces are enabled in your Catalogs, you can also view predefined or customized analytics information for your API Connect Spaces within dashboards.

### About this task

The analytics capabilities and functions are identical across Catalogs and Spaces. However, the analytics data is scoped to a Catalog when dashboards are accessed at the Catalog level, and the data is scoped to a Space when dashboards are accessed at the Space level. The ability to access analytics data at a Catalog or Space level depends on your assigned roles and permissions. For more information about Catalogs and Spaces, see [Working with Catalogs](#) and [Using syndication in API Connect](#).

Restriction: If the visualizations in your dashboards display no data or show data only for a time period in the past, a possible reason might be that access to analytics data within API Connect is switched off. Contact your cloud administrator for confirmation. (For more information, see [Configuring analytics offload for API Connect](#).)

### Procedure

- To open the analytics dashboard for any Catalog, complete the following steps:
  1. Log on to the API Manager that contains the Catalog for which you want to view the analytics.
  2. Select Manage  in the navigation.
  3. From the list of available Catalogs in the API Manager Dashboard, select the required Catalog.
  4. Select Analytics  in the navigation.  
The default analytics dashboard for the Catalog is displayed.
- To open the analytics dashboard for any Space, complete the following steps:
  1. Log on to the API Manager that contains the Catalog for which you want to view the analytics.
  2. Select Manage  in the navigation.
  3. From the list of available Catalogs in the API Manager Dashboard, select the required Catalog.
  4. Select Spaces  in the navigation.
  5. Select the Space that you want to view.
  6. Select Analytics  in the navigation.  
The default analytics dashboard for the Space is displayed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## The applications landing page

In IBM® API Connect, each Catalog or Space has an applications landing page, which is a launching point for accessing the saved searches, dashboards, and visualizations that are available for that Catalog or Space. For more information about Catalogs and Spaces, see [Working with Catalogs](#) and [Using syndication in API Connect](#).

You can navigate to the different applications pages by using the Discover, Visualize, and Dashboard applications icons on the Applications landing page. You can return to this page when you are anywhere within the Analytics navigation by selecting Apps in the breadcrumb trail.

The applications landing page for Catalogs or Spaces displays a table with the following information:

**Name**

The names of the items in the last application that you viewed. For example, if you last viewed dashboards, then it lists the dashboard titles.

**Tags**

Tags that identify at what level the item was created, and the required access level that is required to change them. The following tags are supported:

**admin**

This items is provided with IBM API Connect. You cannot modify items with the **admin** tag. You must clone them if you want to customize them.

**catalog**

This item was created by someone with Catalog management permissions. You can only modify dashboards with the **catalog** tag if you have Catalog management permissions, or higher. If you only have space permissions, then you must clone it to modify it.

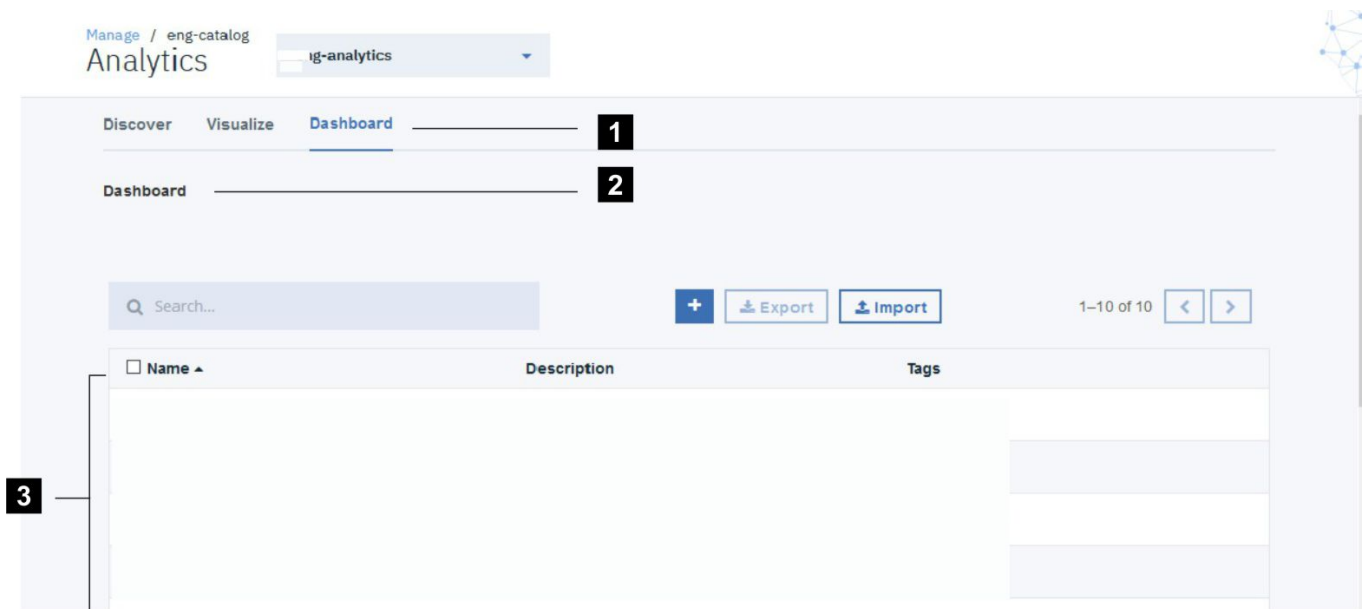
**space**

This dashboard was created by someone with space management permissions. You can only modify dashboards with the **space** tag if you have space management permissions, or higher.

## The screen elements of the applications landing page

The following image shows an example of the applications page of a Sandbox Catalog. While the screen elements of all of the application pages are similar, see the individual application page topics for specific information for each page:

- [Working with searches \(Discover\)](#)
- [Visualization application page](#)
- [Dashboard application page](#)



The screen elements are summarized in the following table.

Screen element	Description
1 Application selector	Indicates your selected application inside the analytics tab navigation. You can select a different application name to view the information for that application.
2 Breadcrumb trail	Indicates your location inside the analytics tab.
3 Instance information	Lists the instances of the last items that you viewed.

## Related tasks

- [Accessing analytics](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Dashboard application page

In IBM® API Connect, each Catalog has its own default landing page, which lists the dashboards that are available for that Catalog.

If Spaces are enabled in your Catalogs, each Space also has its own default landing page with the list of dashboards that are available for the space. For more information about Catalogs and Spaces, see [Working with Catalogs](#) and [Using syndication in API Connect](#).

Note: Either an incognito tab, a separate browser instance, or a full page refresh is required for viewing dashboards in multiple Analytics services simultaneously.

The dashboard landing page for Catalogs or Spaces displays a table with the following information:

**Name**

The name of the dashboard. The names of the default dashboards are based on the information that is contained within the dashboard.

**Description**

A brief description of the dashboard and its purpose.

**Tags**

Tags that identify at what level the dashboard was created, and the required access level that is required to change them. The following tags are supported:

**admin**

This dashboard is provided with IBM API Connect. You cannot modify dashboards with the **admin** tag. You must clone them if you want to customize them.

**catalog**

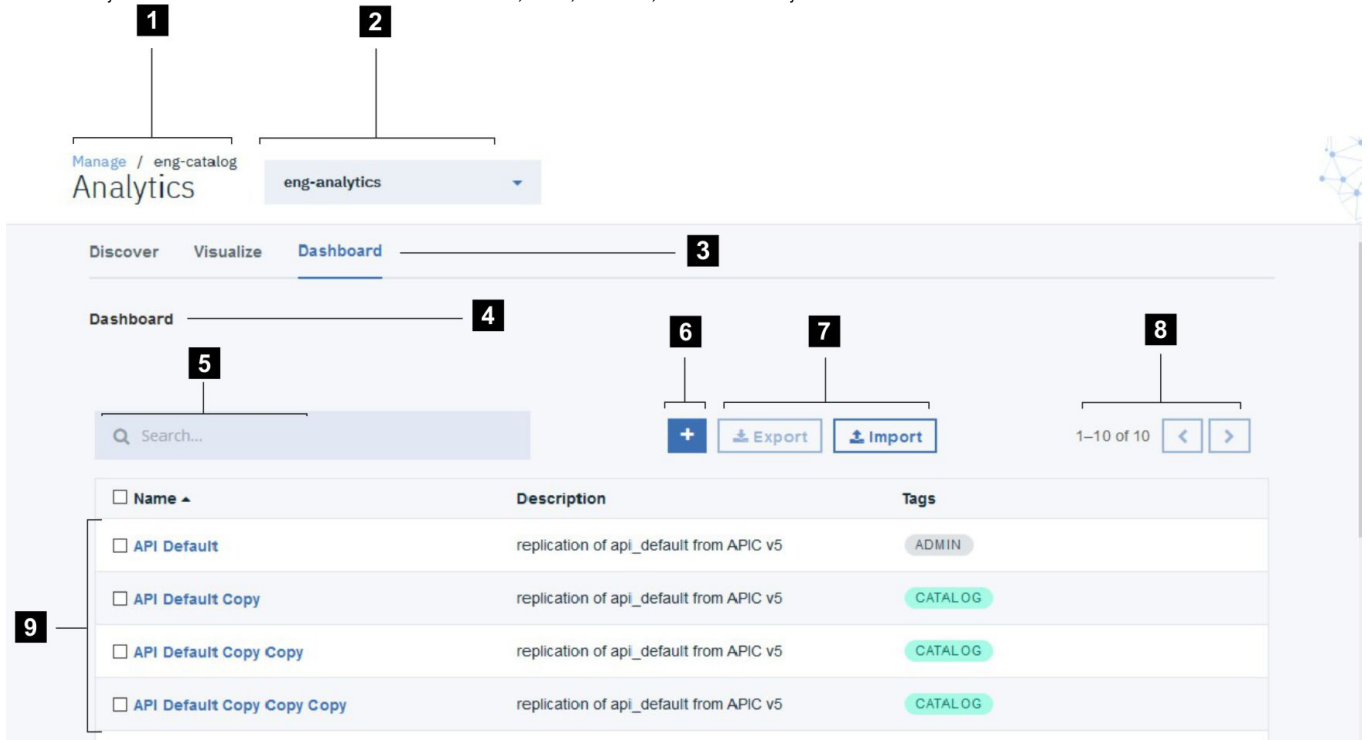
This dashboard was created by someone with catalog management permissions. You can only modify dashboards with the **catalog** tag if you have catalog management permissions. If you only have space permissions, then you must clone it to modify it.

**space**

This dashboard was created by someone with space management permissions. You can only modify dashboards with the **space** tag if you have space management permissions.

## The screen elements of the dashboard application page

The following image shows an example of the Overview dashboard for the Sandbox Catalog with sample data. Custom dashboards that you create for Catalogs or Spaces, and the Analytics dashboards for other entities such as Products, Plans, and APIs, have a similar layout.



The screen elements are summarized in the following table.

Screen element	Description
1 Catalog or Space name	Identifies the selected Catalog or Space.
2 Service identifier	Identifies the analytics service that you are viewing. If you only have one analytics service configured, there is only one possible value.
3 Application selector	Lists the applications that are available for this Catalog or Space. These are only displayed when Apps is selected in the breadcrumb trail.
4 Breadcrumb trail	Indicates your location inside the analytics tab navigation.
5 Search bar	Use this bar to filter the list of dashboards for the Catalog or Space.
6 Create a dashboard	Use this icon to create a customized dashboard. See <a href="#">Creating custom dashboards</a> .
7 Dashboard configuration icons	Use these icons to export dashboards that you want to reuse or back up, import previously exported dashboards to your available dashboards, or create dashboards.
8 Navigation icons	Use this icon to navigate through a large number of dashboards that require multiple screens. Each screen displays a maximum of 20 dashboards.
9 Dashboard information	Names and information about the dashboards for this Catalog or Space.

- **Default dashboards**  
IBM API Connect analytics provides some preconfigured dashboards for you to use to view common analytics data.
- **Cloning a dashboard**  
You can clone an existing dashboard to use it as a base for a customized dashboard.
- **Creating custom dashboards**  
You can use API Connect to create custom dashboards that group together a set of related visualizations.
- **Importing dashboards**  
You can import dashboards from other IBM API Connect users for use, or you can import dashboards from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import dashboards from a Space.

- [Editing dashboards](#)  
You cannot edit the IBM API Connect preconfigured dashboards, but you can clone it to create a custom dashboard to add, edit, or remove visualizations, and to resize or rearrange visualizations. You can also use the Time Picker to change the time range of the data shown in the visualizations.
- [Exporting dashboards](#)  
You can export dashboards for backing up the configuration, or so they can be imported into other Catalogs on your IBM API Connect system. If Spaces are enabled in a Catalog, exported dashboards can also be imported into a Space.
- [Backing up and restoring dashboards](#)  
To ensure you can recreate your IBM API Connect analytics dashboards in the event of data loss, you should export them and save the files as a back-up. Then you can import the files later if needed.
- [Deleting custom dashboards](#)  
If a custom dashboard is no longer required, you can permanently delete it from IBM API Connect.

## Related tasks

---

- [Accessing analytics](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Default dashboards

IBM® API Connect analytics provides some preconfigured dashboards for you to use to view common analytics data.

A list of dashboards is displayed when you open the default dashboards page for the first time. These dashboards provide examples of the data that you can view by using the analytics dashboards. You can use these dashboards as they are, or clone them to customize them to your needs. You cannot modify the existing dashboards. After you have cloned or created some of your own dashboards, your dashboards also appear in this list. The following dashboards are provided with API Connect:

### API Default

Includes general information about your APIs. This is the equivalent dashboard that was provided as api\_default in API Connect Version 5. It includes the following visualizations:

- Status Codes (detailed)  
Lists the detailed status codes of the API requests.
- Errors  
Displays the number of errors in a vertical bar graph.
- Minimum Response Time (ms)  
Lists the fastest response time of the APIs in milliseconds.
- Average response time (ms)  
Lists the average response time across all of the APIs in milliseconds.
- Maximum Response Time (ms)  
Lists the maximum response time of the APIs in milliseconds.
- Response Times (ms)  
Provides a complete list of the response times of the API calls in milliseconds.
- API calls  
Provides a list of the API calls.
- API Calls per day  
Provides a graphical representation of the number of API calls that occurred each day.

### Catalog Default

Includes general information about the most used Products in your Catalog. This is the equivalent dashboard for catalog\_default in API Connect Version 5. It includes the following visualizations:

- Top 5 Products Overall (Daily Usage)  
Displays a graph of your 5 Products that receive the most calls on a daily basis.
- Top 5 APIs Overall (Daily Usage)  
Displays a graph of your 5 APIs that receive the most calls on a daily basis.

### Monitoring Latency

Provides information about the amount of time that elapses after the API request is submitted and the transfer of data begins. It contains the following visualizations:

- Response Times (ms)  
Provides a complete list of the response times of the API calls in milliseconds.
- Response times (>1 sec)  
Displays the calls where the response times are greater than 1 second.  
Note: This visualization runs a search that limits the results to the last 500 calls.
- Average response time (ms)  
Lists the average response time across all of the APIs in milliseconds.
- Data Usage (bytes received)  
Lists the bytes of data that are received to complete your calls.
- Data Usage (bytes sent)  
Lists the bytes of data that are sent to complete your calls.
- Maximum Response Time (ms)  
Lists the maximum response time of the APIs in milliseconds.

## Monitoring Status

Provides information about monitoring the status of your API. This dashboard includes the following visualizations:

### Success Rate

Displays how many of your API calls were successful, compared to how many were submitted.

### Status Codes (simple)

Provides overview status codes for the API calls.

### API calls

Provides a list of the API calls.

### Errors

Displays the number of errors in a vertical bar graph.

### Successes

Lists the successful API calls.

Note: This visualization runs a search that limits the results to the last 500 calls.

## Portal Default

Provides information about the API requests to APIs in the Developer Portal. This is a replication of the portal\_default dashboard in API Connect Version 5. This dashboard includes the following visualizations:

### Success Rate

Displays how many of your API calls were successful, compared to how many were submitted.

### Data Usage (bytes sent)

Lists the bytes of data that are sent to complete your calls.

### Response Times (ms)

Provides a complete list of the response times of the API calls in milliseconds.

## Product Default

Provides information about the Products. This is a replication of the product\_default dashboard in API Connect Version 5. This dashboard includes the following visualizations:

### Developer Organizations

Lists the numerical value of the number of existing developer organizations.

### Apps per Plan

Displays a pie chart with the number of subscriptions to each Plan.

### API Calls per day

Provides a graphical representation of the number of API calls that occurred each day.

### API calls

Provides a list of the API calls.

### API Calls per day

Provides a graphical representation of the number of API calls that occurred each day.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Cloning a dashboard

You can clone an existing dashboard to use it as a base for a customized dashboard.

### About this task

---

You might want to customize a dashboard that is already created. Because the default dashboards that are provided with IBM® API Connect are read-only, you must clone a default dashboard to change it. You can clone any dashboard that you want to make additional changes to it without changing the original dashboard. To clone a dashboard, complete the following steps:

### Procedure

---

1. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards.
2. Select the dashboard that you would like to clone to open it.  
The dashboard opens, and the visualizations for that dashboard are displayed.
3. Select Clone in the menu bar to make a copy of the dashboard.
4. Enter a unique name for the new dashboard.
5. Select Confirm Clone to create the clone.  
The clone is created, and your new dashboard is opened.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating custom dashboards



You can use API Connect to create custom dashboards that group together a set of related visualizations.

## Before you begin

---

To create dashboards, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics, Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

## About this task

---




If Spaces are enabled in the selected Catalog, dashboards created at the Catalog level are added to all Spaces in the Catalog. However, if you create a dashboard within a Space, the dashboard is added to that Space only. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in [Catalogs, Spaces, and Analytics](#).

When you create a dashboard, you can add previously saved visualizations, and then arrange and resize them as required in the dashboard. If you want to add a new visualization that does not yet exist to the new dashboard, you must first create the visualization to make it available for selection from the list of saved visualizations. For more information, see [Creating visualizations](#).

## Procedure

---

To create a custom dashboard, complete the following steps:

1. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards.
2. Select the New Dashboard icon  to open an empty dashboard.
3. To add visualizations to this dashboard, complete the following steps:
  - a. Click Add on the dashboard to add a new visualization.  
The list of saved visualizations is displayed.
  - b. Add one or more saved visualizations to the dashboard as follows:
    - i. From the list of saved and default visualizations, select a visualization.  
Tip: If you have a large set of saved visualizations, you can either browse through the saved list or use the visualization filter.  
The visualization is displayed at the end of the dashboard. Depending on the number of visualizations that are already on the dashboard, you might not see the newly added visualization until you close the saved list as described in step 4.
    - ii. Select any other visualizations that you want to add.
4. Close the list of saved and default visualizations by clicking Add new Visualization.
5. Arrange the visualizations in the required order by using the move icon  to drag and drop the visualizations in the preferred location in the dashboard. You can see this icon in the upper right-hand corner when you hover over a visualization on the dashboard.
6. Resize the visualization containers by dragging outwards or downwards from the lower right-hand corner to increase the size, or dragging inwards or upwards to decrease the size.
7. Optional: Change the time period against which the data in the visualizations is scoped. By default, a Last 15 minutes filter is applied. You can specify a different filter by using the Time Picker icon .
8. To save the new dashboard, complete the following steps:
  - a. Click Save in the menu bar.
  - b. Specify a name for the dashboard in the Title text field.
  - c. Add a description of the dashboard in the Description field that helps you identify it.
  - d. If you want to save the dashboard with the time period that you specified in step 7, select the Store time with dashboard check box.
  - e. Click the Save button to exit edit mode and view the dashboard.

## Results

---

The dashboard is added to a list of saved dashboards and can be opened at any time if you want view, edit, or share it.

## Related tasks

---

- [Editing dashboards](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Importing dashboards

---

You can import dashboards from other IBM® API Connect users for use, or you can import dashboards from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import dashboards from a Space.

Dashboards can be imported from .json files that contain one or more exported dashboard definitions.

Note: JSON files that contain a combination of exported visualizations and dashboards can also be imported as described in the following steps.

## Before you begin

---



To import dashboards, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>Manage permission for the selected Catalog or Space. If you import a dashboard into a Catalog, the analytics data is scoped at the Catalog level, and if imported into a Space, the data is scoped to that Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

## Procedure

---

To import one or more dashboards, complete the following steps:

1. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards.  
The list of saved dashboards is displayed.
2. Select Import.
3. Navigate to the location where the .json file is stored and select the file to import it. If the file contains any dashboards that will overwrite existing ones on your system, new dashboards will be created even if they are duplicates.  
Remember that you cannot overwrite a default dashboard. If an imported dashboard has the same name as a default dashboard, it is saved with the same name but identified with a tag based on the whether it is a Catalog or Space dashboard.  
All dashboards in the .json file are added as individual objects to the Dashboards tab.

## Related tasks

---

- [Exporting dashboards](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Editing dashboards

You cannot edit the IBM® API Connect preconfigured dashboards, but you can clone it to create a custom dashboard to add, edit, or remove visualizations, and to resize or rearrange visualizations. You can also use the Time Picker to change the time range of the data shown in the visualizations.

## Before you begin

---

To edit dashboards, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

## About this task

---

If Spaces are enabled in the selected Catalog, when you edit the dashboard at the Catalog level, your changes are reflected in the Catalog, and are also propagated to the corresponding inherited dashboard in each Space provided the dashboard has not been edited in the Space. However, changes that you make to a dashboard in a Space will apply only within that Space. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in [Catalogs, Spaces, and Analytics](#).

Any visualizations that you want to add to the dashboard must already be in the list of saved visualizations. If you want to add a new visualization that does not yet exist, you must first create the visualization to make it available for selection from the list of saved visualizations. For more information, see [Creating visualizations](#).





## Procedure

---

To edit a dashboard, complete the following steps:

Remember: You cannot directly edit a default dashboard that is provided with IBM API Connect. See [Cloning a dashboard](#) for information about how to clone the dashboard and modify it.

1. Access the dashboard that you want to edit:
  - a. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards. The list of saved dashboards is displayed.
  - b. Select the dashboard that you want to edit to open it.  
If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboard.
  - c. Select Edit in the menu bar of the dashboard. The dashboard opens in edit mode.
2. Modify the contents as required:
  - Add an existing visualization to the dashboard:
    - Select Add to add a visualization. The list of saved visualizations is displayed.
    - Select the visualization you want to add.  
Tip: If you have a large set of saved visualizations, you can either browse through the list or use the visualization filter to locate the visualization. The visualization is added to the dashboard, within a container (but might be obscured by the saved list). You can resize or reposition the visualization as described later.
    - Close the list of saved visualizations by clicking Save.
  - Edit a visualization in the dashboard.  
Important: When you edit a visualization in this way, you must overwrite the original visualization in the saved list when prompted, in order to have the changes reflected in the dashboard you are editing. Be aware however, that your changes will also be reflected in any other dashboards that contain this visualization. (This might be undesired.)

- Hover over the visualization and then click the Edit icon  that is displayed in the upper right-hand corner of the visualization container. The visualization builder page opens.
  - Update and save the visualization configuration as described in [Configuring visualizations](#). When you close the visualization builder page, you return to the dashboard being edited.
  - Remove a visualization from the dashboard by hovering over the visualization and then clicking the Delete icon  that is displayed in the upper right-hand corner of the visualization container. The action only removes the visualization container from the dashboard; it does not delete the visualization itself from the saved list.
  - Resize a visualization container from its lower right-hand corner by dragging inwards or upwards to decrease the size, or dragging outwards or downwards to increase the size.
  - Rearrange the visualizations by using the move icon  to drag and drop the visualizations in the required location in the dashboard. You can see this icon in the upper right-hand corner when you hover over a container.
  - Change the time period against which the data in the visualizations is scoped. You can specify a different filter by using the Time Picker icon .
3. When your changes are complete, save the dashboard as follows:
- a. Click Save in the menu bar.
  - b. If you specified a time period as described in step 2, and you want to save the dashboard with that time filter, select the Store time with dashboard check box.
  - c. Click the Save button and then confirm that you want to overwrite the current dashboard. Your changes are reflected in the saved dashboard.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Exporting dashboards

You can export dashboards for backing up the configuration, or so they can be imported into other Catalogs on your IBM® API Connect system. If Spaces are enabled in a Catalog, exported dashboards can also be imported into a Space.

Dashboards are exported as .json files, which can then be imported. When you export, the filters that are currently applied are preserved in the exported file.

### Before you begin

To export dashboards, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>\_Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

### Procedure

To export one or more dashboards, complete the following steps:

1. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards. The list of saved dashboards is displayed.
2. From the Dashboards tab, export one, several, or all dashboards as follows:
  - To export a single dashboard, locate the dashboard, select its check box, and then select Export.
  - To export several dashboards, locate each dashboard and select its check box. Then select Export.
  - To export all dashboards, select the Select All check box at the beginning of the Name column. Then select Export.

Tip: If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboards.
3. Choose to save the file, which is named export.json by default. The file is saved to the download location that is configured for your browser. If you intend to export more than once, and you are exporting to the same location each time, you might want to rename each file with a meaningful name to help you differentiate between them.

### Related tasks

- [Importing dashboards](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Backing up and restoring dashboards

To ensure you can recreate your IBM® API Connect analytics dashboards in the event of data loss, you should export them and save the files as a back-up. Then you can import the files later if needed.

### About this task

API Connect does not back up dashboards, so it is good practice to create your own backups by exporting all of your dashboards.

This task only operates on the dashboard metadata, not on the analytics data.

## Procedure

---

1. Export your dashboards as explained in the topic, [Exporting dashboards](#).
2. If you need to restore the dashboards, import them as explained in the topic, [Importing dashboards](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deleting custom dashboards

If a custom dashboard is no longer required, you can permanently delete it from IBM® API Connect.

Note: You cannot delete the default dashboards.

## Before you begin

---

To delete custom dashboards, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

## About this task

---

If Spaces are enabled in the selected Catalog, when you delete the dashboard at the Catalog level, the corresponding inherited dashboard is also deleted from each Space if it has not been edited in the Space. If you delete a dashboard that was created in a Space and is unique to that Space, the dashboard is removed from that Space.

If a custom dashboard is inherited from a Catalog and has not been edited in a Space, you cannot delete that dashboard from the Space, and must delete it from the Catalog if necessary. Similarly, an inherited dashboard that has been updated in a Space also cannot be deleted from that Space, and must be deleted from the Catalog.



For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in [Catalogs, Spaces, and Analytics](#).

## Procedure

---

To delete one or more dashboards, complete the following steps:

1. If you are not already in the Dashboard application, select Dashboard to view the list of available dashboards.  
The list of saved dashboards is displayed.  
Tip: After the deletion, you are returned to the currently loaded dashboard, so it might be best to start this task from a dashboard that you do not intend to delete at the current time. If you delete the currently loaded dashboard, you are returned to the Overview dashboard by default, with the following error message displayed in the notification banner: `Could not locate that dashboard (id: dashboard_ID)`. You can click OK to close the message.
2. From the Dashboards tab, delete one or several dashboards as follows:

- To delete a single dashboard, locate the dashboard, select its check box, and then click the Delete icon .
- To delete several dashboards, locate each dashboard and select its check box. Then click the Delete icon .

Tips:

- If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboards.
- Each default dashboard is tagged with a **Admin** label. Default dashboards that have been customized are additionally tagged with either a **Catalog** or a **Space** label. Any dashboard that is tagged with a **Admin** label cannot be deleted.
- Any dashboard that is inherited from a **Catalog** must be deleted from the Catalog.

3. Confirm the deletion.

Note: If a default dashboard was selected for deletion, a message is displayed in the notification banner to indicate that default dashboards cannot be deleted.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Visualization application page

In IBM® API Connect, you can use existing visualizations or create your own visualizations to organize the data on your dashboards.

You can access the visualizations application page by selecting Analytics in the navigation for a Catalog. Select Apps\_>Visualize to open the Visualization page.

The visualization landing page displays a table with the following information:

Name

The name of the visualization. The names of the default visualizations are based on the information that is contained within the visualization. This list contains both preconfigured visualizations and customized visualizations.

Type

The format of the data in the visualization.

Tags

Tags that identify at what level the visualization was created, and the required access level that is required to change them. The following tags are supported:

admin

This visualization is provided with IBM API Connect. You cannot modify visualizations with the `admin` tag. You must clone them if you want to customize them.

catalog

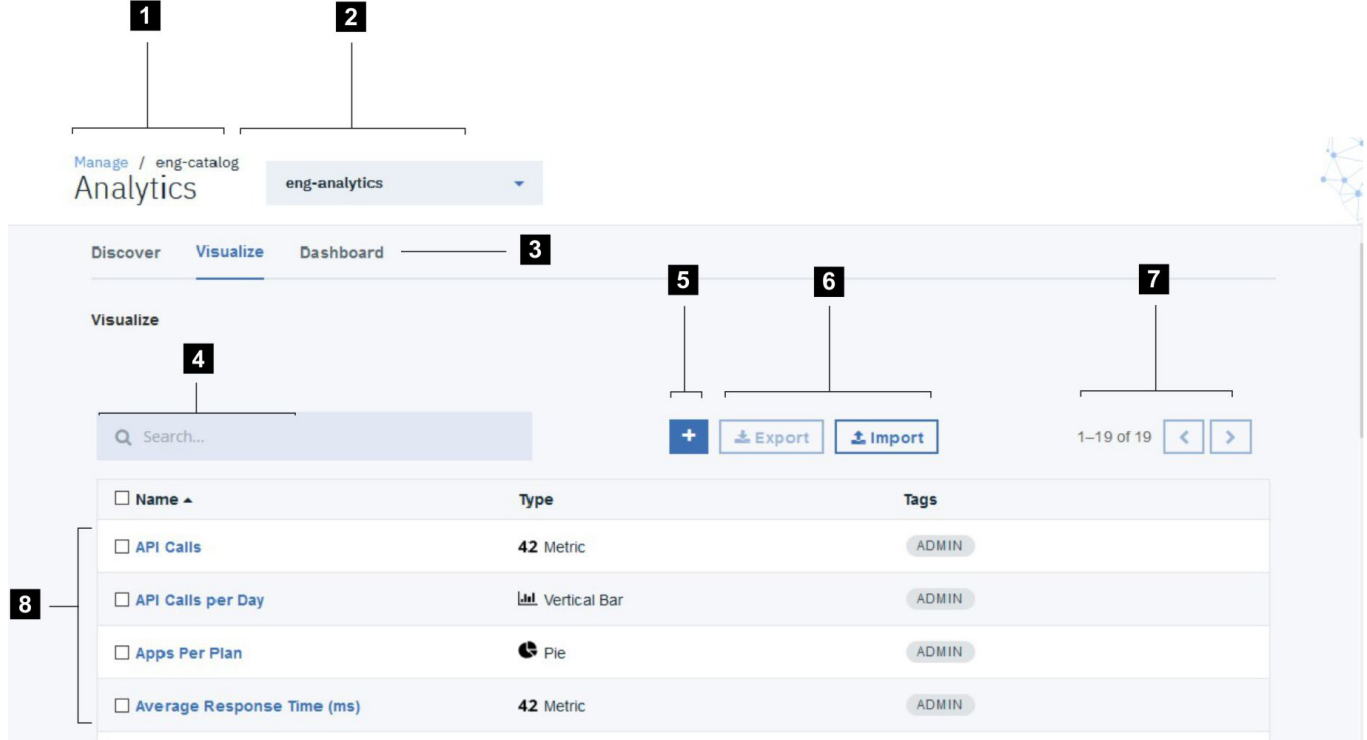
This visualization was created by someone with catalog management permissions. You can only modify visualizations with the `catalog` tag if you have catalog management permissions. If you only have space permissions, then you must clone it to modify it.

space

This visualization was created by someone with space management permissions. You can only modify visualizations with the `space` tag if you have space management permissions.

## The screen elements of the visualize application page

The following image shows an example of the visualizations for the Sandbox Catalog with sample data. Custom visualizations that you create for Catalogs or Spaces, and the Analytics visualizations for other entities such as Products, Plans, and APIs, have a similar layout.



The screen elements are summarized in the following table.

Screen element	Description
1 Catalog or Space name	Identifies the selected Catalog or Space.
2 Service identifier	Identifies the analytics service that you are viewing. If you only have one analytics service configured, there is only one possible value.
3 Application selector	Indicates your selected application inside the analytics tab navigation. You can select a different application name to view the information for that application.
4 Search bar	Use this bar to filter your visualizations for the Catalog or Space.
5 Create a visualization	Use this icon to create a customized visualization. See <a href="#">Creating visualizations</a> .
6 Visualization configuration icons	Import or export your visualization to integrate a visualization or share one.
7 Navigation icons	Use these icons to navigate through a large number of visualizations that require multiple screens. Each screen displays a maximum of 20 visualizations.
8 Visualization information	Names and information about the saved visualizations for this Catalog or Space.

- [Visualizations](#)  
Visualizations provide a way for you to apply a series of search criteria to your indexed data, compute metrics, and then graphically present the results in a convenient format for analysis or review. IBM API Connect analytics provides some preconfigured visualizations for you to use to view common analytics data on your dashboards. You can also create your own visualizations.
- [Applying filters to change the sampling of data displayed in visualizations](#)  
You can apply different types of filters to change your view of the data in your IBM API Connect visualizations.
- [Creating visualizations](#)  
You can create visualizations from the Visualize application page or when modifying a custom dashboard in IBM API Connect. All visualizations that you create are added to a list of saved visualizations, and can be used in any dashboard.
- [Importing visualizations](#)  
You can import visualizations from other IBM API Connect users for use in your dashboards, or you can import visualizations from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import visualizations from a Space.

- [Exporting visualizations](#)  
You can export visualizations so they can be imported by other IBM API Connect users, or into other Catalogs on your system. If Spaces are enabled in a Catalog, exported visualizations can also be imported into a Space.
- [Backing up and restoring visualizations](#)  
To ensure you can recreate your IBM API Connect analytics visualizations in the event of data loss, you should export them and save the files as a back-up. Then you can import the files later if needed.
- [Deleting custom visualizations](#)  
If no longer required for use in your IBM API Connect dashboards, you can permanently delete your custom visualizations.

## Related tasks

---

- [Accessing analytics](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Visualizations

Visualizations provide a way for you to apply a series of search criteria to your indexed data, compute metrics, and then graphically present the results in a convenient format for analysis or review. IBM® API Connect analytics provides some preconfigured visualizations for you to use to view common analytics data on your dashboards. You can also create your own visualizations.

## Default visualizations

---

A list of visualizations is displayed when you select Visualize in the application selector and open the visualization application page for the first time. These are preconfigured visualizations of some common ways to parse and view analytics data. You can use these visualizations as they are, or clone them to customize them to your needs. You cannot modify the existing visualizations. After you have cloned or created some of your own visualizations, your visualizations also appear in this list. The following default visualizations are provided with API Connect:

### API calls

Provides a list of the API calls.

### API Calls per day

Provides a graphical representation of the number of API calls that occurred each day.

### Apps per Plan

Displays a pie chart with the number of subscriptions to each Plan.

### Average response time (ms)

Lists the average response time across all of the APIs in milliseconds.

### Data Usage (bytes received)

Lists the bytes of data that are received to complete your calls.

### Data Usage (bytes sent)

Lists the bytes of data that are sent to complete your calls.

### Developer Organizations

Lists the numerical value of the number of existing developer organizations.

### Errors

Displays the number of errors in a vertical bar graph.

### Maximum Response Time (ms)

Lists the maximum response time of the APIs in milliseconds.

### Minimum Response Time (ms)

Lists the fastest response time of the APIs in milliseconds.

### Response Times (ms)

Provides a complete list of the response times of the API calls in milliseconds.

### Response times (>1 sec)

Displays the calls where the response times are greater than 1 second.

Note: This visualization runs a search that limits the results to the last 500 calls.

### Status Codes (detailed)

Lists the detailed status codes of the API requests.

### Status Codes (simple)

Provides overview status codes for the API calls.

### Subscribed Apps

Lists the number of subscribed apps.

### Success Rate

Displays how many of your API calls were successful, compared to how many were submitted.

### Successes

Lists the successful API calls.

Note: This visualization runs a search that limits the results to the last 500 calls.

### Top 5 APIs Overall (Daily Usage)

Displays a graph of your 5 APIs that receive the most calls on a daily basis.

### Top 5 APIs Overall (Overall Usage)

Displays a graph of your 5 APIs that receive the most calls..

### Top 5 Products Overall (Daily Usage)

Displays a graph of your 5 Products that receive the most calls on a daily basis.

### Top 5 Products Overall (Overall Usage)

Displays a graph of your 5 Products that receive the most calls.

## Created visualizations

You cannot directly modify the default visualizations. You can clone and modify them, or create your own from scratch. The visualizations that you clone or create are added to the list of default visualizations. The created visualizations are designated with a **Catalog** or a **Space** tag.

API Connect supports the following types of visualizations:

- Area chart
- Bar chart
- Coordinate map
- Data table
- Gauge
- Goal
- Heat map
- Line chart
- Markdown widget
- Metric
- Pie chart
- Region map
- Tag cloud
- Tile map
- Vertical bar chart

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Applying filters to change the sampling of data displayed in visualizations

You can apply different types of filters to change your view of the data in your IBM® API Connect visualizations.

- [Filtering API events in your visualizations](#)
- [Specifying a time period and auto-refresh rate for the data in your visualizations](#)
- [Drilling down into data in your time-based visualizations](#)
- [Applying a filter by using the legend in a visualization](#)

## Filtering API events in your visualizations

In a dashboard, the visualizations depict information that relates to all the API events that occur for the selected Catalog in the IBM API Connect organization, and which are scoped to a specified time range. You can further filter the API events by using a free text search or a Lucene search query.

### Procedure

To filter the API events, complete the following steps:

1. To perform a free text search, delete the asterisk character (\*) if shown, and then enter a text string in the search bar. For example, to filter for an API called **accounts**, enter `accounts` in the search bar.

Then, press Enter or click the Search icon .

The visualizations are refreshed to show the results of your search query.


2. To use the Lucene query syntax, complete the following steps:


- a. In the search bar, delete the asterisk character (\*) if shown, and then enter a search string.

For basic search queries, use the following guidelines:

Guideline	Example
Construct searches on field names and their actual values by using this syntax: <code>field_name:value</code> . Field names and their values are case-sensitive.	<code>app_name:legacyquote</code>
Where required for numeric fields, use comparison operators such as greater than (>), less than (<), or equal to (=).	<code>rateLimit.count:&gt;10</code>
Use the logic operators AND, OR, and NOT to combine search terms.	<code>app_name:livequote OR app_name:legacyquote</code>

For detailed information about the Lucene query syntax, see [Apache Lucene - Query Parser Syntax](#). For information about the API event fields that you can specify in search queries, see [API event record fields](#).

Tip: Two quick ways to check for the field names on certain visualizations are to check the tooltip, or to click the caret icon  (which is displayed in the lower left-hand corner when you hover over the container) to display the raw data behind the visualization.

- b. Press Enter or click the Search icon .

The visualizations are refreshed to show the results of your search query.


3. To revert to your previous view of the dashboard, delete the search query in the search bar and press Enter.

## Specifying a time period and auto-refresh rate for the data in your visualizations

You can change the time period to which your visualization data relates by using the Time Picker. The defined time filter will be applied to all relevant visualizations in the IBM API Connect dashboard.

### Procedure

To apply a time filter and auto-refresh rate, complete the following steps:

1. From the dashboard, click the Time Picker icon .
2. From the Time Picker, use one of these options to set a time filter:

#### Quick

Select this option to choose a predefined value such as Today, Yesterday, This week, or Last *n* minutes.




#### Relative

Select this option to specify a start time that is relative to now; for example, 20 seconds, minutes, hours, days, weeks, or months ago, optionally rounded to the specified unit of time. The date and time that corresponds to your "relative" selection is displayed before the fields. Click Go.

#### Absolute

Select this option to specify a precise time range. Either use the calendars to select a From and To date, or enter the values directly into the fields by using the date and time format specified underneath the fields. Click Go.

Notice that Auto-refresh is also shown to the left-hand of the Time Picker icon when you open the Time Picker.

3. If you want to additionally specify a frequency at which the data should automatically be refreshed in your visualizations, click Auto-refresh and then select a predefined refresh interval.
4. If you set an auto-refresh interval as described in the previous step, click the auto-refresh value, which is displayed next to the Time Picker icon , to confirm your settings and close the time selection panels. If you did not set an auto-refresh interval, close the Time Picker panel by clicking within the box where the Time Picker icon  is located. The search query is resubmitted as you make your selections and the visualizations in the dashboard are automatically refreshed to show the matching data. The specified quick, relative, or absolute filter setting is shown to the right-hand side of the Time Picker icon . If set, the auto-refresh interval is shown to the left-hand side of the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.  
Tip: To switch off the auto-refresh capability, click the auto-refresh value next to the Time Picker icon, and click Off. Then, close the Refresh Interval panel by clicking within the Auto-refresh box next to the Time Picker icon.

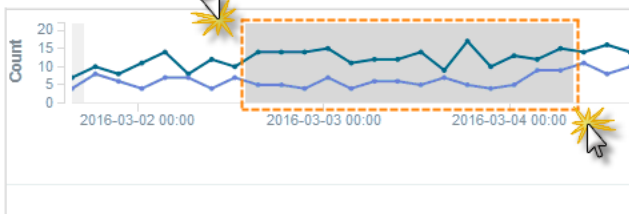
## Drilling down into data in your time-based visualizations


On time-based visualizations that display histograms, you can zoom in on a specific time range on the chart. This is equivalent to applying a filter for an "absolute" time period.

### Procedure

To zoom in on your data, complete one of the following steps:

- In the chart, move your cursor to the area that depicts the start time, and hover over the x-axis so that the cursor changes to a plus (+). Click and then drag the mouse to select a boxed area that depicts the time range you want to examine. Release the mouse button to zoom in on the area and view the data in greater detail.



The filter is applied to all time-based visualizations in the current dashboard, and the start and end time range is shown to the right-hand side of the Time Picker icon .

Tip: To remove the filter, click the browser Back button. Alternatively, use the Time Picker icon to select the previous, or a different time range.

- Bar charts only: To zoom in on a particular bar, click that bar in the chart, and then in the filter banner that is displayed underneath the search bar, click Apply Now.

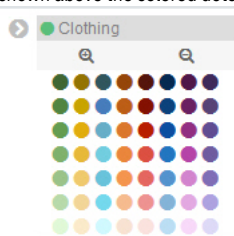
## Applying a filter by using the legend in a visualization

For any chart that includes a legend, you can use the legend labels to apply inclusion (or *positive*) filters to that chart and other relevant charts on the dashboard. When you use the legend in this way to specify inclusion, data is displayed only for the selected item. You can alternatively apply exclusion (or *negative*) filters to exclude the data for a selected item. You can also apply other filter conditions in addition to inclusion and exclusion filters.


### Procedure

To apply a filter from the legend of a chart such as an area chart, line chart, pie chart, or vertical bar chart, complete the following steps:


1. In the chart, click a legend label to apply a filter for that item.  
The color picker opens as displayed in the following example, which shows the label and color picker for a Product named **Clothing**. Notice that two icons are also shown above the colored dots.



Tip: You can change the color of the lines, slices, or bars on a chart by clicking the colored dots in the color picker to select a preferred color. To retain these colors, you will have to save the dashboard.

2. Apply an inclusion or exclusion filter as follows:
  - To apply an inclusion filter that shows data associated with that item only (for example, the **Clothing** Product), click the Positive Filter icon . A blue inclusion filter oval and an Actions twistie are displayed beneath the search bar of the dashboard, as shown in the following example. The filter condition (which in this case is `product_name: "Clothing"`) is also shown in the filter oval.

product\_name: "Clothing" Actions ▶

- To apply an exclusion filter that excludes data about that item from the chart, click the Negative Filter icon . A red exclusion filter oval and an Actions twistie are displayed beneath the search bar of the dashboard, as shown in the following example. The filter condition (for example, `product_name: "Clothing"`) is also shown in the filter oval.




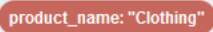


product\_name: "Clothing" Actions ▶

3. Hover over the filter oval to view the filter icons and use these icons to apply other filter conditions.

The following image shows an example of the filter icons for an inclusion filter, and also shows the Actions twistie in an expanded state. The Actions twistie provides equivalent options for the icons. For an exclusion filter, similar icons are shown in a red oval.



All filters: Enable Disable Pin Unpin Invert Toggle Remove

Icon	Description	Equivalent Actions option
	Click this icon to temporarily disable the inclusion or exclusion filter that was set in step 2. When you disable an inclusion filter, the filter oval is shown in a blue striped color, and the chart displays all the data again. (For a disabled exclusion filter, the filter oval is shown in a red striped color.) To enable the filter again, click the icon again.	Disable Enable  Toggle is an alternative for Enable/Disable.
	Click this icon to pin the filter. A pin icon is shown to the left-hand side of the filter condition in the blue (or red) oval as an indication. Pinning a filter causes it to be applied to other dashboards that you open. To unpin the filter, hover over the oval and click the pin icon again.	Pin Unpin
	Click this icon to toggle the filter to instead show excluded items (that is, those items that do <i>not</i> match the filter condition). The toggle action switches between applying an inclusion filter and an exclusion filter to the data in the visualization. When the exclusion filter is applied, the filter oval switches from blue to red  .	Invert
	Click this icon to remove the filter and restore the chart to its original state.	Remove
	Click this icon to display the JSON representation of the filter. If required, you can directly modify the JSON code to change your filter query and then specify an alias that can be used as the filter name. To see an example of how to use this option, see the <a href="#">Edit Filter</a> section within the <i>Filtering by Field</i> topic in the Kibana documentation.	

Example of how an inclusion filter works when enabled: Suppose the filter condition `product_name: "Clothing"` is applied to the first visualization, which shows the five most active Products. A second visualization in the same dashboard, which shows the five most active APIs, would automatically be refreshed to display data only for those APIs in the Product named `Clothing`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating visualizations

You can create visualizations from the Visualize application page or when modifying a custom dashboard in IBM® API Connect. All visualizations that you create are added to a list of saved visualizations, and can be used in any dashboard.

### Before you begin


To create visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the `Analytics:manage` permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

### About this task

If Spaces are enabled in the selected Catalog, visualizations created at the Catalog level are added to all Spaces in the Catalog. However, if you create a visualization within a Space, the visualization is added to that Space only. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in [Catalogs, Spaces, and Analytics](#).

To create a visualization from a new dashboard page, complete the following steps:

### Procedure

- Click the New Dashboard icon  to create a new dashboard.
- Select Add to prepare to add visualizations.
- Select Add new Visualization.
- From the Select visualization type page, choose a visualization type, and then configure and save it as described in [Configuring visualizations](#). You return to the new dashboard page.

### Results



The new visualization is added to the list of saved visualizations and can be selected for use.

- [Configuring visualizations](#)  
Use this information either when creating a visualization or when editing an existing visualization.
- [Editing visualizations](#)  
You can edit a visualization either from a dashboard that contains that visualization or by managing your saved objects.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



## Configuring visualizations

Use this information either when creating a visualization or when editing an existing visualization.

### Before you begin

---

As a starting point for this task, you must have completed one of the following actions:

- Clicked Add, and then Create new Visualization in a new dashboard page to open the Select visualization type page. (In this case, start from step [1](#) in the procedure that follows.)
- Clicked Edit from an existing dashboard, and then clicked Add to open the Create new Visualization page. (In this case, start from step [1](#) in the procedure that follows.)
- Clicked the Edit icon  on a visualization container on an existing dashboard, to open the visualization builder page. (In this case, start from step [2](#) in the procedure that follows.)
- Clicked the Edit icon  for a visualization to open the visualization builder page. (In this case, start from step [2](#) in the procedure that follows.)

To configure visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics > Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

### About this task

---

When you configure visualizations, you use the following Elasticsearch aggregations to define the type and level of information to be retrieved and displayed:

- metrics: You can configure one or more metric aggregations that will calculate metrics based on values extracted from the indexed data fields. (Indexed fields are searchable and are available for use in visualizations.)
- buckets: Buckets operate in a similar way to SQL GROUP BY statements and enable aggregate functions to be performed on a filtered data set. You can configure one or more bucket aggregations that will sort your data according to the criteria specified.

For detailed information about aggregations, see the [Elasticsearch aggregations reference](#). For information about the indexed data fields that can be specified while configuring visualizations, see [API event record fields](#).

You can also specify view options for each visualization type; for example, the capability to show or hide the tooltip and legend.

**Note:** The API Manager analytics component loads with a preselected default index pattern, which identifies the index against which search and analytics are run, and which scopes the Kibana queries to the default Catalog. To ensure tenant and Catalog isolation, this default index pattern cannot be customized by users.

### Procedure

---

To configure a visualization, complete the following steps:

1. From the Create New Visualization page, choose the type of visualization that you want to create:
  - Area chart
  - Bar chart
  - Coordinate map
  - Data table
  - Gauge
  - Goal
  - Heat map
  - Line chart
  - Markdown widget
  - Metric
  - Pie chart
  - Region map
  - Tag cloud
  - Tile map
  - Vertical bar chart


Tip: If necessary, review the guidance on the page to help you decide which type you need.



2. From the visualization builder page, define the content and layout for the visualization you are creating or editing.  
You can configure your settings by using the visualization builder controls in the left-hand pane, and then view the results of your actions in the preview canvas on the right.
  - a. From the Data tab in the visualization builder, configure the metric and bucket aggregations for the visualization.
  - b. From the Options tab in the visualization builder, specify view options for the visualization.

For full details about completing the visualization builder for your selected visualization, see the following information in the Kibana documentation:

- [Area Charts](#)
- [Bar Charts](#)
- [Coordinate Maps](#)
- [Data Table](#)
- [Gauge Chart](#)
- [Goal Charts](#)
- [Heat Map Chart](#)
- [Line Charts](#)
- [Markdown Widget](#)
- [Metric](#)
- [Pie Charts](#)
- [Region Maps](#)
- [Timelion](#)
- [Tag Clouds](#)
- [Vertical Bar Charts](#)

For additional information about visualizations and aggregations, see the [Creating a Visualization](#) section in the Kibana documentation.

As you configure each setting in the visualization builder, you can click Apply changes  to view the results of your action within the preview canvas, or click

Discard changes  to undo a change. You can also click the Refresh icon  to refresh the visualization preview. If you are creating a complex visualization, you might find it useful to save and name the visualization (as described in step 3) after the initial metric or bucket aggregations are configured, and then save in stages as you configure more aggregations for the visualization.

3. When the configuration is complete, save the visualization:
  - a. Click Save.
  - b. Specify a name for the visualization in the text field (if not already specified).
  - c. Click the Save button. (If you have previously saved, confirm that you want to overwrite.)
4. Select Analytics in the navigation to exit the visualization builder page.

You return to the previous dashboard.

  - If you created a visualization, it is added to the saved list.
  - If you edited an existing visualization, your changes are reflected in any dashboards that contain that visualization.

## Related tasks

---

- [Creating custom dashboards](#)
- [Creating visualizations](#)
- [Editing dashboards](#)
- [Editing visualizations](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Editing visualizations

You can edit a visualization either from a dashboard that contains that visualization or by managing your saved objects.

### Before you begin

---

To edit visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

### About this task

---

If Spaces are not enabled in the selected Catalog, all dashboards that contain the modified visualization will be updated to reflect the changes made for that visualization. If Spaces are enabled in the selected Catalog, when you edit a visualization at the Catalog level, your changes are reflected in all dashboards (which contain that visualization) in the Catalog. Your changes are also reflected in the corresponding visualization in each Space if the visualization was inherited from the Catalog and has not been edited in the Space. However, changes that you make to a visualization in a Space will apply only within that Space. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in [Catalogs, Spaces, and Analytics](#).


### Procedure

---


To edit a visualization, complete either of the following steps:

- Edit a visualization in a dashboard.

Important: When you edit a visualization in this way, you must overwrite the original visualization in the saved list when prompted, in order to have the changes reflected in the dashboard you are editing. Be aware however, that your changes will also be reflected in any other dashboards that contain this visualization. (This might be undesired.)

  1. From a dashboard that contains the visualization you want to edit, locate the visualization, hover over it, and then click the Edit icon  that is displayed in the upper right-hand corner of the container. The visualization builder page opens.
  2. Update and save the visualization configuration as described in [Configuring visualizations](#).

When you close the visualization builder page, you return to the dashboard and can see the updated visualization.

- Edit a visualization by managing your saved objects.
  1. From a dashboard, click Edit, then Add.  
The list of saved visualizations is displayed.
  2. Click Manage Visualizations to open the Manage Saved Objects page.  
This page contains separate tabs that show a listing of all your saved dashboards and visualizations, and the total counts.
  3. From the Visualizations tab, locate the visualization you want to edit.  
Tip: If you have a large set of saved visualizations, you can either browse through the list or use the filter to locate the visualization.
  4. Click the Edit button  for the visualization. The visualization builder page opens.
  5. Update and save the visualization configuration as described in [Configuring visualizations](#).  
When you close the visualization builder page, you return to the previous dashboard. If you now open any dashboard that contains the edited visualization, your changes will be reflected in that dashboard.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Importing visualizations

You can import visualizations from other IBM® API Connect users for use in your dashboards, or you can import visualizations from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import visualizations from a Space.

Visualizations can be imported from .json files that contain one or more exported visualization definitions.

Note: JSON files that contain a combination of exported visualizations and dashboards can also be imported as described in the following steps.

### Before you begin

To import visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_...\_Manage permission for the selected Catalog or Space. If you import a visualization into a Catalog, the analytics data is scoped at the Catalog level, and if imported into a Space, the data is scoped to that Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

### Procedure

To import one or more visualizations, complete the following steps:

1. From a dashboard, click Edit, then Add.  
The list of saved visualizations is displayed.
2. Click Manage Visualizations to open the Manage Saved Objects page.  
This page contains separate tabs that show a listing of all your saved dashboards and visualizations, and the total counts. The Visualizations tab is displayed by default.
3. Click the Import button.
4. Navigate to the location where the .json file is stored and select the file to import it. If the file contains any visualizations that will overwrite existing ones on your system, new visualizations will be created even if they are duplicates.  
All visualizations in the .json file are added as individual objects to the Visualizations tab.
5. Click the Close button to close the page and return to the dashboard.

### Related tasks

- [Exporting visualizations](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Exporting visualizations

You can export visualizations so they can be imported by other IBM® API Connect users, or into other Catalogs on your system. If Spaces are enabled in a Catalog, exported visualizations can also be imported into a Space.

Visualizations are exported as .json files, which can then be imported.

### Before you begin

To export visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>\_Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

---

## Procedure

To export one or more visualizations, complete the following steps:

1. From a dashboard, click Edit, then Add.  
Tip: You can also start from the Visualize application page by selecting Visualize.  
The list of saved visualizations is displayed.
2. From the Visualizations tab, export one, several, or all visualizations as follows:
  - To export a single visualization, locate the visualization, select its check box, and then click Export.
  - To export several visualizations, locate each visualization and select its check box. Then click Export.
  - To export all visualizations, select the Select All check box and then click Export.Tip: If you have a large set of saved visualizations, you can either browse through the list or use the filter to locate the visualizations.
3. Choose to save the file, which is named export.json by default. The file is saved to the download location that is configured for your browser. If you intend to export more than once, and you are exporting to the same location each time, you might want to rename each file with a meaningful name to help you differentiate between them.

---

## Related tasks

- [Importing visualizations](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Backing up and restoring visualizations

To ensure you can recreate your IBM® API Connect analytics visualizations in the event of data loss, you should export them and save the files as a back-up. Then you can import the files later if needed.

---

## About this task

API Connect does not back up visualizations, so it is good practice to create your own backups by exporting all of your visualizations.

This task only operates on the visualization metadata, not on the analytics data.

---

## Procedure

1. Export your visualizations as explained in the topic, [Exporting visualizations](#).
2. If you need to restore the visualizations, import them as explained in the topic, [Importing visualizations](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting custom visualizations

If no longer required for use in your IBM® API Connect dashboards, you can permanently delete your custom visualizations.

Note: You cannot delete the default visualizations.

---

## Before you begin

To delete custom visualizations, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics\_>\_Manage permission for the selected Catalog or Space. For information about enabling Spaces in a new or existing Catalog, see [Creating and configuring Catalogs](#) and [Enabling Spaces in a Catalog](#). For more information about assigning Catalog or Space permissions to a role, see [Managing Catalog membership](#) and [Managing Space membership](#).

---

## About this task



If Spaces are enabled in the selected Catalog, when you delete a visualization at the Catalog level, the corresponding inherited visualization is also deleted from each Space if it has not been edited in the Space. If you delete a visualization that was created in a Space and is unique to that Space, the visualization is removed from that Space.

If a custom visualization is inherited from a Catalog and has not been edited in a Space, you cannot delete that visualization from the Space, and must delete it from the Catalog if necessary. Similarly, an inherited visualization that has been updated in a Space also cannot be deleted from that Space, and must be deleted from the Catalog.

## Procedure

To delete one or more visualizations, complete the following steps:

1. From a dashboard, click Edit, then Add.  
Tip: You can also start from the Visualize application page by selecting Visualize. The list of saved visualizations is displayed.
2. From the Visualizations tab, delete one or several visualizations as follows:

- To delete a single visualization, locate the visualization, select its check box, and then click the Delete icon .
- To delete several visualizations, locate each visualization and select its check box. Then click the Delete icon .

Tips:

- If you have a large set of saved visualizations, you can either browse through the list or use the filter to locate the visualizations.
- Each default visualization is tagged with a **Admin** label. Default visualizations that have been customized are tagged with a **Catalog** or **Space** tag. Any visualization that is tagged with an **Admin** tag cannot be deleted.
- Each custom visualization is tagged with a **Catalog** or **Space** tag. Any visualization that is at the Space level but inherited from the Catalog level cannot be deleted at the Space level. It must be deleted at the Catalog level.

3. Confirm the deletion.

Note:

- If a default visualization was selected for deletion, a message is displayed in the notification banner to indicate that default visualizations cannot be deleted.
- If you delete a visualization that is currently inserted in one or more dashboards, the visualization container is retained in those dashboards and displays the following message: `Could not locate that visualization (id: visualization_ID)`. You will have to remove the visualization container from these dashboards.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

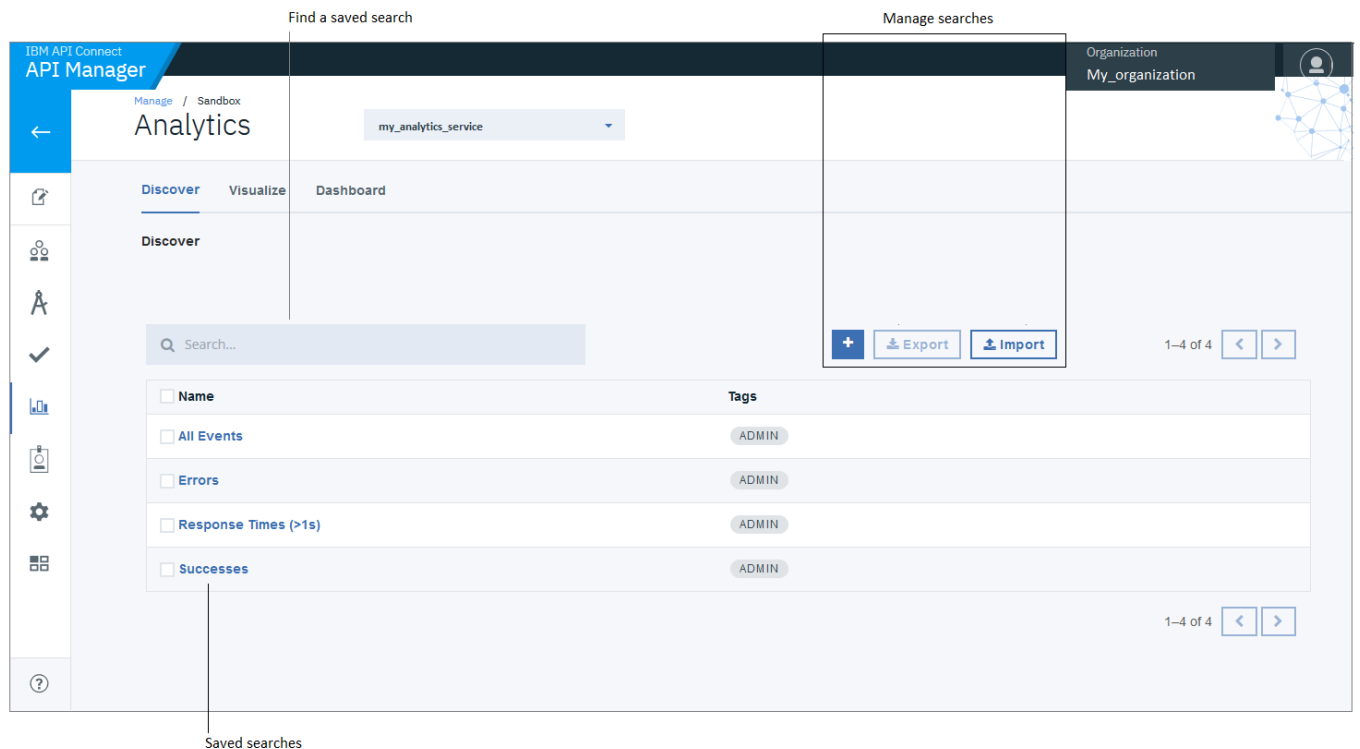
## Working with searches (Discover)


Use the IBM® API Connect Analytics Discover tab to create and run searches against your analytics data.

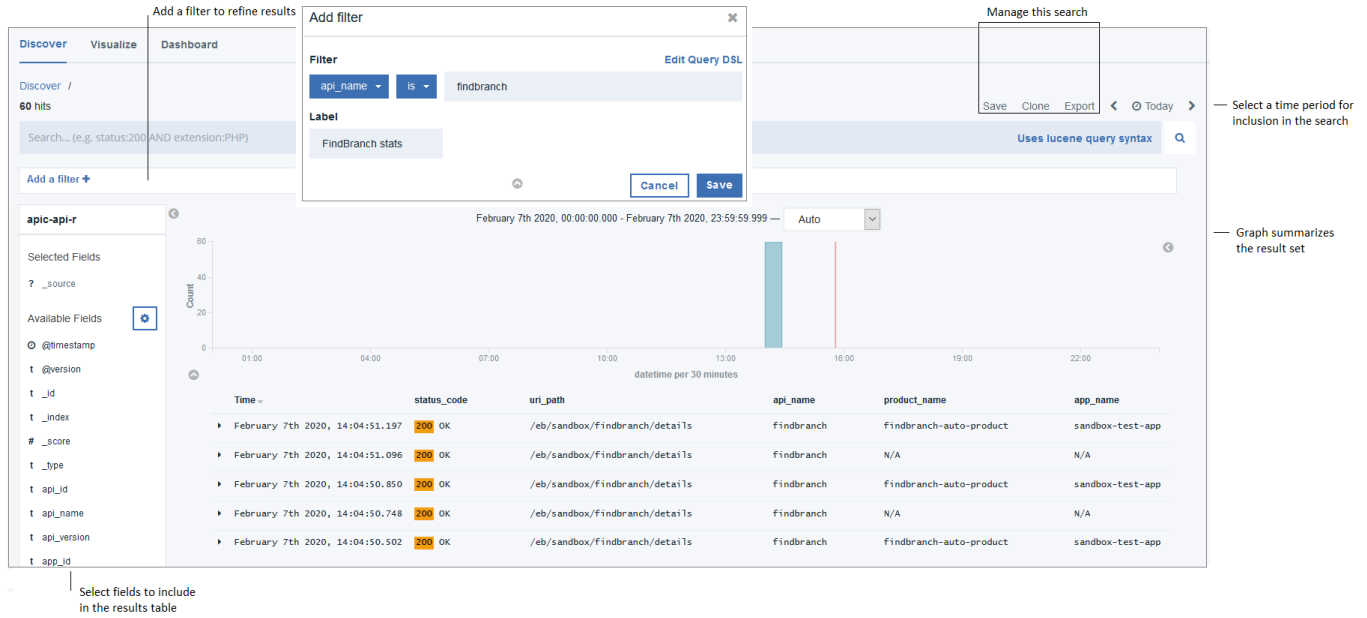
Each search is based on a query that is written using Lucene query syntax, which is described in the [Elasticsearch documentation](#).

Saved searches display a title and a tag that indicates the level of permissions needed to modify the query. For example, if a saved search is tagged with **catalog** then you must have Analytics Manage permissions at the Catalog scope to modify it; otherwise you must clone it and customize your own copy. Searches tagged with **admin** are predefined and cannot be modified but they can be cloned and then customized.

The following image shows the Discover page with the predefined searches.



Any user with Analytics View permissions can run a saved search and view the results, as well as export a saved search to JSON format by selecting the search and clicking Export. Users with Analytics Manage permissions can additionally create new searches by clicking  and defining a query, or by clicking Import and importing a JSON file that contains a query definition. The following image shows the results of the predefined "Successes" search.



The screenshot displays the 'Discover' tab in the IBM API Connect Analytics interface. It shows a search for 'findbranch' with 60 hits. An 'Add filter' dialog is open, showing a filter for 'api\_name' with the value 'findbranch'. Below the dialog, a bar chart shows a single bar at 16:00. A table below the chart lists search results with columns for Time, status\_code, uri\_path, api\_name, product\_name, and app\_name. The table contains five rows of data, all with a status\_code of '200 OK' and a uri\_path of '/eb/sandbox/Findbranch/details'. On the left, a sidebar shows 'Selected Fields' and 'Available Fields'. On the right, a 'Manage this search' dialog is open, showing 'Save', 'Clone', and 'Export' options. Annotations with arrows point to the 'Add filter' dialog, the 'Manage this search' dialog, the bar chart, and the table.

- **[Creating searches](#)**  
Use the IBM API Connect Analytics Discover tab to create new searches by defining a search query, or to import a search query that is stored in a JSON format file. You can then use saved searches to create visualizations.
- **[Editing searches](#)**  
Use the IBM API Connect Analytics Discover tab to modify searches.
- **[Exporting searches](#)**  
You can export searches for backing up the definitions, or so they can be imported into other Catalogs on your IBM API Connect system. If Spaces are enabled in a Catalog, exported searches can also be imported into a Space.
- **[Importing searches](#)**  
You can import searches from other IBM API Connect users for use, or you can import searches from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import searches from a Space.
- **[Backing up and restoring searches](#)**  
To ensure you can recreate your IBM API Connect searches in the event of data loss, you should export them and save the files as a back-up. Then you can import the search definitions later if needed.
- **[Deleting searches](#)**  
Use the IBM API Connect Analytics Discover tab to delete searches.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating searches

Use the IBM® API Connect Analytics Discover tab to create new searches by defining a search query, or to import a search query that is stored in a JSON format file. You can then use saved searches to create visualizations.


## Before you begin

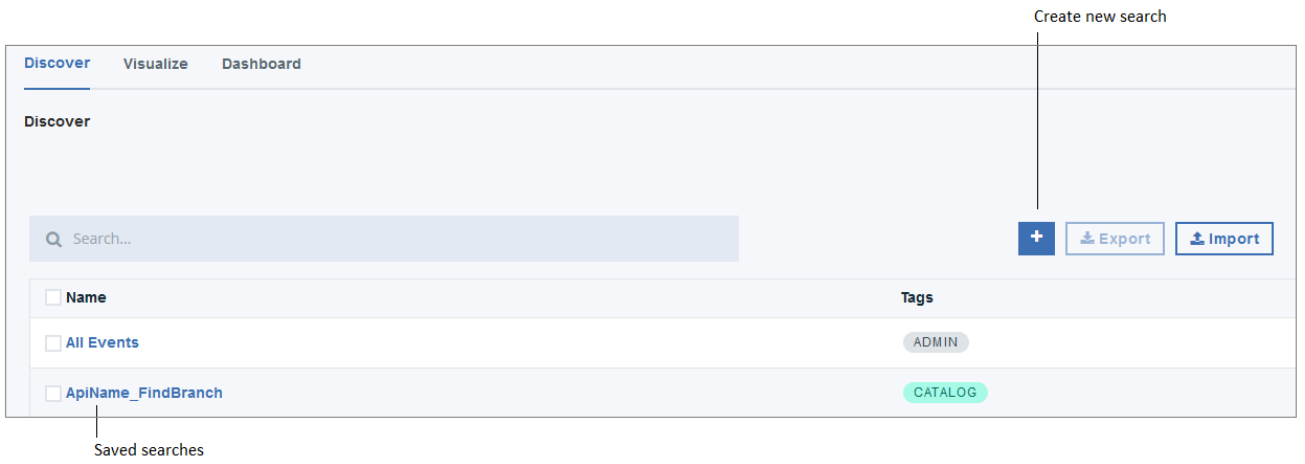
To create searches, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics - Manage permission for the selected Catalog or Space.

## About this task

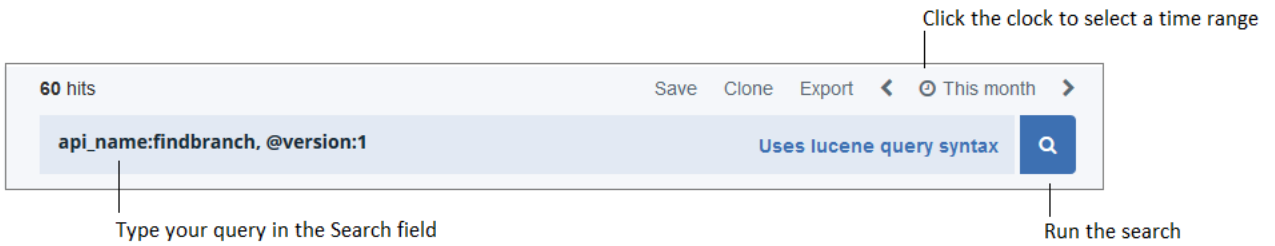
When you create a search at the Catalog level, the search is automatically added to all of the Spaces in that Catalog. When you create a search in a Space, the search is added only to the current Space.


## Procedure

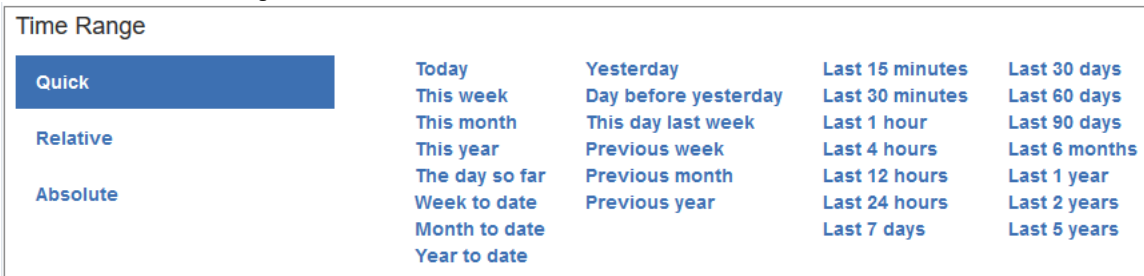
1. [Navigate to the Analytics feature for the Catalog or Space where you want to create a search.](#)
2. Click Discover to open the Discover tab, where you can create and manage searches.
3. Click .



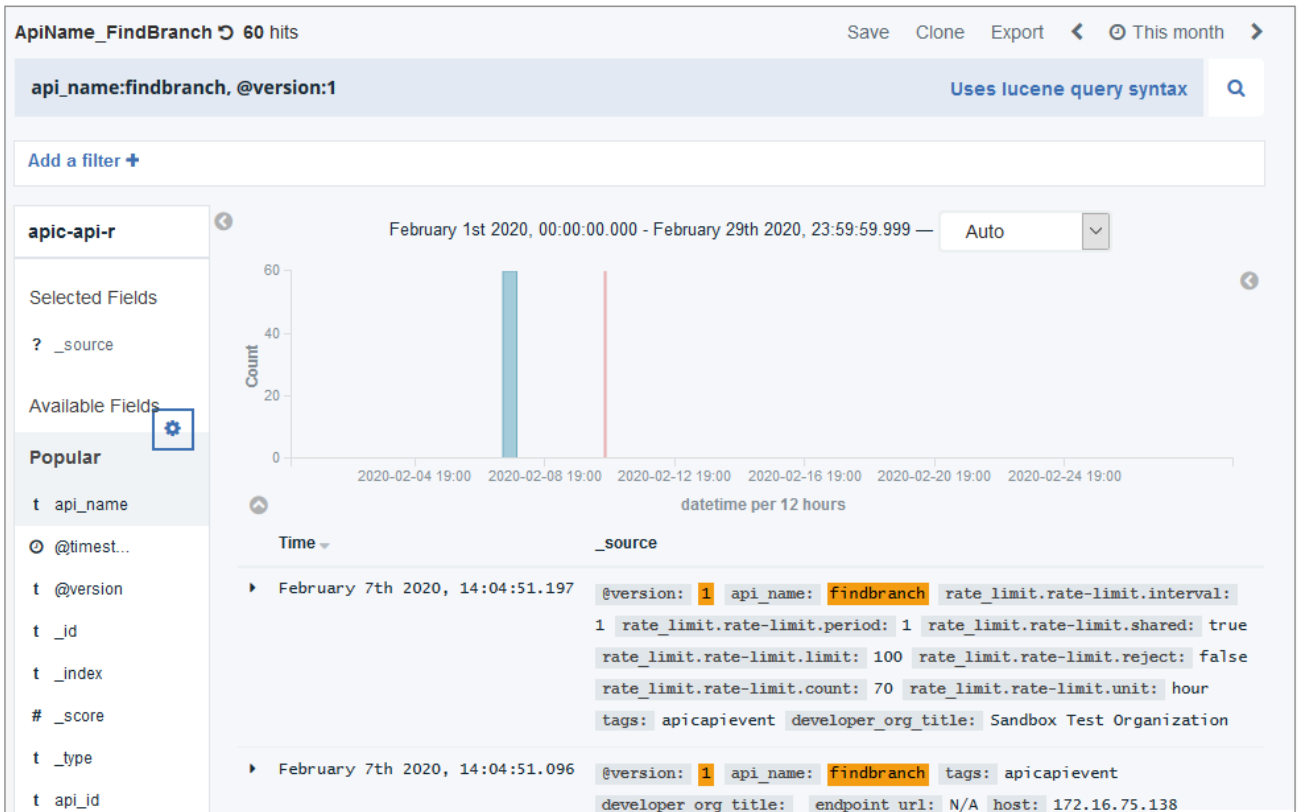
4. Use the Search field to define a query using [Lucene query syntax](#) as explained in the Elasticsearch documentation.



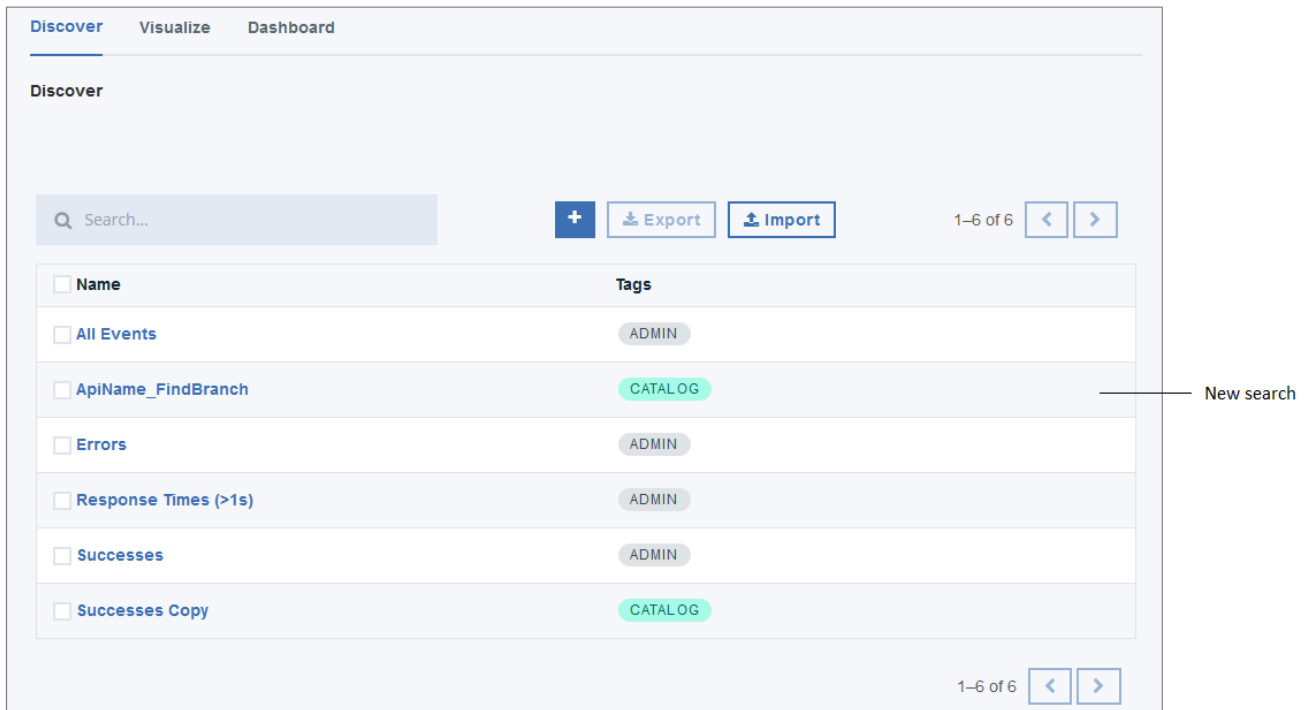
5. Click  and select a time range for the API events to be included in the search results.



6. Click  to run the search.



7. Save the search by clicking Save, providing a name for the search (the name defaults to "New Saved Search"), and clicking **Save**. The saved search is added to the displayed list of saved searches.



## Results

When your searches are saved, you can use them on the Visualize tab to display search results in graphical formats such as charts and maps.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## Editing searches

Use the IBM® API Connect Analytics Discover tab to modify searches.

### Before you begin

---

To edit searches, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics - Manage permission for the selected Catalog or Space.

### About this task

---

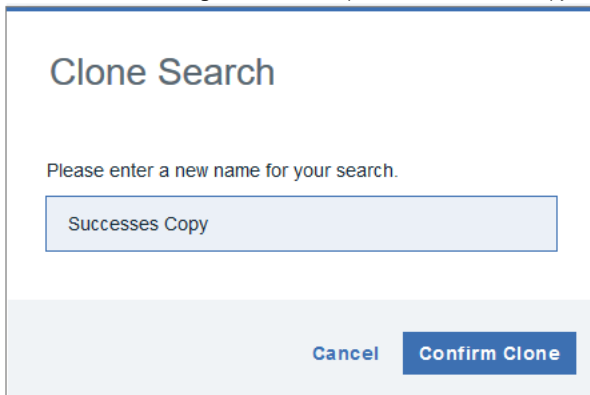
When you modify a search, you can run it immediately and see the new result set on the Discover tab. The new results might also appear in the visualizations and dashboards that use the search. Whether the visualizations and dashboards based on that search are updated to reflect the new result set depends on their scope and inheritance.

For example, if you select a Space and modify the search there, then the change only applies to the version of the search that runs within that Space. Versions of the search stored in other Spaces are not updated. If you modify the search at the Catalog scope, then the changes are applied across the entire Catalog and all Spaces within in it.

### Procedure

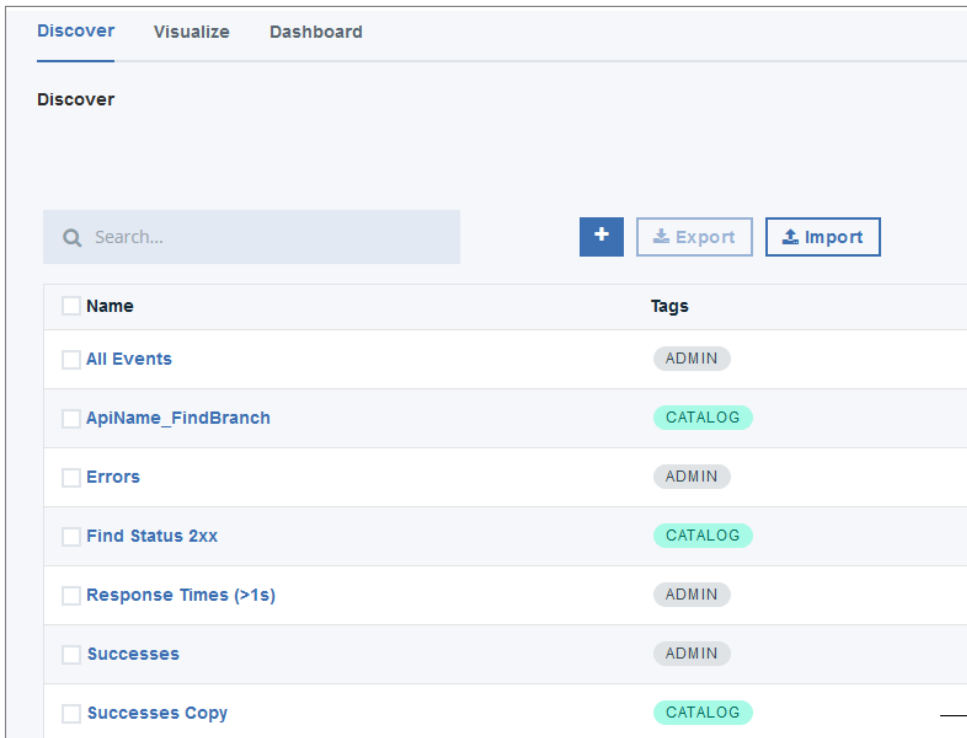
---

1. [Navigate to the Analytics feature for the Catalog or Space where you want to modify a search.](#)
2. Click Discover to open the Discover tab, where you can create and manage searches.
3. If you want to modify a search that you don't have permission to edit, clone it first by completing the following steps:  
In addition to belonging to a role with Analytics - Manage permission, you must have access to the scope in which the search was created if you want to edit that search directly. If you do not have the correct permissions to edit the search, you can clone it and then edit the copy to customize it. The tag that displays next to the search name indicates the scope of permissions required to edit the search. A predefined search (displays the **Admin** tag) cannot be modified and must always be cloned. A search tagged with **Space** can only be modified by users who have access to the Space in which the search was created. A search tagged with **Catalog** can be modified by all users of the current Catalog.
  - a. Click the name of the search that you want to clone.
  - b. Click Clone.
  - c. In the Clone Search dialog box, click Clone, provide a name for the copy (the name defaults to "Search-name Copy"), and then click Confirm Clone.



The image shows a 'Clone Search' dialog box. At the top, it says 'Clone Search'. Below that, it asks 'Please enter a new name for your search.' There is a text input field containing 'Successes Copy'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Confirm Clone'.

The new, editable copy of the search immediately appears in the list of saved searches.



4. In the list of searches, click the name of the search that you want to modify.
5. Edit the search query and the time frame for the search as needed.
6. Click Save to update the saved search or optionally store a new copy of the search.

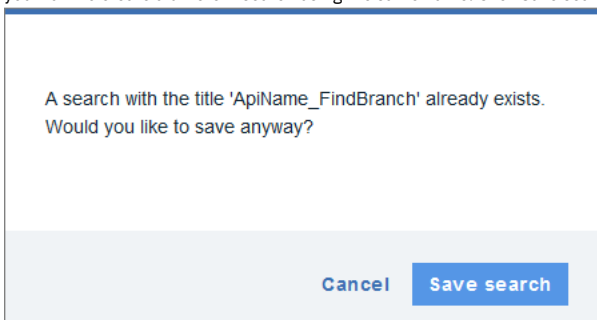
A dialog box displays where you can optionally rename the search or save it as a new search. Complete the fields and click **Save**.

Name of search being edited                      Name of new copy of search

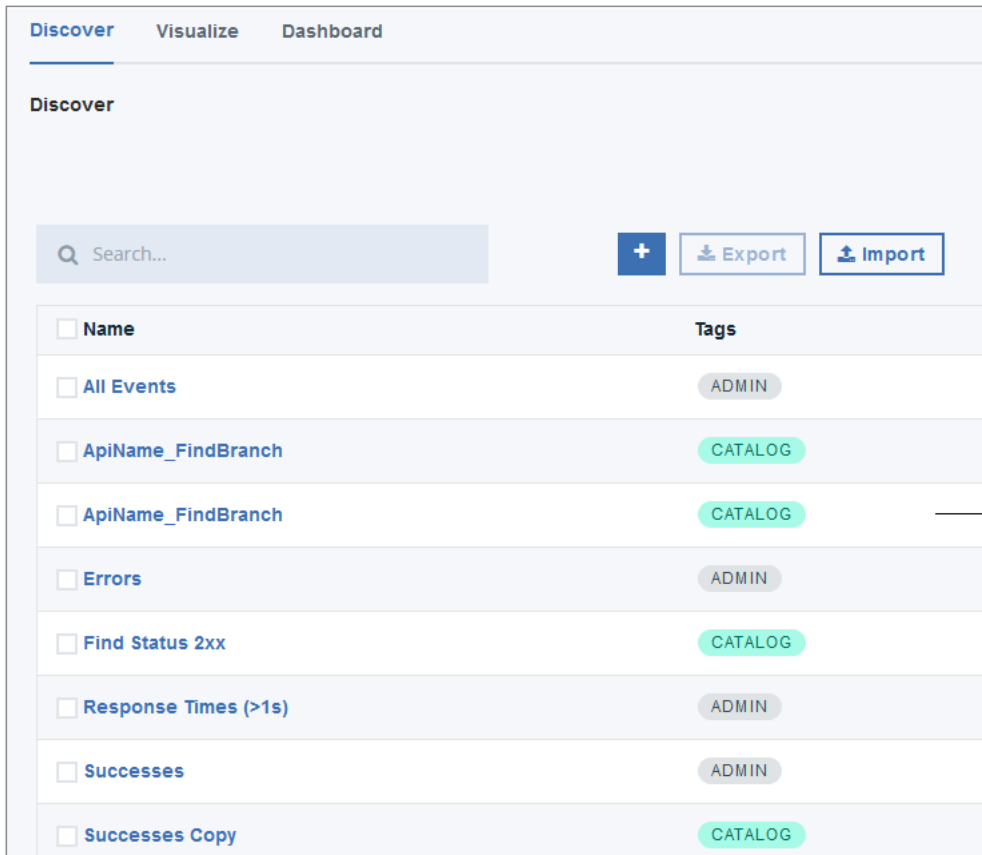


Save copy as a new search

Note: If you provide a new name for the search, it is saved with that name and the search that you originally selected to modify is left unchanged. If you select Save as a new search then a new search is created with the specified name. If you save a new search but do not change the name, a message displays to confirm that you want to create a different search using the same name. Click Save search to confirm that you want to save it as a new search.



If you save the search with an duplicate name, the search list displays both versions of the search with the older search appearing first.



Newest copy of search

You can edit searches and save them with new names to remove duplication.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Exporting searches

You can export searches for backing up the definitions, or so they can be imported into other Catalogs on your IBM® API Connect system. If Spaces are enabled in a Catalog, exported searches can also be imported into a Space.

## Before you begin

To export searches, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics - Manager permission for the selected Catalog or Space.

## About this task

Searches are exported as .json files, which can then be [imported](#).

## Procedure

1. [Navigate to the Analytics feature for the Catalog or Space where you want to create a search.](#)
2. Click Discover to open the Discover tab, where you can create and manage searches.
3. Select the searches that you want to export into a single JSON file.

Discover Visualize Dashboard

Discover

Search...

Export Import

Name	Tags
<input checked="" type="checkbox"/> All Events	ADMIN
<input checked="" type="checkbox"/> ApiName_FindBranch	CATALOG
<input type="checkbox"/> ApiName_FindBranch	CATALOG
<input type="checkbox"/> Errors	ADMIN
<input checked="" type="checkbox"/> Find Status 2xx	CATALOG
<input type="checkbox"/> Response Times (>1s)	ADMIN

Select all

Export

- Click
- Choose whether to open the file now or save it as export.json to your local "downloads" location.

## Results

The selected searches are exported to a single file called export.json.

Tip: If you plan to export multiple files, rename them after each download to avoid confusion.

```

1  [
2  {
3    "_type": "search",
4    "_source": {
5      "title": "All Events",
6      "description": "",
7      "hits": 0,
8      "columns": [],
9      "sort": [
10     "datetime",
11     "desc"
12   ],
13   "version": 1,
14   "kibanaSavedObjectMeta": {
15     "searchSourceJSON":
16     "{\"index\":\"apic-api-r\",\"highlightAll\":true,\"version\":true,\"filter\":[]}"
17   },
18   "default": true
19 }
20 ]

```

You can import the searches in this file to another Space or Catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Importing searches

You can import searches from other IBM® API Connect users for use, or you can import searches from another Catalog on your system. If Spaces are enabled in a Catalog, you can also import searches from a Space.

### Before you begin

To import searches, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics - Manage permission for the selected Catalog or Space.


### About this task

Searches can be imported from .json files that contain one or more exported search definitions. When you import a search at the Catalog level, the search is automatically added to all of the Spaces in that Catalog. When you import a search into a Space, the search is added only to the current Space.

Your search query must be contained in a JSON format file. If the file contains multiple search definitions, they will all be imported. For each search definition, the value in the `title` field is used as the new search's name.

```
1 [
2   {
3     "_type": "search",
4     "_source": {
5       "title": "Find Status 2xx",
6       "description": "",
7       "hits": 0,
8       "columns": [
9         "status_code",
10        "uri_path",
11        "api_name",
12        "product_name",
13        "app_name"
14      ],
15      "sort": [
16        "datetime",
17        "desc"
18      ],
19      "version": 1,
20      "kibanaSavedObjectMeta": {
21        "searchSourceJSON":
22        "{\"index\":\"apic-api-r\",\"highlightAll\":true,\"version\":true,\"filter\":[],\"query\":{\"query_string\":{\"query\":\"status_code:2*\",\"analyze_wildcard\":true}}}"
23      },
24      "catalog_id": "dd010c40-8bd6-460d-a12ff-fbcd3635f13f",
25      "org_id": "3f535b9b-baef-39ab-8ffd-495839b32408"
26    }
27  ]
```

### Procedure

1. [Navigate to the Analytics feature for the Catalog or Space where you want to create a search.](#)
2. Click Discover to open the Discover tab, where you can create and manage searches.
3. Click  Import.
4. Select the JSON file containing the search query and click Open.

### Results

The imported search is saved automatically and added to the displayed list of saved searches.

The screenshot shows the 'Discover' tab in IBM API Connect. At the top, there are navigation tabs for 'Discover', 'Visualize', and 'Dashboard'. Below the navigation, there is a search bar with the text 'Search...'. To the right of the search bar are buttons for '+', 'Export', and 'Import'. Further right, it shows '1-7 of 7' with left and right arrow buttons. Below this is a table of searches:

Name	Tags
<input type="checkbox"/> All Events	ADMIN
<input type="checkbox"/> ApiName_FindBranch	CATALOG
<input type="checkbox"/> Errors	ADMIN
<input type="checkbox"/> Find Status 2xx	CATALOG
<input type="checkbox"/> Response Times (>1s)	ADMIN
<input type="checkbox"/> Successes	ADMIN

An arrow points from the text 'Imported search' to the 'Find Status 2xx' search entry in the table.

If the JSON file contains multiple search definitions, all of the searches in the file are imported added to the search list. If an imported search's title duplicates an existing search title, all of the searches are retained and display in the search list with duplicate names. You can edit searches and save them with new names to remove duplication.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Backing up and restoring searches

To ensure you can recreate your IBM® API Connect searches in the event of data loss, you should export them and save the files as a back-up. Then you can import the search definitions later if needed.

### About this task

API Connect does not back up searches, so it is good practice to create your own backups by exporting all of your saved searches.

This task only operates on the search definition, not on the analytics data that is returned in the results.

### Procedure

1. Export your search definitions to JSON format as explained in the topic, [Exporting searches](#).
2. If you need to restore the searches, import the definitions from the JSON files as explained in the topic, [Importing searches](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting searches

Use the IBM® API Connect Analytics Discover tab to delete searches.

### Before you begin

To delete searches, you must either be the owner of the API provider organization, or be assigned a role that has the Analytics - Manage permission for the selected Catalog or Space.

Searches are used in visualizations. When you delete a search, it is removed from all visualizations where it was previously used. A user who views a visualization based on a search that was deleted sees the following error message: `Could not locate that search (id: search_ID)`. Before deleting a search, you should remove it from visualizations to ensure that users don't encounter the error.

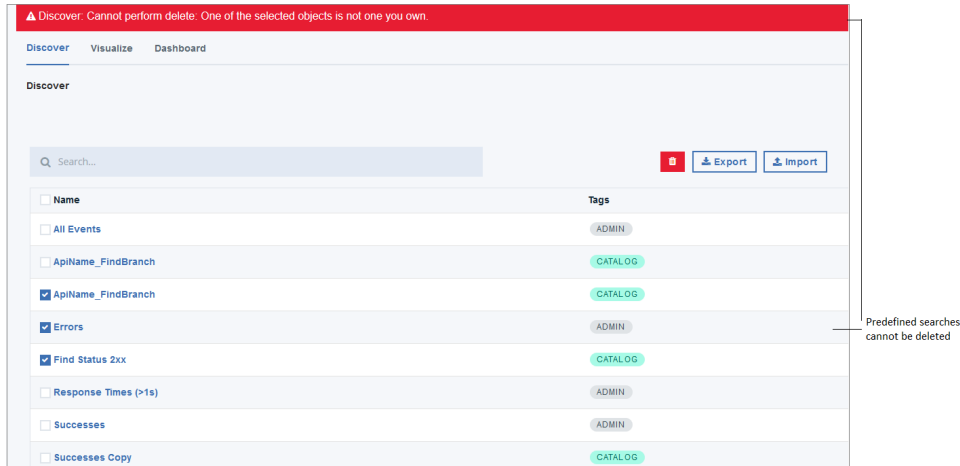
## About this task

In addition to belonging to a role with Analytics - Manage permission, you must have access to the scope in which the search was created if you want to delete that search. The tag that displays next to the search name indicates the scope of permissions required to edit the search. A predefined search (displays the **Admin** tag) cannot be deleted. A search tagged with **Catalog** can be deleted by all users of the current Catalog. A search tagged with **Space** can only be deleted by users who have access to the Space in which the search was created. Deleting a search from a Space removes it only from the current space. Other versions of the search stored in different Spaces are not deleted.


## Procedure

1. [Navigate to the Analytics feature for the Catalog or Space where you want to modify a search.](#)
2. Click Discover to open the Discover tab, where you can create and manage searches.
3. Select the searches that you want to delete.

Tip: If you have a large set of saved searches, you can either browse through the list or use the filter to locate the searches.



4. Click .

5. If you really want to delete the selected searches, click  to confirm.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Exporting API event data

You can obtain analytics and API event data from the API Manager user interface of IBM® API Connect or by using REST API calls.

## About this task

From the API Manager UI, you can view and export the raw analytics data that is returned in your configured visualizations. You can also view the individual API event records that are generated for the aggregated data sets in your visualizations, and you can collectively export all the API event records that relate to all visualizations in a dashboard. Any analytics or API event data that you export is saved to a comma-separated values (CSV) file.

You can alternatively obtain analytics data for a specific Catalog and API provider organization by using REST API calls to return JSON objects that contain the analytics data.

- [Using the UI to export API event data](#)  
Use the IBM API Connect Analytics interface to export API events data to CSV or JSON Lines format.
- [Using REST APIs to export events data to JSON or YAML](#)  
Use the IBM API Connect REST APIs to export API events data to JSON or YAML format.
- [API event record fields](#)

An API event is logged each time an API operation is invoked, and an event record is generated for each API event in the Gateway server. The API event record contains information about the API call and the content of the record depends on the logging policy that is set for the operation.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using the UI to export API event data

Use the IBM® API Connect Analytics interface to export API events data to CSV or JSON Lines format.

### About this task


---

Use the Analytics Discover tab to search for event data and then export the *hits* (the records in the search results) to a file. Each hit represents one API invocation in the Gateway server. When you view the hits in the UI, the display is limited to 500 records. To view the entire set of hits, export the data to a CSV or JSON Lines file.

### Procedure

---

1. Navigate to the Analytics page and click Discover.
2. Run a search.

Select a predefined search to run, or click  to define your own search, and then run it. Optionally apply filters to refine the results. The export option is only available when a search returns at least one hit.

3. Click Export (this option displays before the search results).
4. In the "Export Hits" panel, select the Filetype for the exported data.  
There are two options for the file's type:

- a. JSON Lines (.txt): Exports data to a .txt file that contains a complete JSON record for each hit.
- b. Comma-separated Values (.csv): Exports to a .csv file that can be viewed either in a text editor, a spreadsheet, or some other CSV viewer.  
If you intend to view the .csv file in a viewer that does not support special characters, select the following options as needed to ensure compatibility with your viewer:
  - i. Escape double quotes: Insert an additional quotation mark (") as an escape character before each " that is already included in the file.
  - ii. Escape special characters: Insert a quotation mark (") as an escape character before each of the following characters: = \_ + % @The content of the API event records in the CSV file depends on the logging policy that is set for the operation. If the logging policy includes the HTTP headers and the payload, then these details are included in the API event record. Where the payload is large, it might take longer to complete the export of the analytics data. For more information, see [activity-log.policy](#). For information about the fields that are included in the CSV file, see [API event records](#).

5. In the panel, click Export to start the export operation.
6. When the "The export has completed successfully." message displays, click Download to save the file, or click Another one? to run a new export operation. (Remember to save the current export results before running a new export).

Attention:

- If your browser is configured to block pop-ups, the download might be blocked. The precise behavior varies across browsers, but if you see a notification about "blocked pop-ups", reconfigure the browser to always allow pop-ups for the API Manager host address. Then check the "Export Hits" panel for the Download option, which completes the operation. If you do not see a Download option, repeat the export operation.
  - If you navigate away from the Analytics interface before the exported file is downloaded, the download operation is suspended. If you return to Analytics before any other export operation begins and before the file expires, you might see a Download option that allows you to finish downloading the file. Otherwise, you must repeat the entire export operation.
  - Each downloaded file is saved to the download location that is configured for your browser, using a randomly generated file name. If you intend to export more than once, and you are exporting to the same location each time, you might want to rename each file with a meaningful name.
- [Exporting aggregated analytics data from a visualization](#)  
From a visualization, you can view the raw aggregated data from which the graphical representation is constructed, and then export that data to a CSV file.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Exporting aggregated analytics data from a visualization

From a visualization, you can view the raw aggregated data from which the graphical representation is constructed, and then export that data to a CSV file.

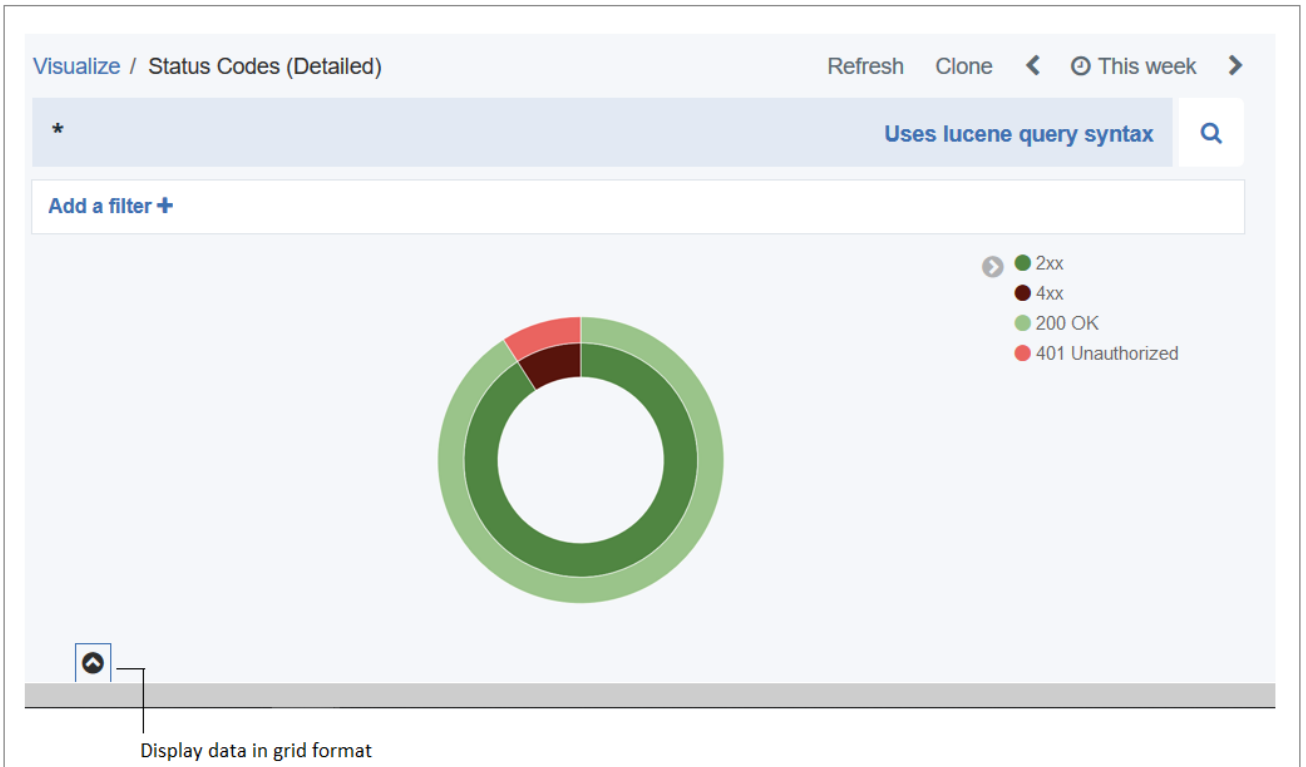
### Procedure


---

To export analytics data from a visualization, complete the following steps.

1. Navigate to the Analytics page and click Visualize.
2. Select a visualization by clicking it.  
For example, you might select the visualization called "Status Codes (Detailed)" as shown in the following image:





3. Display the data in grid format by clicking the  that displays after the visualization. In the resulting grid, a row is displayed for each aggregated data set that matches the search criteria configured in the visualization's aggregation builder. The raw data shown in the columns and rows depends on the type of visualization accessed and the aggregations configured for that visualization.

Visualize / Status Codes (Detailed) Refresh Clone < This week >

\* Uses lucene query syntax

Add a filter +

Table

filters	Count	status_code.keyword: Descending	Count
1xx	0		
2xx	20	200 OK	20
3xx	0		
4xx	2	401 Unauthorized	2
5xx	0		

Export: Raw Formatted

Page Size 10

Display in graph format Export options

4. Export a tabular format of the raw data that is displayed in the grid:
- To export the data as stored in Elasticsearch, click the Raw link.
  - To export Kibana formatted data that is similar to what you see in the grid, click the Formatted link.
5. When prompted, choose to save the file, which is named *visualization\_name.csv* by default. The file is saved to the download location that is configured for your browser. If you intend to export more than once from the same or another visualization, and you are exporting to the same location each time, you might want to rename each file with a meaningful name.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Using REST APIs to export events data to JSON or YAML

Use the IBM® API Connect REST APIs to export API events data to JSON or YAML format.

The Analytics REST APIs let you create analytics objects and retrieve data. When you reference an API, use the API's own endpoint on your management server.

For details on the APIs, browse to the **API Connect Analytics APIs 1.0.0** section of the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## API event record fields

An API event is logged each time an API operation is invoked, and an event record is generated for each API event in the Gateway server. The API event record contains information about the API call and the content of the record depends on the logging policy that is set for the operation.

You can use the activity-log policy to configure your logging preferences for the API event details that are stored in the Analytics component. By default, invocation details are logged if an API call is successful, and invocation, header, and payload (message body) details are logged if an API call results in an error code. To override these default settings and change the level of detail that is included in the API event record, you can add the activity-log policy to your API assembly and then configure the policy's properties. For example:

- To include details about the request body or response body in the API event record for a successful API call, you can add an activity-log policy to the associated API operation and set the content type to **payload**.  
Restriction: The payload logging feature is disabled for IBM® Cloud instances that are hosted in the Frankfurt region. The requirements for storing Sensitive Personal Information (SPI) are more restrictive in that region, so the payload information cannot be saved.
- To include details about the HTTP request headers or HTTP response headers in the API event record for a successful API call, you can add an activity-log policy to the associated API operation and set the content type to either **header** or **payload**.

For more information about how to configure your logging preferences, see [activity-log policy](#) and [Including components in your assembly](#).

Tip: A log policy field is included in the event record to identify the logging setting. To see examples of the invocation, header, and payload details that can be included in an API event record, see:

- [Example: Event record with invocation details \(activity logging setting\)](#)
- [Example: Event record with invocation and header details \(header logging setting\)](#)
- [Example: Event record with invocation, header, and payload details \(payload logging setting\)](#)

The following table lists the static set of fields that are displayed in an API event record. When creating visualizations, you can use these fields to configure aggregations that define the type and level of information to be retrieved and displayed. For more information, see [Configuring visualizations](#). If you have configured your logging preferences to include header and payload details, your header and payload fields will additionally be available for selection while configuring aggregations, enabling you to create visualizations based on header and payload data if required.

Note: If you want to include the `client_geoup` and `gateway_geoup` fields in your Analytics data, ask your administrator to configure it as explained in [Configuring output plugins for analytics offload](#).

Table 1. API event record fields

Field name	Type	Description
<code>datetime</code>	Date	A time stamp that records when the record was written by the <a href="#">Logstash</a> data collection engine that feeds data into Elasticsearch.
<code>api_id</code>	String	The API identifier.
<code>api_name</code>	String	The name of the API.
<code>api_version</code>	String	The version number of the API.
<code>app_id</code>	String	The identifier for the registered application.
<code>app_name</code>	String	The name of the registered application.
<code>app_type</code>	String	The application type, with a value of <b>Production</b> or <b>Development</b> .
<code>bytes_received</code>	Number	The number of bytes received on the inbound request.
<code>bytes_sent</code>	Number	The number of bytes sent on the outbound request.
<code>client_geoup.area_code</code>	Number	The public switched telephone network (PSTN) area code of the client, as identified from its IP address.
<code>client_geoup.city_name</code>	String	The city name of the client, as identified from its IP address.
<code>client_geoup.continent_code</code>	String	The two-letter continent code of the client, as identified from its IP address.
<code>client_geoup.country_code2</code>	String	The two-letter country code of the client, as identified from its IP address.
<code>client_geoup.country_code3</code>	String	The three-letter country code of the client, as identified from its IP address.

Field name	Type	Description
client_geop.country_name	String	The country name of the client, as identified from its IP address.
client_geop.dma_code	Number	The Designated Market Area (DMA) code of the client, as identified from its IP address.
client_geop.ip	String	The IP address of the client.
client_geop.latitude	Number	The latitude of the client location, as identified from its IP address.
client_geop.location	String	The longitude and latitude of the client location (separated by a comma), as identified from its IP address.
client_geop.longitude	Number	The longitude of the client location, as identified from its IP address.
client_geop.postal_code	String	The postal code of the client, as identified from its IP address.
client_geop.real_region_name	String	The full name of the region that corresponds to the IP address of the client.
client_geop.region_name	String	The abbreviated form of the region that corresponds to the IP address of the client.
client_geop.timezone	String	The time zone of the client, as identified from its IP address.
client_id	String	The unique ID of the client that is attached to the API request.
client_ip	String	The original client IP address, as obtained from the <b>X-Forwarded-For</b> header.
datetime	Date	A time stamp that records when the API was executed. The time stamp is always shown in coordinated universal time UTC.
developer_org_id	String	The identifier for the consumer organization that owns the application.
endpoint_url	String	When the request failed, identifies the proxy or invoke target URL on which the request failed. It is not included with a successful request.
developer_org_name	String	The name of the consumer organization that owns the application.
env_id	String	The Catalog identifier.
env_name	String	The name of the Catalog.
gateway_geop.area_code	Number	The public switched telephone network (PSTN) area code of the gateway, as identified from its IP address.
gateway_geop.city_name	String	The city name of the gateway, as identified from its IP address.
gateway_geop.continent_code	String	The two-letter continent code of the gateway, as identified from its IP address.
gateway_geop.country_code2	String	The two-letter country code of the gateway, as identified from its IP address.
gateway_geop.country_code3	String	The three-letter country code of the gateway, as identified from its IP address.
gateway_geop.country_name	String	The country name of the gateway, as identified from its IP address.
gateway_geop.dma_code	Number	The Designated Market Area (DMA) code of the gateway, as identified from its IP address.
gateway_geop.ip	String	The IP address of the gateway.
gateway_geop.latitude	Number	The latitude of the gateway location, as identified from its IP address.
gateway_geop.location	String	The longitude and latitude of the gateway location (separated by a comma), as identified from its IP address.
gateway_geop.longitude	Number	The longitude of the gateway location, as identified from its IP address.
gateway_geop.postal_code	String	The postal code of the gateway, as identified from its IP address.
gateway_geop.real_region_name	String	The full name of the region that corresponds to the IP address of the gateway.
gateway_geop.region_name	String	The abbreviated form of the region that corresponds to the IP address of the gateway.
gateway_geop.timezone	String	The time zone of the gateway, as identified from its IP address.
gateway_ip	String	The IP address of the gateway.
headers.field_name	String	A component of the header section of a message. Note: The following types of headers are considered sensitive and will not show in analytics data for security reasons: <ul style="list-style-type: none"> <li>Any secret key configured in the API security</li> <li>Any header that contains <b>secret</b></li> <li>Any header that contains <b>Authorization</b></li> </ul>
host	String	The host name or IP address.
http_user_agent	String	The value of the User Agent header on the inbound request.
immediate_client_ip	String	The client IP address that is directly in front of the gateway. In most cases this is a load balancer.

Field name	Type	Description
latency_info.started	Number	The time delay (in milliseconds) between when the request was received and when the corresponding task was started by the gateway. Starting a task comprises multiple steps to prepare for executing an API; for example, completing the TCP/TLS handshake, verifying an app's client ID and secret, and matching the request URI to a Catalog, API, and Plan. When the gateway receives a request, the "Start" duration is set to 0. The duration of each step within the Start task is then added up, and the total represents the duration of the Start task.
latency_info.task	String	The API transaction that was processed.
log_policy	String	The defined logging policy. Values include none, event, headers, and payload.
org_id	String	The identifier for the provider organization that owns the API and associated Products.
org_name	String	The name of the provider organization that owns the API and associated Products.
plan_id	String	The Plan identifier.
plan_name	String	The name of the Plan.
plan_version	String	The version number of the Plan.
product_name	String	The Product name.
product_title	String	The title of the Product.
product_version	String	The version number of the Product.
query_string	String	The URL query string value on the inbound request.
rate_limit.count	Number	The number of API calls made in the defined rate limit time window.
rate_limit.limit	Number	The maximum number of requests an application is allowed to make to the API during a specified time window.
rate_limit.period	Number	The time window that is used to set a rate limit for API calls.
rate_limit.reject	String	An indication of whether calls that exceed the specified rate limit will be rejected. If true, the API call will be rejected with a 429 status code. If false, a record is created in the Activity log.
rate_limit.shared	String	An indication of whether the rate limit is shared at a Plan level by all operations, or whether a rate limit is specified on individual operations.
request_body	String	The body of the inbound request.
request_http_headers.field_name	String	A component of the HTTP header section of the inbound request; for example, the acceptable encodings, the identification string for the user agent, or the proxies through which the request was sent.
request_method	String	The method of the inbound request.
request_protocol	String	The protocol of the inbound request.
resource	String	The name of the operation.
resource_id	String	The operation identifier.
resource_path	String	The operation path.
response_body	String	The body of the outbound response.
response_http_headers.field_name	String	A component of the HTTP header section of the outbound response; for example, the MIME type of the content or the data and time when the message was sent.
status_code	String	The status code set on the outbound response.
time_to_serve_request	Number	The time elapsed (in milliseconds) from when the request was received by the backend application to when it sent a response.
transaction_id	String	The identifier for the API transaction.
uri_path	String	The URI path on the inbound request.

## Example: Event record with invocation details (activity logging setting)

```
{
  "datetime": "2016-09-29T22:17:43.404Z",
  "latency_info": [
    {
      "task": "Start",
      "started": 2
    },
    {
      "task": "security-appID",
      "started": 7
    },
    {
      "task": "Plan Limit",
      "started": 11
    },
    {
      "task": "proxy",
      "started": 12
    }
  ],
  "api_version": "1.0.0",
  "product_version": "1.0.0",
  "product_name": "INTERNAL_QS",
  "plan_version": "1.0.0",
  "uri_path": "/macs-shack/sb/AccountService",
  "request_method": "POST",
  "log_policy": "activity",
  "request_protocol": "https",
  "query_string": [],
  "request_body": "",
  "response_body": "",
  "bytes_received": 256,
}
```

```

"bytes_sent": 256,
"time_to_serve_request": 301,
"status_code": "200 OK",
"request_http_headers": [],
"response_http_headers": [],
"org_name": "macs-shack",
"api_name": "accountservice",
"catalog_name": "sb",
"resource_path": "post",
"plan_name": "default",
"developer_org_name": "macs-shack",

"client_geoup": {
  "ip": "9.20.152.215",
  "country_code2": "US",
  "country_code3": "USA",
  "country_name": "United States",
  "continent_code": "NA",
  "region_name": "NC",
  "city_name": "Durham",
  "postal_code": "27709",
  "latitude": 35.994,
  "longitude": -78.8986,
  "dma_code": 560,
  "area_code": 919,
  "timezone": "America/New_York",
  "real_region_name": "North Carolina",
  "location": [
    -78.8986,
    35.994
  ]
},
"gateway_geoup": {
  "ip": "9.79.12.126",
  "country_code2": "US",
  "country_code3": "USA",
  "country_name": "United States",
  "continent_code": "NA",
  "region_name": "NC",
  "city_name": "Durham",
  "postal_code": "27709",
  "latitude": 35.994,
  "longitude": -78.8986,
  "dma_code": 560,
  "area_code": 919,
  "timezone": "America/New_York",
  "real_region_name": "North Carolina",
  "location": [
    -78.8986,
    35.994
  ]
}
}

```

## Example: Event record with invocation and header details (header logging setting)

```

{
  "datetime": "2016-09-29T22:53:46.766Z",
  "latency_info": [
    {
      "task": "Start",
      "started": 3
    },
    {
      "task": "security-appID",
      "started": 8
    },
    {
      "task": "Plan Limit",
      "started": 84
    },
    {
      "task": "activity-log",
      "started": 86
    },
    {
      "task": "proxy",
      "started": 88
    }
  ],
  "api_version": "1.0.0",
  "product_version": "1.0.0",
  "product_name": "__INTERNAL_QS__",
  "plan_version": "1.0.0",
  "uri_path": "/macs-shack/sb/AccountService",
  "request_method": "POST",
  "log_policy": "header",
  "request_protocol": "https",
  "query_string": [],
  "request_body": "",
  "response_body": "",
  "bytes_received": 256,
  "bytes_sent": 256,
  "time_to_serve_request": 317,

```

```

"status_code": "200 OK",
"request_http_headers": [
  {
    "Host": "apimanager.host.com"
  },
  {
    "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0"
  },
  {
    "Accept": "application/xml"
  },
  {
    "Accept-Language": "en-US,en;q=0.5"
  },
  {
    "Accept-Encoding": "gzip, deflate"
  },
  {
    "APIm-Debug": "true"
  },
  {
    "Content-Type": "text/xml"
  },
  {
    "SOAPAction": "getBalance"
  },
  {
    "Referer": "https://apimanager.host.com/apim/"
  },
  {
    "Content-Length": "256"
  },
  {
    "Origin": "https://apimanager.host.com"
  },
  {
    "Via": "1.1 AwAABaygGU-"
  },
  {
    "X-Client-IP": "9.79.12.126"
  },
  {
    "X-Global-Transaction-ID": "1364721"
  }
],
"response_http_headers": [
  {
    "Content-Type": "text/xml; charset=ISO-8859-1"
  },
  {
    "Date": "Thu, 29 Sep 2016 22:53:46 GMT"
  },
  {
    "X-Powered-By": "Servlet/3.0"
  },
  {
    "X-Vcap-Request-Id": "452d95be-0304-4f73-7429-7186ca6be843"
  },
  {
    "X-Global-Transaction-ID": "1364721"
  },
  {
    "Access-Control-Expose-Headers": "APIm-Debug-Trans-Id, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, X-Global-Transaction-ID"
  },
  {
    "Access-Control-Allow-Origin": "https://apimanager.host.com"
  },
  {
    "Access-Control-Allow-Methods": "POST"
  },
  {
    "Access-Control-Allow-Credentials": "true"
  }
],
"org_name": "macs-shack",
"api_name": "accounts-service",
"catalog_name": "sb",
"resource_path": "post",
"plan_name": "default",
"developer_org_name": "macs-shack",

"client_geoip": {
  "ip": "9.20.152.215",
  "country_code2": "US",
  "country_code3": "USA",
  "country_name": "United States",
  "continent_code": "NA",
  "region_name": "NC",
  "city_name": "Durham",
  "postal_code": "27709",
  "latitude": 35.994,
  "longitude": -78.8986,
  "dma_code": 560,
  "area_code": 919,
  "timezone": "America/New_York",

```

```

    "real_region_name": "North Carolina",
    "location": [
      -78.8986,
      35.994
    ]
  },
  "gateway_geoip": {
    "ip": "9.79.12.126",
    "country_code2": "US",
    "country_code3": "USA",
    "country_name": "United States",
    "continent_code": "NA",
    "region_name": "NC",
    "city_name": "Durham",
    "postal_code": "27709",
    "latitude": 35.994,
    "longitude": -78.8986,
    "dma_code": 560,
    "area_code": 919,
    "timezone": "America/New_York",
    "real_region_name": "North Carolina",
    "location": [
      -78.8986,
      35.994
    ]
  }
}

```

## Example: Event record with invocation, header, and payload details (payload logging setting)

```

{
  "datetime": "2016-09-29T22:26:28.667Z",
  "latency_info": [
    {
      "task": "Start",
      "started": 3
    },
    {
      "task": "security-appID",
      "started": 8
    },
    {
      "task": "Plan Limit",
      "started": 11
    },
    {
      "task": "activity-log",
      "started": 12
    },
    {
      "task": "proxy",
      "started": 269
    }
  ],
  "api_version": "1.0.0",
  "product_version": "1.0.0",
  "product_name": "INTERNAL_QS__",
  "plan_version": "1.0.0",
  "uri_path": "/macs-shack/sb/AccountService",
  "request_method": "POST",
  "log_policy": "payload",
  "request_protocol": "https",
  "query_string": [],
  "request_body": "<SOAP-ENV:Envelope xmlns:SOAP-ENV=\"http://schemas.xmlsoap.org/soap/envelope/\"><SOAP-ENV:Header/><SOAP-ENV:Body><ban:getBalance xmlns:ban=\"http://bankA.sample.ibm.com/\">\n  <arg0>3</arg0>\n</ban:getBalance></SOAP-ENV:Body></SOAP-ENV:Envelope>",
  "response_body": "<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"><soap:Body><ns2:getBalanceResponse xmlns:ns2=\"http://bankA.sample.ibm.com/\"><return>4</return></ns2:getBalanceResponse></soap:Body></soap:Envelope>",
  "bytes_received": 256,
  "bytes_sent": 256,
  "time_to_serve_request": 603,
  "status_code": "200 OK",
  "request_http_headers": [
    {
      "Host": "apimanager.host.com"
    },
    {
      "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0"
    },
    {
      "Accept": "application/xml"
    },
    {
      "Accept-Language": "en-US,en;q=0.5"
    },
    {
      "Accept-Encoding": "gzip, deflate"
    },
    {
      "APIM-Debug": "true"
    },
    {
      "Content-Type": "text/xml"
    }
  ],
}

```

```

{
  "SOAPAction": "getBalance"
},
{
  "Referer": "https://apimanager.host.com/apim/"
},
{
  "Content-Length": "256"
},
{
  "Origin": "https://apimanager.host.com"
},
{
  "Via": "1.1 AQAAPSLVfg-"
},
{
  "X-Client-IP": "9.79.12.126"
},
{
  "X-Global-Transaction-ID": "1204915"
}
],
"response_http_headers": [
  {
    "Content-Type": "text/xml; charset=ISO-8859-1"
  },
  {
    "Date": "Thu, 29 Sep 2016 22:26:28 GMT"
  },
  {
    "X-Powered-By": "Servlet/3.0"
  },
  {
    "X-Global-Transaction-ID": "1204915"
  },
  {
    "Access-Control-Expose-Headers": "APIM-Debug-Trans-Id, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, X-Global-Transaction-ID"
  },
  {
    "Access-Control-Allow-Origin": "https://apimanager.host.com"
  },
  {
    "Access-Control-Allow-Methods": "POST"
  },
  {
    "Access-Control-Allow-Credentials": "true"
  }
],
"org_name": "macs-shack",
"api_name": "accountservice",
"catalog_name": "sb",
"resource_path": "post",
"plan_name": "default",
"developer_org_name": "macs-shack",

"client_geoup": {
  "ip": "9.20.152.215",
  "country_code2": "US",
  "country_code3": "USA",
  "country_name": "United States",
  "continent_code": "NA",
  "region_name": "NC",
  "city_name": "Durham",
  "postal_code": "27709",
  "latitude": 35.994,
  "longitude": -78.8986,
  "dma_code": 560,
  "area_code": 919,
  "timezone": "America/New_York",
  "real_region_name": "North Carolina",
  "location": [
    -78.8986,
    35.994
  ]
},
"gateway_geoup": {
  "ip": "9.79.12.126",
  "country_code2": "US",
  "country_code3": "USA",
  "country_name": "United States",
  "continent_code": "NA",
  "region_name": "NC",
  "city_name": "Durham",
  "postal_code": "27709",
  "latitude": 35.994,
  "longitude": -78.8986,
  "dma_code": 560,
  "area_code": 919,
  "timezone": "America/New_York",
  "real_region_name": "North Carolina",
  "location": [
    -78.8986,
    35.994
  ]
}
]

```



```
}  
}
```

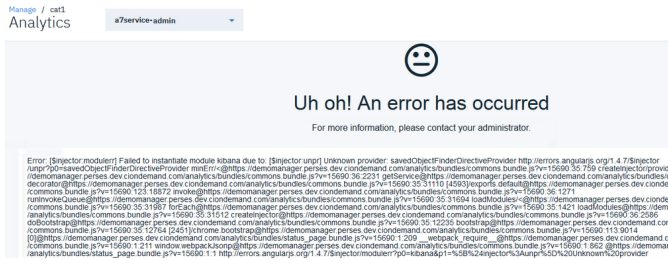
**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x](#) and [later product documentation](#).

## Troubleshooting Analytics start-up

If you receive an error starting the Analytics service after installing, upgrading, or recovering from a catastrophic failure of IBM® API Connect, you might be able to resolve it by completing this task.

### Uh oh! An error has occurred

In rare cases, the Kibana index can fail and the following message displays:



This error typically indicates that the index was corrupted. Complete the following steps to verify the problem and delete the corrupted index.

Before you begin, determine the names of one of the deployment's storage-coordinating pods (called *analytics-storage-coordinating-pod* in the steps that follow) and one of the client pods (called *analytics-client-pod* in the steps). To see the pod names, run the following command:

```
kubectl get pods
```

In the following sample output, a storage-coordination pod is named `r70eaa1a0f2-analytics-storage-coordinating-5d87d4c76-btpfq` and a client pod is named `r70eaa1a0f2-analytics-client-68c499d5f9-7g7vm`:

NAME	READY	STATUS	RESTARTS	AGE
r70eaa1a0f2-analytics-client-68c499d5f9-7g7vm	1/1	Running	0	37m
r70eaa1a0f2-analytics-client-68c499d5f9-hjx22	1/1	Running	0	82m
r70eaa1a0f2-analytics-cronjobs-retention-1559698200-7tjjh	0/1	Completed	0	18h
r70eaa1a0f2-analytics-cronjobs-rollover-1559762100-fnhnd	0/1	Completed	0	24m
r70eaa1a0f2-analytics-ingestion-7f5d748759-5cgrs	1/1	Running	0	83m
r70eaa1a0f2-analytics-ingestion-7f5d748759-wjvkd	1/1	Running	0	83m
r70eaa1a0f2-analytics-mtls-gw-6988bd8cbf-g88p5	1/1	Running	0	83m
r70eaa1a0f2-analytics-mtls-gw-6988bd8cbf-xxzvb	1/1	Running	0	83m
r70eaa1a0f2-analytics-operator-747f75c4f-7965r	1/1	Running	0	83m
r70eaa1a0f2-analytics-storage-coordinating-5d87d4c76-btpfq	1/1	Running	0	83m
r70eaa1a0f2-analytics-storage-coordinating-5d87d4c76-sr9mg	1/1	Running	0	83m
r70eaa1a0f2-analytics-storage-coordinating-5d87d4c76-t8djr	1/1	Running	0	82m
r70eaa1a0f2-analytics-storage-data-0	1/1	Running	0	70m
r70eaa1a0f2-analytics-storage-data-1	1/1	Running	0	76m
r70eaa1a0f2-analytics-storage-data-2	1/1	Running	0	80m
r70eaa1a0f2-analytics-storage-master-0	1/1	Running	0	72m
r70eaa1a0f2-analytics-storage-master-1	1/1	Running	0	75m
r70eaa1a0f2-analytics-storage-master-2	1/1	Running	0	80m

1. Verify that you have both the `.kibana` and `.kibana-6` indexes by running the following command:

```
kubectl exec -it analytics-storage-coordinating-pod -- curl_es /_cat/indices?v
```

In the `index` column of the response, look for `.kibana` and `.kibana-6`:

```
health status index      uid      pri rep docs.count docs.deleted store.size pri.store.size  
green open   .kibana-6      9AC2HDC3TLuKTRlvm0Bt0A  1  2      1      0      9.3kb      3.1kb  
green open   .kibana        WLMq7241ST6YRjI8p3Vp1Q  1  2      1      0      14.6kb     4.8kb  
green open   .apic-config   Zlh3BNn-T4GFWMHnBDG3w  1  2      0      0      573b      191b  
green open   .export-status 4uvi5QRVRlWQ-1sum1-Krg  5  2      148889  0      350.2mb   116.8mb  
green open   apic-api-2019.06.04-1 0WC8pIuqKqR6USJgRdGQ  5  2      8000    0      24.7mb    8.2mb
```

If the response includes both indexes, it indicates a problem with the index creation. Proceed to the next step.

Note: If you received the "Uh oh! An error has occurred" page and the response to this command only includes the `.kibana-6` index, then something else is wrong with the cluster and you should contact IBM Support for assistance.

2. Set the `.kibana` index to be read-and-write enabled by running the following command, replacing `analytics-storage-coordinating-pod` with the name of the storage-coordinating pod that you determined at the beginning of this task:

```
kubectl exec -it analytics-storage-coordinating-pod -- curl_es -XPUT .kibana/_settings -d '{"index.blocks.write":false}'
```

When you see the following response: `{"acknowledged":true}`, continue to the next step.

Note: If you see a different response, make sure the request was correct and try again. If the command still does not work, then something else is wrong with the cluster and you should contact IBM Support for assistance.

3. Delete the `.kibana-6` index by running the following command:

```
kubectl exec -it analytics-storage-coordinating-pod -- curl_es -XDELETE .kibana-6
```

When you see the following response: {"acknowledged":true} to indicate that the deletion was successful, continue to the next step.

Note: If you see a different response, then the index was not deleted. Make sure the request was correct and try again. If the delete operation still fails, then something else is wrong with the cluster and you should contact IBM Support for assistance.

- Restart a single analytics client pod by running the following command and replacing `analytics-client-pod` with the name of the pod (which you determined at the beginning of this task):

```
kubectl delete pod analytics-client-pod
```

Sample response: pod "analytics-client-pod" deleted

Wait for the response to ensure that the pod was successfully deleted. Then wait a few minutes for a new pod to start automatically. To check the status of the pods, run the following command:

```
kubectl get pods
```

- Navigate to the Analytics page and refresh to verify that the dashboard now displays correctly.
- (Optional) If you have a backup of `ui` or `.kibana-6`, you restore it now as explained in [Backing up and restoring the analytics database](#) and [Backing up and restoring the analytics database](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Administering user access

If you have permission to administer users in IBM® API Connect, you can add and delete users. After users are removed from an organization through deletion, the user account remains in API Manager.

### About this task

You can also authorize users to perform different roles within your organization. To make any changes to the users that can access your organizations, use the following tasks:

- [Viewing your user information](#)  
You can view the users that can access your IBM API Connect organization.
- [Adding provider organization users and assigning roles](#)  
If you have the permissions that are required to edit users in IBM API Connect, you can add users to your provider organization, remove users, assign roles and perform other user administration tasks.
- [Creating custom roles](#)  
If you have permission to edit roles in IBM API Connect, you can create custom roles, and assign permissions, in a provider organization. You can create as many custom roles as you want.
- [Removing a user from a provider organization](#)  
If you have permission to edit users, you can remove a user from a provider organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Viewing your user information

You can view the users that can access your IBM® API Connect organization.

### Procedure

To view your organization user information, complete the following steps:

In the navigation pane of the API Manager UI, click  Members.

### Results

All of your user accounts are listed.

### What to do next

You can modify your user accounts; see [Adding provider organization users and assigning roles](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding provider organization users and assigning roles

If you have the permissions that are required to edit users in IBM® API Connect, you can add users to your provider organization, remove users, assign roles and perform other user administration tasks.

### About this task


---

You can add a user to your provider organization by inviting them to be an organization member; the user receives an invitation email with an activation link that enables them to complete the addition operation.

### Procedure

---

To add users and assign user roles for your provider organization, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Members.
2. Click Invite member.
3. Enter the email address of the user.
4. Select the roles that you want to assign to the user.
5. Click Invite.

### Results

---

The user is added to the list of provider organization members, and an email invitation is sent. The status is shown as `Pending` until the recipient of the email clicks the link in the email to complete the creation of their account, after which the status changes to `Enabled`.

### What to do next

---

The new user can access the API Manager user interface. The user's authorization within API Manager is defined by the roles that are assigned to them.

### Related tasks

---

- [Creating custom roles](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating custom roles

If you have permission to edit roles in IBM® API Connect, you can create custom roles, and assign permissions, in a provider organization. You can create as many custom roles as you want.

### About this task

---

Permissions can be applied on an organizational or Catalog level.



For a description of the permissions, and details of the default user roles and default permissions assigned to those roles, see [API Connect user roles](#).

Note: In API Manager, the Organization Owner role has full access and cannot be edited or deleted. All other roles, including custom roles, can be deleted. If you delete a role, users lose that role. If a user loses that role, their account remains in API Manager, enabling you to add a role to the user at a future date.

### Procedure

---

To create a custom role, complete the following steps:

1. In the navigation pane of the API Manager UI, click  Settings.
2. Click Roles, then click Add.  
The Create a New Role page opens.
3. Enter a role Title and an optional Summary. A Name is entered automatically.  
Note: The value in the Name field is a single string that is used to identify the role in developer toolkit CLI commands. The Title is used for display.  
To view the CLI commands to manage roles, see [apic roles](#).
4. Use the check boxes to assign permissions to the new role.
5. When you are finished, click Save.
6. To delete a role, click the options icon  alongside the role that you want to delete, click Delete this role, then click Confirm to complete the role deletion.

Note: You can delete a role only at the provider organization level. You cannot delete a role at the Catalog level. Nor can you delete a role at the Space level. You can, however, assign Catalog-specific permissions to the role; for details, see [Creating and configuring Catalogs](#). You can also assign Space-specific permissions; for details, see [Managing user access in a Space](#); for more information on Spaces, see [Using syndication in API Connect](#).

## Results

---

The custom role is created and assigned the permissions that you selected.

## What to do next

---

Assign the custom role to a user.

## Related tasks

---

- [Adding provider organization users and assigning roles](#)
- [Creating and configuring Catalogs](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Removing a user from a provider organization

---

If you have permission to edit users, you can remove a user from a provider organization.

## About this task

---

After you have removed the user from your IBM® API Connect provider organization, their resources are deleted and the user cannot access any of the organization's artifacts.



Note: The user account of the deleted user remains in the associated API Connect user registry. The user can still log in to the API Manager user interface but only information links are available; there is no access to view or manage any resources.

If you later invite the same user again, they can re-activate their account. If you are using a Local User Registry, the user must re-activate their account by using the Sign In option, **not** by completing the registration form and using the Sign Up option; attempting to re-register will fail. For details on how to invite a user to join your organization, see [Adding provider organization users and assigning roles](#).

## Procedure

---

To remove a user from a provider organization complete the following steps:

1. In the navigation pane of the API Manager UI, click  Members.
2. Alongside the user that you want to delete, click the options icon , click Delete, then click Confirm to complete the user deletion.  
Note: You cannot delete the owner of a provider organization.

## Results

---

The user is removed from the provider organization, and the user account remains in API Manager.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing your API Manager password


---

You can change your API Manager password in IBM® API Connect.

## Procedure

---

To change your password, complete the following steps:

1. Log in to the API Manager user interface.
2. Click the User icon  then click Account.
3. In the Change password section, enter your current password and your new password, and confirm the new password.  
Passwords must satisfy the following requirements:
  - Cannot be re-used for at least 8 password cycles
  - Cannot contain more than 2 consecutive repeating characters

- Must include at least 8 characters
- Must include 1 or more characters from at least 2 of the following categories:
  - Lowercase letters
  - Uppercase letters
  - Numbers
  - Special characters – only the following special characters are allowed:

```
!
\"
#
$
%
&
\
(
)
*
+
,
-
.
/
:
;
<
=
>
?
@
[
\\
]
^
_
{
|
}
~
```

4. Click Change Password.

## Results

Your password is changed.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Resolving login problems by increasing HTTP header size

You can resolve login problems for the API Manager UI by increasing the maximum HTTP client header size.

### About this task

Login attempts to the user interface might fail with error **502 (Bad Gateway)**. This error can occur when logging in from a browser that has a large number of cookies. The error occurs because size of the login request exceeds the maximum HTTP client header size.

The maximum HTTP client header size is limited for security reasons. To workaround this issue, you can clear the browser cache and cookies, or open an incognito window from the browser, and then retry the login.

Alternatively, you can increase the maximum HTTP client header size. Use caution when increasing the maximum size because larger headers can raise a security risk to your system.

### Procedure

1. **SSH** to your management system or appliance.
2. Enter the following commands:

```
kubectl get deployment -n namespace | grep apim-v2
kubectl edit deployment -n namespace apiconnect-apim-v2-deployment
```

3. In the **env** section of the deployment configuration, change **--max-http-header-size=12000** to a larger value that works for your environment. For example:

```
- name: NODE_OPTIONS
  value: --max-old-space-size=8094 --max-http-header-size=16000
```

4. Save the updated configuration.
5. Run `kubectl get pod -n namespace` a few times until the new `apiconnect-apim-v2` pod is up and running.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Reference

Reference information for the API Manager component in IBM® API Connect.

- [API gateway response codes](#)

When an API is called, different HTTP status codes are returned by the gateway to indicate whether the request was successfully completed.

---

## Related information

- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## API gateway response codes

When an API is called, different HTTP status codes are returned by the gateway to indicate whether the request was successfully completed.

The response codes used in IBM® API Connect correspond to the registered HTTP status codes that are typically generated to provide informational (1xx), successful (2xx), redirection (3xx), client error (4xx), or server error (5xx) responses, as described at <https://tools.ietf.org/html/rfc7231#section-6> and [Hypertext Transfer Protocol \(HTTP\) Status Code Registry](#).

In API Connect, successful responses vary depending on the API being called. Other response codes can also be generated, depending on the implementation of the assembly and the response from the external systems. The standard reasons listed for the registered HTTP status codes are considered adequate for most responses that are returned; these response codes and their causes are therefore not listed here.

In certain cases, a client or server error response code can be caused by a condition that is specific to API Connect. The following table contains a list of these error response codes and identifies possible causes for these codes being returned. For some error codes, multiple causes are possible.

Table 1. Error codes and their causes

Error Code	Cause
401 Unauthorized	<ul style="list-style-type: none"><li>• The required client identification has not been successfully provided.</li><li>• User authentication failed or did not take place.</li><li>• The application is not registered with the plan that is used.</li><li>• The application is not active.</li></ul>
404 Not Found	<ul style="list-style-type: none"><li>• Information for the provider organization or environment was not found.</li><li>• The API URL was not found in the organization or environment.</li></ul>
405 Method Not Allowed	The API URL was found, but no operation was found that supports the requested HTTP verb.
406 Not Acceptable	The API cannot produce any responses that are supported by the application.
429 Too Many Requests	The rate limit has been exceeded for the plan or operation being used.
500 Internal Server Error	An error occurred while executing this request.
503 Service Unavailable	The status of an API was switched from online to offline, making the API unavailable across all Products in which it is contained. For more information, see <a href="#">Managing your Products in the API Manager UI</a> and <a href="#">Managing API Products using the developer toolkit</a> .

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## API Manager tutorials

Tutorials for using the API Manager user interface in IBM® API Connect.

---

## Prerequisites

Because you are working online, using API Manager, to complete the following tutorials you must be the Organization Manager of an API Connect account. Or, you received an invite to join API Connect as an Administrator. See [Administering Consumer organizations](#) for more information.

If you have received an email inviting you to create an API Connect user account, follow the instructions in your email to create your API Connect account.

## Overview

---

In the following tutorials, you will use API Manager to create simple SOAP API definitions that can be accessed through the Developer Portal and called through the gateway in the same manner as other API Connect API definitions.

- **[Tutorial: Creating a proxy REST API definition](#)**  
This tutorial shows you how to define and implement a REST API definition that proxies an existing service.
- **[Tutorial: Creating a SOAP API](#)**  
This tutorial shows you how to create an API definition by using a SOAP service Web Service Definition Language (WSDL). This API definition simplifies creating and managing access to the SOAP service.
- **[Tutorial: Creating a REST API definition that invokes an existing SOAP service](#)**  
This tutorial shows you how to expose an existing SOAP service and transform the XML data that is returned by it into specified JSON data.
- **[Tutorial: Mapping JSON Content](#)**  
This tutorial shows you how to map message content from one format or schema to another format or schema.
- **[Tutorial: Importing an API](#)**  
This tutorial shows you how to import an existing OpenAPI 2.0 definition.
- **[Tutorial: Supersede a Product](#)**  
This tutorial shows you how to supersede a published product with another product.
- **[Tutorial: Implementing OAuth Security](#)**  
This tutorial shows you how to create a native OAuth provider using API Manager.
- **[Tutorial: Implementing OpenID Connect Security](#)**  
This tutorial shows you how to add OpenID Connect capability to an existing native OAuth provider using API Manager.
- **[Tutorial: Generate a JSON Web Token \(JWT\)](#)**  
This tutorial shows you how to define and implement a REST API definition that generates a JSON Web Token (JWT).
- **[Tutorial: Validate a JSON Web Token \(JWT\)](#)**  
This tutorial shows you how to define and implement a REST API definition that validates a JSON Web Token (JWT).
- **[Tutorial: Creating a Client Application](#)**  
This tutorial shows you how to create a client application using the API Manager. The ability to create a new application using the API Manager allows members of a provider organization, such as API developers or testers, to create the environment needed to complete an API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Creating a proxy REST API definition

This tutorial shows you how to define and implement a REST API definition that proxies an existing service.

### About this tutorial

---

In this tutorial you will complete the following lessons:

1. [Creating a REST API definition](#)
2. [Testing the REST API](#)

Note: The Sandbox catalog must be configured to use either a DataPower® API Gateway, or a DataPower Gateway (v5 compatible), or both. See [Creating and configuring Catalogs](#).

### Creating a REST API definition

---

Add and define a REST API to return the branch details of an example BankA.

To add and define a REST API, complete the following steps:

1. Log in to API Manager.
2. In the Welcome page, click the Develop APIs and Products tile.

Welcome to the API Manager  
Choose an option to get started

- Develop APIs and Products**  
Edit, assemble, secure and test APIs.  
Package APIs using products for publishing to consumers.
- Manage Catalogs**  
Manage active APIs and consumers
- Manage Resources**  
Configure user registries, OAuth providers and TLS
- Manage Settings**  
Edit settings for roles, notifications and more.
- Learn more**  
Documentation and tutorials with step-by-step instructions
- Connect**  
Find expert answers in the API Connect community forum

3. Click Add > API.

Develop

APIs and Products

Add ▾

- API
- Product

You haven't added any APIs or Products

4. Select From target service. Click Next.



Create

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations

Import

- Existing OpenAPI**  
Use an existing definition of a REST proxy

Cancel Next

- Enter the appropriate information to create a REST API definition.
  - In the Title field, enter `Branches`.
  - The Name and Base Path fields autopopulate with the terms `branches` and `/branches` respectively.
  - Leave the Version field at `1.0.0`.
  - In the Target Service URL field, enter `https://apictutorials.mybluemix.net/branches`.

### Info

Enter details of this API

**Title**  
Branches

**Name**  
branches

**Version**  
1.0.0

**Base path (optional)**  
/branches

**Description (optional)**

---

**Target Service URL**  
Enter the URL for the target service you would like to proxy

**Target service URL**  
https://apictutorials.mybluemix.net/branches

6. Click Next.
7. In the Security dialog, take the following steps.
  - a. Select Limit API calls on a per key basis.
  - b. Click Next.

8. You see the progress as the new API gets created. When it is done, you see a Summary. Click Edit API.

9. On the Design page, click Definitions in the side bar.
10. Click Add.

11. In the Name field, enter address, and a Description of The format of the address object.
12. Using the same Definitions panel, configure the Properties definition according to the following table. Create new properties by clicking Add.

Table 1. Properties

Property Name	Description	Type	Example
street1	The first line of the address	string	4660 La Jolla Village Drive
street2	The second line of the address	string	Suite 300
city	The city of the address	string	San Diego
state	The state of the address	string	CA
zip_code	The zip code of the address	string	92122

**Name**  
address

**Type**  
object

**Description (optional)**

**Properties** Add

PROPERTIES NAME	PROPERTIES TYPE	PROPERTIES EXAMPLE	PROPERTIES DESCRIPTION	DELETE
street1	string	4660 La Jolla Village Drive	first line of the address	
street2	string	Suite 300	second line of the address	
city	string	San Diego	city of the address	
state	string	CA	state of the address	
zip_code	string	92122	zip code of the address	

Additional properties

Cancel Save

This is an OpenAPI schema definition and is presented to developers in the Developer Portal to provide them with information about the type of data to expect in their response.

13. Click Save.
14. Create a second definition by clicking Add in the Definitions panel.
15. Name the definition `branch` and, in the Description field, enter `The format of the branch field`.
16. Configure the branch definition to have the properties listed in the following table by creating new properties and editing the default property. Create new properties by clicking Add .

Table 2. Properties

Property Name	Description	Type	Example
address	The address of the branch	address	
type	The type of branch	string	atm
id	The ID of the branch	string	9d72ece0-7e7b-11e5-9038-55f9f9c08c06

## Edit definitions

**Name**  
branch

**Type**  
object

**Description (optional)**

**Properties** Add

PROPERTIES NAME	PROPERTIES TYPE	PROPERTIES EXAMPLE	PROPERTIES DESCRIPTION	DELETE
address	address		Address of branch	
type	string	atm	Type of branch	
id	string	9d72ece0-7e7b-11e5-9038-55f9f9cC	ID of branch	

Additional properties

Cancel Save

Notice that for the address property, the type of the property references another definition within your API and the example is blank. In this manner, you can create complex data structures.

17. Click Save.
18. In the side bar, select Paths to display the Paths panel.

Develop  
Branches 1.0.0

Design Source Assemble

API Setup  
Security Definitions  
Security  
**Paths**  
Definitions  
Properties  
Target Services  
Categories

**Paths** Add

NAME

No items found

19. Click Add.
20. In the Path name field, enter /details.
21. In the Operations section, click Add.
22. Select GET and click Add.

## Path

Paths identify the REST resources exposed by the API. An operation combines a path with an HTTP verb, parameters, and definitions for requests and responses. [Learn more](#)

**Path name**

/details

**Path Parameters** Add

REQUIRED	NAME	LOCATED IN	TYPE	DESCRIPTION	DELETE
----------	------	------------	------	-------------	--------

**Operations** Add

NAME
GET

Cancel Save

23. Click Save.

24. Click /details in the list of available paths.

API Setup

Security Definitions

Security

**Paths**

Definitions

Properties

Target Services

Categories

### Paths

NAME
/details

25. Click GET in the list of Operations.

26. Scroll down. In the Response section, click Add.

a. Enter 200 in the STATUS CODE field.

b. Select `branch` in the SCHEMA field.

c. Enter 200 OK in the DESCRIPTION field.

**Response** Add

STATUS CODE	SCHEMA	DESCRIPTION	DELETE
200	branch	200 OK	

Cancel Save

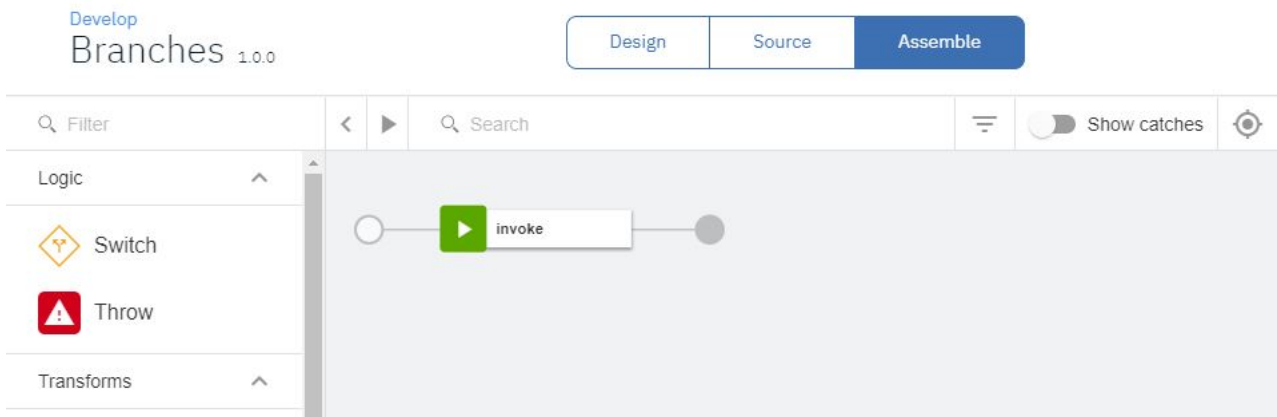
27. Click Save.

## Testing the REST API

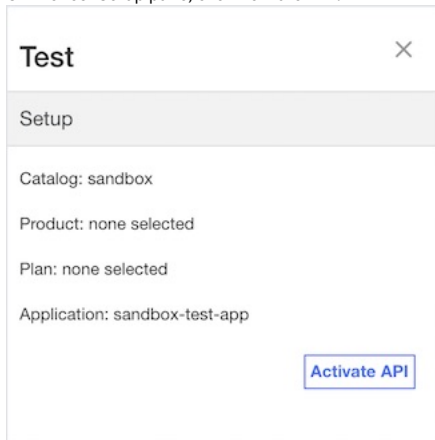
Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test the REST API, complete the following steps:

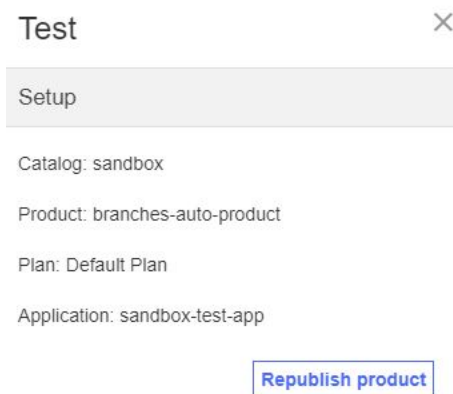
1. Click Assemble.
2. Click the Test icon



3. On the test Setup pane, click Activate API.



4. The Test pane refreshes. A Product and Plan are automatically selected.



5. Scroll down. Select the get /details Operation.

**Operation**

Choose an operation to invoke:

**Operation**

get /details

---

**Identification**

clientId

06b824f9c78a2bc07e81fb635a6a2b03

---

Repeat

Repeat the API invocation a set number of times, or until the stop button is clicked

Stop after:  Stop on error

**Invoke**

6. Click Invoke. You may encounter a yellow error box with a URL embedded in it. Click this URL to override a browser certificate error.

**Response**

Status code:

-1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.

[https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client\\_id=uv0cflw0](https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client_id=uv0cflw0)

7. Click Invoke again. The response contains branch data.

### Test

Invoke

Response

Status code:  
200 OK

Response time:  
4292ms

Headers:  
apim-debug-trans-id: -b5659ae3-1d12-49d5-b984-65556c192fcc  
x-global-transaction-id: 196c55655ade11d600021501  
content-type: application/json; charset=utf-8

Body:

```
[
  {
    "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
    "type": "atm",
    "address": {
      "street1": "600 Anton Blvd.",
      "street2": "Floor 5",
```

## What you did in this tutorial

---

In this tutorial, you completed the following activities:

- Created a REST API definition.
- Tested a REST API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tutorial: Creating a SOAP API

This tutorial shows you how to create an API definition by using a SOAP service Web Service Definition Language (WSDL). This API definition simplifies creating and managing access to the SOAP service.

### About this tutorial

---

In API Manager, you create an API by importing the WSDL for an existing SOAP service. Both SOAP 1.1 and SOAP 1.2 standards are supported by API Connect. When called, the API takes a SOAP request from the API caller and uses it to make its own request to the SOAP service. The API then returns the response of the SOAP service. In this tutorial, the SOAP service returns the balance of an account corresponding to a user identifier.

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

### Creating a SOAP API

---

To create an API for an existing SOAP service, complete the following steps:

1. Download the SOAP WSDL file [AccountService.txt](#). Rename this file `AccountService.wsdl`.
2. Log into the API Manager.
3. Click the Develop APIs and Products tile.



Welcome to the API Manager  
Choose an option to get started

- Develop APIs and Products**  
Edit, assemble, secure and test APIs. Package APIs using products for publishing to consumers.
- Manage Catalogs**  
Manage active APIs and consumers
- Manage Resources**  
Configure user registries, OAuth providers and TLS
- Manage Settings**  
Edit settings for roles, notifications and more.
- Learn more**  
Documentation and tutorials with step-by-step instructions
- Connect**  
Find expert answers in the API Connect community forum

4. Click Add > API.

Develop

APIs and Products

Add ▾

- API
- Product

You haven't added any APIs or Products

5. Select From existing WSDL service (SOAP proxy). Click Next.

## Add API

Create

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations

Import

- Existing OpenAPI**  
Use an existing definition of a REST proxy

Cancel Next

6. Click Browse. Select the `AccountService.wsdl` file from your filesystem. The file uploads and is parsed by API Connect. You see a confirmation. Click Next.

Create from

Select the target wsdl file to create from

AccountService.wsdl X

WSDL has been successfully validated X

Next Cancel

7. Select `AccountService`. Click Next.

### Select Service

Select a WSDL service from the imported file

TITLE	DESCRIPTION
<input checked="" type="checkbox"/> AccountService	getBalance

Cancel Back Next

8. On the Info screen, leave the inputs unchanged. Click Next.

Develop AccountService 1.0.0 Design Source Assemble

API Setup Security Definitions Info Enter the API summary details

9. On the Secure screen, check Limit API calls on a per key basis. In the Activate API section, check Activate API. Click Next.

### Secure

Configure the security of this API

Secure using Client ID

Limit API calls on a per key basis

100 / 1 hour

CORS

### Activate API

This API will be available to be invoked when the following option is enabled.

Activate API

Cancel Back Next

10. The screen now displays progress through the steps to create and publish the new API. When this process completes, you see a summary of results, including the API Endpoint URL of the new API and the credentials of the Application automatically subscribed to the new API.

## Create API from Existing WSDL Service (SOAP proxy)

Generated OpenAPI 2.0 definition

Applied security

Added rate limits

Your API is online!

API Base URL  
URLs for all operations in the API begin with this value.

https://qa- .com/tutorial/sandbox/AccountService

https://qa- .com/tutorial/sandbox/AccountService

API Subscription

Client ID  
c6b72fa575bfd18bce4458aa81391897

Client Secret  
+iH+taX6PUgTQkF7nIMJ1yFZlqEtkstKOYkB2m53Ixc=

Edit API

11. Click Edit API.

You have created a SOAP API and included it in a Product. The WSDL file has provided the information needed to configure the API's inputs and response.

## Testing your SOAP API

Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To add your SOAP API to a Product and Plan, and then test it, complete the following steps:

1. Click Assemble.

Develop AccountService 1.0.0

Design Source Assemble

API Setup

Security Definitions

Info

Enter the API summary details

2. On the Assemble page, click the Test icon. The test tool opens.

Develop AccountService 1.0.0

Design Source Assemble

Filter Search Show catches

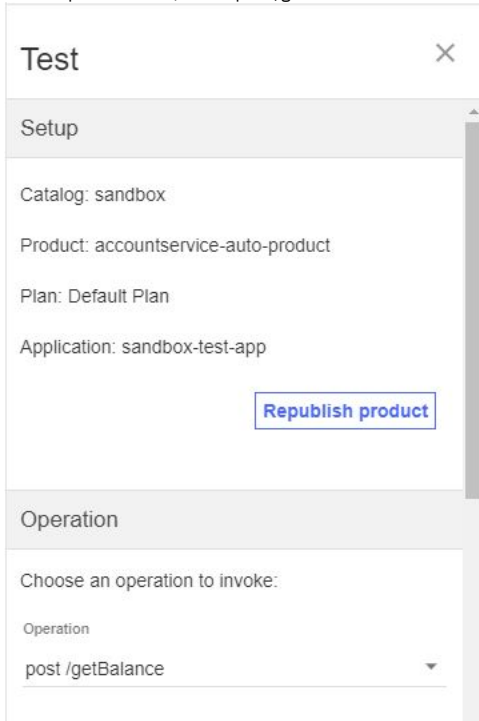
Logic

Switch

Throw

invoke

3. In the Operation field, select post /getBalance.

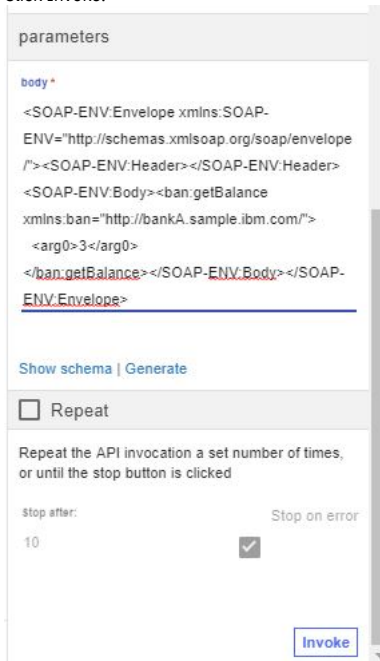


4. In the Body field, of the Parameters section, enter the following request body:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Header></SOAP-ENV:Header><SOAP-ENV:Body><ban:getBalance xmlns:ban="http://bankA.sample.ibm.com/"><arg0>3</arg0></ban:getBalance></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

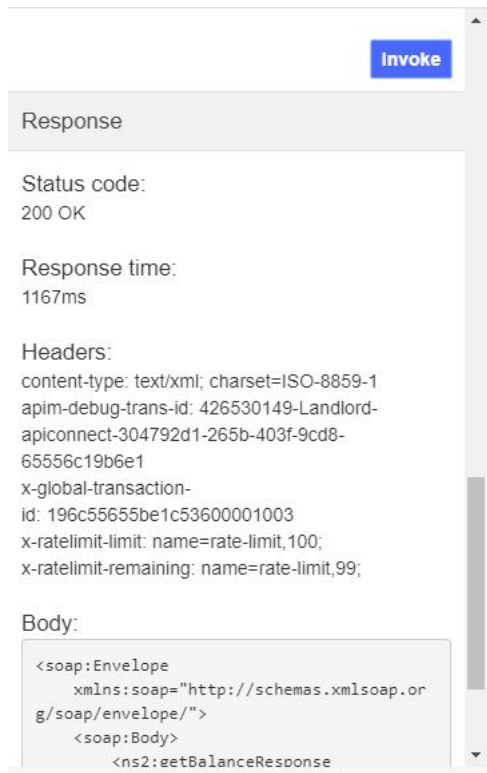
This request body is the same as would ordinarily be passed to the SOAP endpoint, and requests the balance of the account identified by the "arg0" node. However, unlike the SOAP service, the API requires identification using a Client ID, which is enforced by the gateway server.

5. Click Invoke.



The response is displayed.

Note: If you are using a self-signed certificate, you might be prompted to visit a provided URL. Click the link and then accept the certificate before returning to API Manager and clicking Invoke again.



## What you did in this tutorial

---

In this tutorial, you completed the following activities:

- Created a SOAP API
- Tested your SOAP API

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Creating a REST API definition that invokes an existing SOAP service

This tutorial shows you how to expose an existing SOAP service and transform the XML data that is returned by it into specified JSON data.

### About this tutorial

---

In API Manager, you will create a REST API that accesses a SOAP service.

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

### Setting up a REST API definition

---

To set up a REST API, complete the following steps:

1. Download the SOAP WSDL file [AccountServicing.txt](#). Rename this file AccountServicing.wsdl.
2. Log in to API Manager.
3. Click the Develop APIs and Products tile.

Welcome to the API Manager  
Choose an option to get started

- Develop APIs and Products**  
Edit, assemble, secure and test APIs.  
Package APIs using products for publishing to consumers.
- Manage Catalogs**  
Manage active APIs and consumers
- Manage Resources**  
Configure user registries, OAuth providers and TLS
- Manage Settings**  
Edit settings for roles, notifications and more.
- Learn more**  
Documentation and tutorials with step-by-step instructions
- Connect**  
Find expert answers in the API Connect community forum

4. Click Add > API.

Develop

APIs and Products

Add ▾

- API
- Product

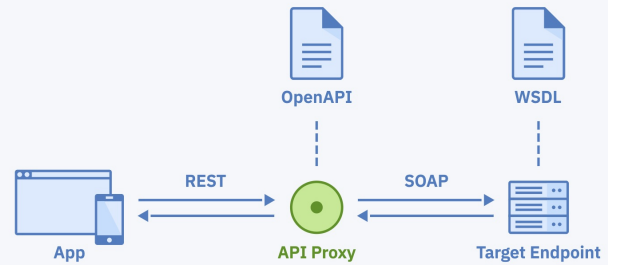
You haven't added any APIs or Products

5. Select From existing WSDL service (REST proxy). Click Next.

## Add API

### Create

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations




### Import

- Existing OpenAPI**  
Use an existing definition of a REST proxy or SOAP API

6. Click Browse. Select the `AccountServicing.wsdl` file from your filesystem. The file uploads and is parsed by API Connect. You see a confirmation. Click Next.  
**Create API from Existing WSDL Service (REST proxy)**

### Create from

Select the target wsdl file to create from

  
`AccountService.wsdl`

✔ WSDL has been successfully validated

Cancel Next

7. On the Select Service screen, click Next.



## Create API from Existing WSDL Service (REST proxy)

### Select Service

Select a WSDL service from the imported file

TITLE	DESCRIPTION
<input checked="" type="checkbox"/> AccountServicing	getBalance

8. On the Info screen, leave the inputs unchanged. Click Next.

## Create API from Existing WSDL Service (REST proxy)

### Info

Enter details of this API

**Title**  
AccountServicing

**Name**  
accountservicing

**Version**  
1.0.0

**Base path (optional)**

**Description (optional)**

9. On the Secure screen, check Limit API calls on a per key basis. In the Activate API section, check Activate API. Click Next.

## Secure

Configure the security of this API

Secure using Client ID

Limit API calls on a per key basis

100 / 1 hour

CORS

## Activate API

This API will be available to be invoked when the following option is enabled.

Activate API

Cancel Back Next

10. The screen now displays progress through the steps to create and publish the new API. When this process completes, you see a summary of results, including the API Endpoint URL of the new API and the credentials of the Application automatically subscribed to the new API. Click Edit API.

## Summary

- Generated OpenAPI 2.0 definition
- Applied security
- Added rate limits
- Your API is online!

API Base URL  
URLs for all operations in the API begin with this value.

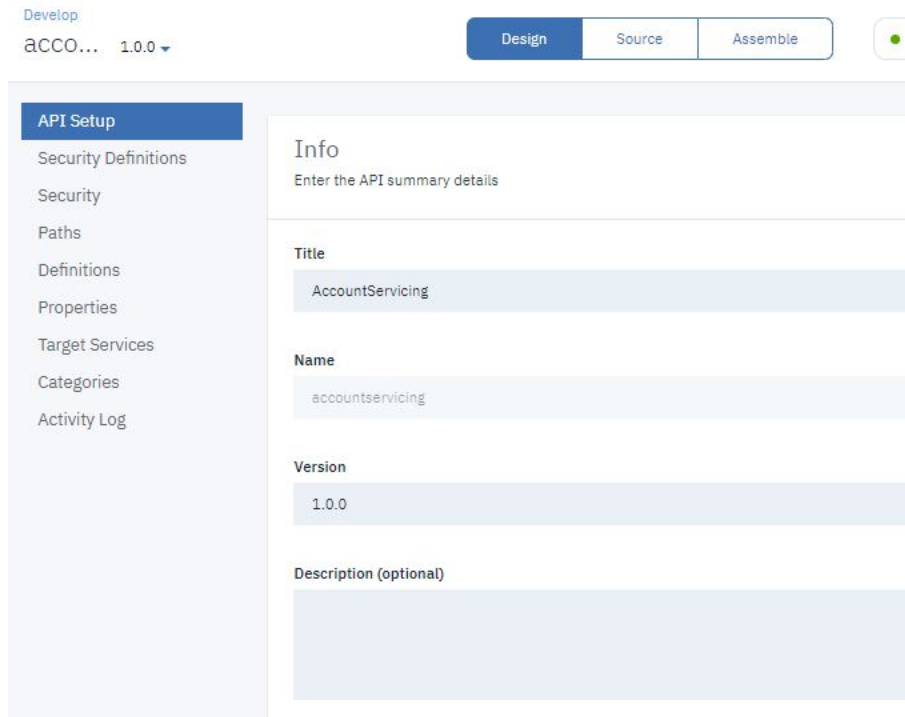
https://apim.ibm.com:9443/tutorial/sandbox/AccountServicing

API Subscription

<b>Client ID</b>	0e6d7b509f69ce5d84dc2f5bb5e93092
<b>Client Secret</b>	3GW8QmPNlrz0F6+tNFCviDwhiK3d0NN0jjZ6+IBR1Is=

Edit API

11. The Design tab for the draft of your API definition opens. Click Assemble.

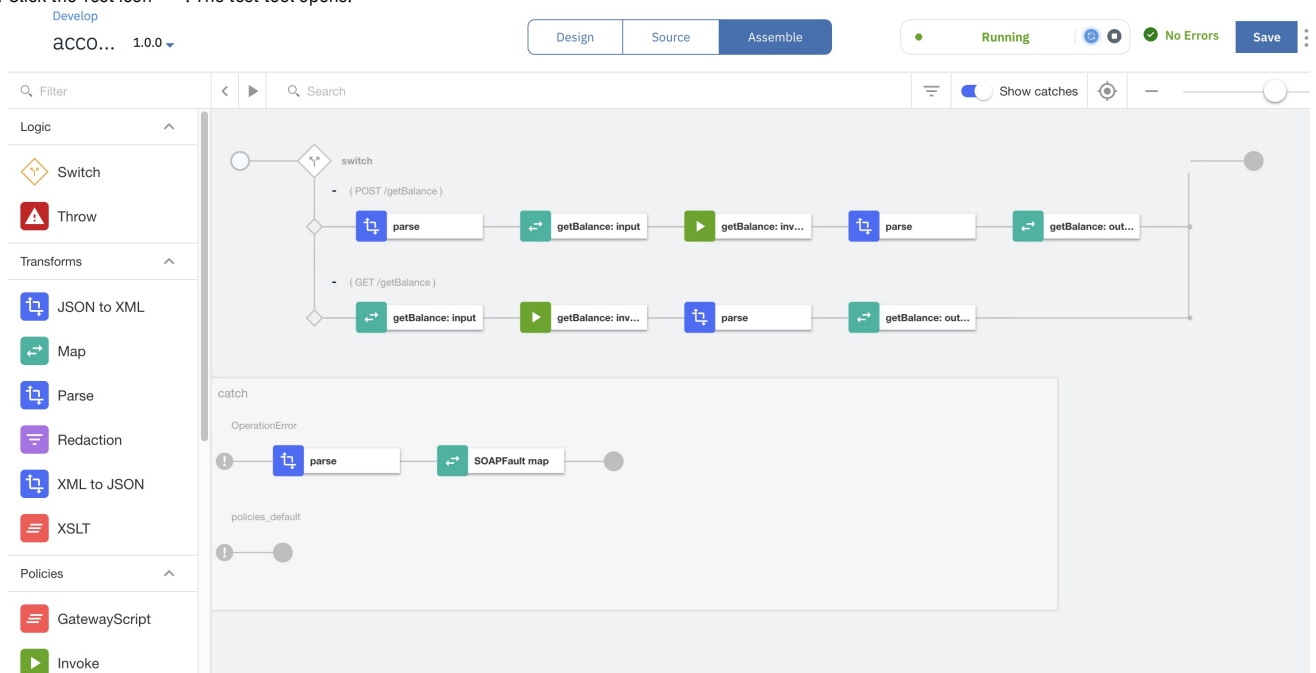


## Test your API definition

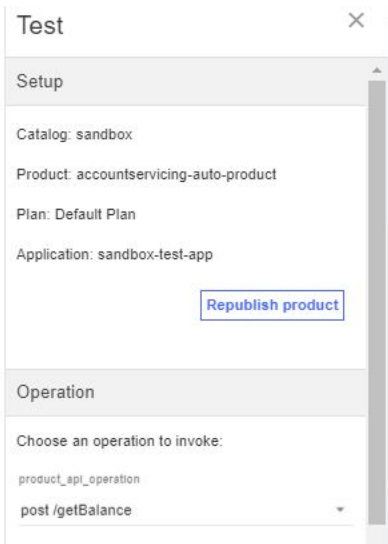
Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test your API definition by using the API Manager test tool, complete the following steps:

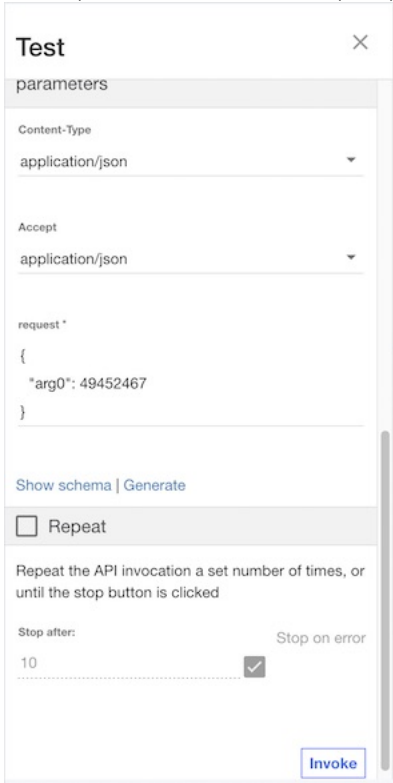
1. Click the Test icon . The test tool opens.



2. In the Operation field, select post /getBalance.



3. In the request field, click Generate. A sample request fills the field.



4. Click Invoke. The response is displayed.

Note: If you are using a self-signed certificate, you might be prompted to visit a provided URL. Click the link and then accept the certificate before returning to API Manager and clicking Invoke again.

Response

Status code:  
200 OK

Response time:  
4390ms

Headers:  
content-language: en-US  
x-global-transaction-id: 196c55655dad6ed000035c2  
x-ratelimit-limit: name=rate-limit,100;  
content-type: application/json  
x-ratelimit-remaining: name=rate-limit,99;

Body:

```
{
  "return": 1584.79
}
```

---

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Set up a REST API definition
- Configured an API to invoke an existing web service and return its output
- Tested your API definition

---

## Related information

- [Creating an API definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Mapping JSON Content

This tutorial shows you how to map message content from one format or schema to another format or schema.

---

## Before You Begin

This task can be completed by users who are assigned one of the following roles:

- Catalog Owner
- Catalog Administrator

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

---

## About this tutorial

In this tutorial you are going to complete the following lessons:

- [Create a New API](#)
- [Map JSON Content](#)

---

## Create a New API

Take the following steps to create a new API.

1. Log in to API Manager.
2. In the Welcome page, click the Develop APIs and Products tile.

Welcome to the API Manager  
Choose an option to get started

- Develop APIs and Products**  
Edit, assemble, secure and test APIs. Package APIs using products for publishing to consumers.
- Manage Catalogs**  
Manage active APIs and consumers
- Manage Resources**  
Configure user registries, OAuth providers and TLS
- Manage Settings**  
Edit settings for roles, notifications and more.
- Learn more**  
Documentation and tutorials with step-by-step instructions
- Connect**  
Find expert answers in the API Connect community forum

3. Click Add > API.

Develop

APIs and Products

Add ▾

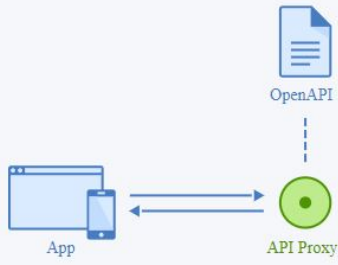
- API
- Product

You haven't added any APIs or Products

4. Select New OpenAPI. Click Next.

**Create**

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations



**Import**

- Existing OpenAPI**  
Use an existing definition of a REST proxy

5. Specify basic information about the API.
  - a. In the Title field, enter `Mapper`.
  - b. Enter `/map` in the Base Path field.
  - c. Enter `1.0.0` the Version field.

**Info**  
Enter details of this API

**Title**  
Mapper

**Name**  
mapper

**Version**  
1.0.0

**Base path (optional)**  
/map

**Description (optional)**

6. Click Next.
7. Make no changes on the Secure screen. Click Next.

### Secure

Configure the security of this API

- Secure using Client ID
- CORS

Cancel Back Next

8. You see the progress as the new API gets created. When it is done, you see a Summary. Click Edit API.

### Summary

- Generated OpenAPI 2.0 definition
- Applied security

[Edit API](#)

9. The Design tab for the draft of your API definition opens.

10. Scroll down. Select application/json in the Consumes section and also in the Produces section.

Develop  
Mapper 1.0.0

Design\* Source Assemble

**API Setup**

- Security Definitions
- Security
- Paths
- Definitions
- Properties
- Target Services
- Categories
- Activity Log

### Consumes

**Consumes (optional)**

- application/json
- application/xml

**Add media type (optional)**

### Produces

**Produces (optional)**

- application/json
- application/xml

**Add media type (optional)**

11. Leave the remaining fields unchanged. Click Save.

12. Click Paths on the navigation bar.

13. Click Add.



API Setup Paths **Add**

Security Definitions

Security


**Paths**

Definitions

Properties

Target Services

NAME

  
No items found

14. In the Path name field of your newly created Path, enter /jsonmap.

### Path

Paths identify the REST resources exposed by the API. An operation combines a path with an HTTP verb, parameters, and definitions for requests and responses. [Learn more](#)

**Path name**


/jsonmap

**Path Parameters** **Add**

REQUIRED	NAME	LOCATED IN	TYPE	DESCRIPTION	DELETE
----------	------	------------	------	-------------	--------

**Operations** **Add**

NAME

  
No items found

**Cancel** **Save**

15. In the Operations section, click Add.
- a. Select POST from the list.
  - b. Click Add.

### Add Operation

**Operations (optional)**

GET

PUT

POST

DELETE

OPTIONS

HEAD

PATCH

16. Click Save.
17. Click /jsonmap in the list of available paths.

Develop

Design
Source
Assemble

## Mapper 1.0.0

API Setup

Security Definitions

Security

Paths

Definitions

Properties

### Paths

NAME
/jsonmap

18. Click the POST operation.
19. Click Add in the Parameters section.

REQUIRED	NAME	LOCATED IN	TYPE	DESCRIPTION	DELETE

STATUS CODE	SCHEMA	DESCRIPTION	DELETE

20. Take the following actions.
  - a. Select **REQUIRED**.
  - b. Enter `body` in the **NAME** field.
  - c. Select `body` in the **LOCATED IN** field.
  - d. Select `string` in the **TYPE** field.

REQUIRED	NAME	LOCATED IN	TYPE	DESCRIPTION	DELETE
<input checked="" type="checkbox"/>	body	body	string		

21. In the Response section, change the DESCRIPTION for the 200 response code to 200 OK.

STATUS CODE	SCHEMA	DESCRIPTION	DELETE
200	string	200 OK	

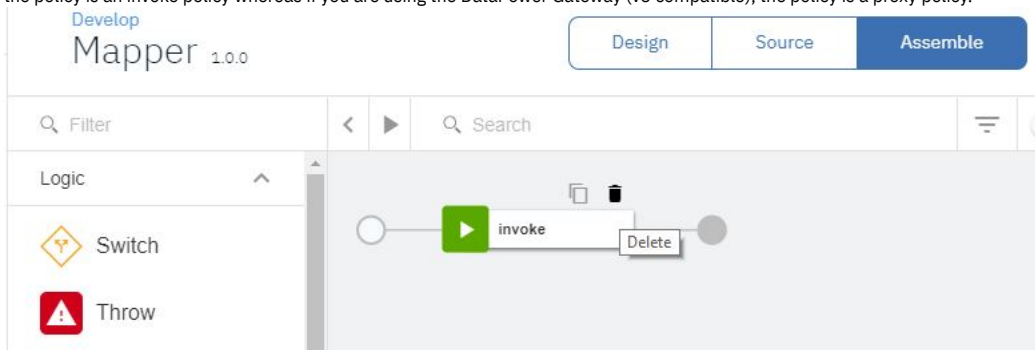
Cancel Save

22. Click Save.

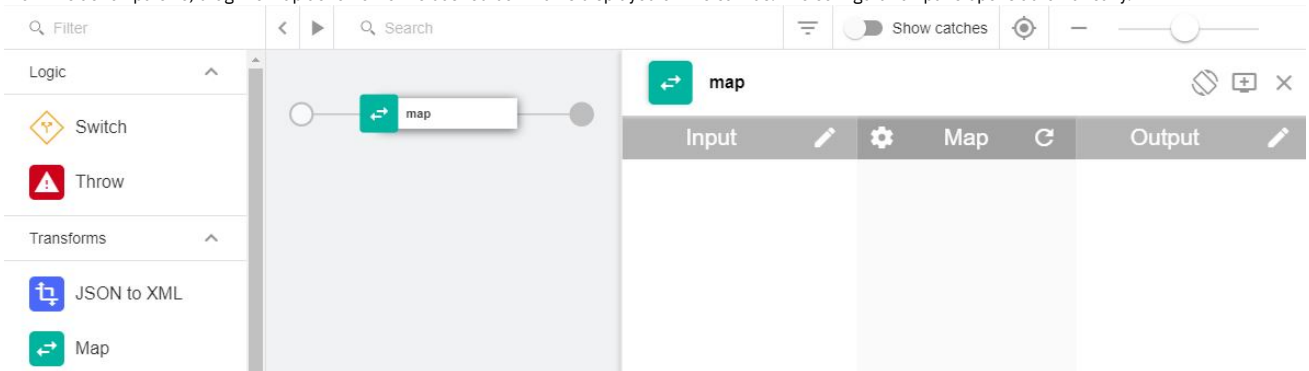
## Map JSON Content

1. Click Assemble.

2. Delete the existing policy on the canvas by hovering your cursor over the policy and then clicking the Delete icon ; if you are using the DataPower API Gateway, the policy is an invoke policy whereas if you are using the DataPower API Gateway (v5 compatible), the policy is a proxy policy.



3. From the action palette, drag the Map action onto the dashed box that is displayed on the canvas. The configuration pane opens automatically.



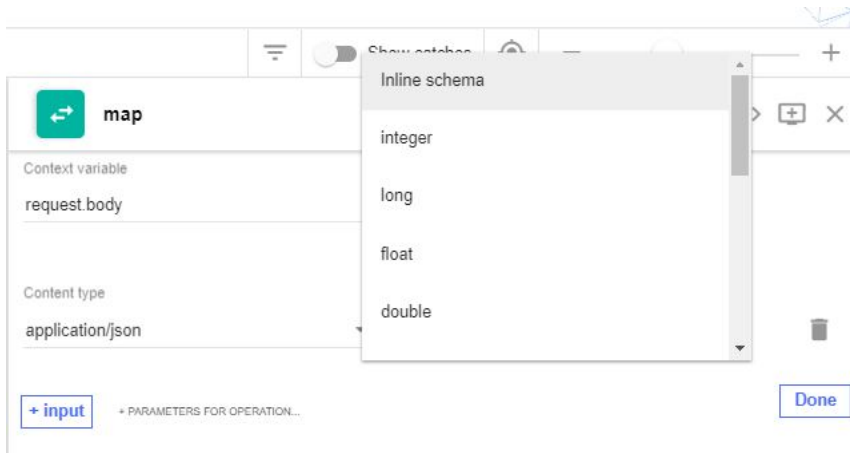
4. Click the Edit inputs icon.

5. Click + input.

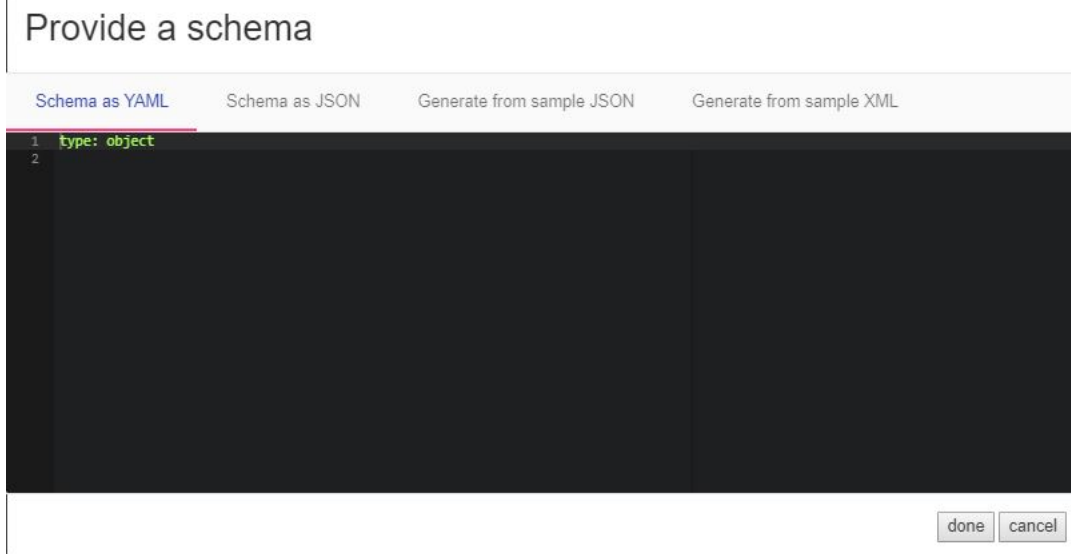
6. Configure the input according to the following table:

Table 1. JSON input

Property	Value
Context variable	request.body
Name	input
Content type	application/json
Definition	Inline schema



7. A dialog pane opens. Click Generate from sample JSON.



8. Paste the following sample JSON into the box.

```
{ "order":
  { "customer":
    { "name":
      {
        "firstname": "John",
        "middlename": "Q",
        "lastname": "Smith"
      },
      "address":
      {
        "line1": "550 King St",
        "line2": "Dept 5",
        "city": "Littleton",
        "state": "MA",
        "country": "USA",
        "code": "01460"
      }
    },
    "items":
    [ {"item": "shoes", "color": "black", "qty": 2, "price": 23.50},
      {"item": "socks", "color": "argyle", "qty": 2, "price": 3.95},
      {"item": "pants", "color": "grey", "qty": 1, "price": 48.00}
    ]
  }
}
```

## Provide a schema

Schema as YAML    Schema as JSON    **Generate from sample JSON**    Generate from sample XML

```

4  {
5  "firstname": "John",
6  "middlename": "Q",
7  "lastname": "Smith"
8  },
9  "address":
10 {
11 "line1": "550 King St",
12 "line2": "Dept 5",
13 "city": "Littleton",
14 "state": "MA",
15 "country": "USA",
16 "code": "01460"
17 }
18 }
19 },
20 "items":
21 [{"item": "shoes", "color": "black", "qty": 2, "price": 23.50},
22 {"item": "socks", "color": "argyle", "qty": 2, "price": 3.95},
23 {"item": "pants", "color": "grey", "qty": 1, "price": 48.00}]
24 }

```

generate    done    cancel

- Click generate.
- The OpenAPI code to describe the schema appears. Click done. The dialog box closes.
- Click Done to complete the Input map configuration.

**map**

Context variable	Name
request.body	input

Content type    Definition \*

application/json    Inline schema

input    + PARAMETERS FOR OPERATION...    Done

- Click the Edit outputs icon in the Output column of the property sheet.

**map**

Input	Map	Output
- input {		
- order {		
- customer {		
- name {		
firstname string		
middlename string		
lastname string		
add property...		
}		
- address {		
line1 string		
line2 string		
city string		
state string		
country string		
code string		
add property...		
}		
}		

- Click + output.
- Configure the input according to the following table:

Table 2. JSON output

Property	Value
Context variable	message.body
Name	output

Property	Value
Content type	application/json
Definition	Inline schema

- A dialog pane opens. Click Generate from sample JSON.
- Paste the following example JSON into the box.

```
{ "order":
  {
    "date": "12-12-12",
    "customer": "John Smith",
    "address":
      {
        "street": "king",
        "citystatezip": "lit MA 01469",
        "country": "USA"
      }
    ,
    "items": [{"type": "shoes", "color": "black", "qty": 2, "price": 23.50}]
  }
}
```

## Provide a schema

Schema as YAML
Schema as JSON
Generate from sample JSON
Generate from sample XML

```
1 { "order":
2   {
3     "date": "12-12-12",
4     "customer": "John Smith",
5     "address":
6       {
7         "street": "king",
8         "citystatezip": "lit MA 01469",
9         "country": "USA"
10      }
11     ,
12     "items": [{"type": "shoes", "color": "black", "qty": 2, "price": 23.50}]
13   }
14 }
```

generate
done
cancel

- Click generate.
- The OpenAPI code to describe the schema appears. Click done. The dialog box closes.
- Click Done.

↔
**map**

🔒
<>
+
×

Context variable	Name
message.body	output

Content type	Definition *
application/json	<div style="display: flex; align-items: center;"> <span style="margin-right: 5px;">▾</span> <span>Inline schema</span> <span style="margin-left: 10px;">▾</span> <span style="margin-left: 5px;">&lt;&gt;</span> <span style="margin-left: 5px;">🗑</span> </div>

+ output
+ OUTPUTS FOR OPERATION...

Done

- Click Save.

Source Assemble\* Save

Show catches

map

Input	Map	Output
<pre> -input {   -order {     -customer {       -name {         firstname string         middlename string         lastname string         add property...       }     }     -address {       line1 string       line2 string       city string       state string       country string       code string       add property...     }   }   add property... </pre>		<pre> -output {   -order {     date string     customer string     -address {       street string       citystatezip string       country string       add property...     }   }   -items [     -{       type string       color string       qty number       price number       add property...     }   ]   add property... </pre>

- Click the handle alongside the *date* field in the Output column. A dialog box opens.
- Enter the following JavaScript code in the box; `new Date().toGMTString()`, then click ok.

### Configure mapping

Value

```
1 new Date().toGMTString()
```

Default

A default value which will be used if the inputs to the map are not defined.

cancel delete ok

- Click the handle alongside the *firstname* field in Input and click the handle alongside the *customer* field in Output.
- Click the handle alongside the *middlename* field in Input and click the handle alongside the *customer* field in Output.
- Click the handle alongside the *lastname* field in Input and click the handle alongside the *customer* field in Output.

<pre> -order {   -customer {     -name {       firstname string       middlename string       lastname string       add property...     }   } </pre>		<pre> -order {   date string   customer string   -address {     street string     citystatezip string     country string     add property...   } </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------	--	---------------------------------------------------------------------------------------------------------------------------------------------------------

- Click the handle alongside the *customer* field in Output. A dialog box opens.
- Enter the following code in the box. Click ok.

```

var name = $(input.order.customer.name.firstname) + ' ';
if($(input.order.customer.name.middlename)) {
  name += $(input.order.customer.name.middlename) + ' ';
}
name += $(input.order.customer.name.lastname);
name

```

### Configure mapping

Mapped from:

- input.order.customer.name.firstname
- input.order.customer.name.middlename
- input.order.customer.name.lastname

Value

```

1 var name = $(input.order.customer.name.firstname) + ' ';
2 if($(input.order.customer.name.middlename)) {
3   name += $(input.order.customer.name.middlename) + ' ';
4   name += $(input.order.customer.name.lastname);
5   name

```

Default

A default value which will be used if the inputs to the map are not defined.

- Click the handle alongside the *address line1* field in Input and click the handle alongside the *street* field in Output.
- Click the handle alongside the *address line2* field in Input and click the handle alongside the *street* field in Output.

- Click the handle alongside *street* in Output. This opens a dialog box.
- Enter the following code in the box. Click ok.

```

var street = $(input.order.customer.address.line1) + ' ';
if($(input.order.customer.address.line2)) {
  street += $(input.order.customer.address.line2);
}
street

```

### Configure mapping

Mapped from:

- input.order.customer.address.line1
- input.order.customer.address.line2

Value

```

1 var street = $(input.order.customer.address.line1) + ' ';
2 if($(input.order.customer.address.line2)) {
3   street += $(input.order.customer.address.line2);
4   street

```

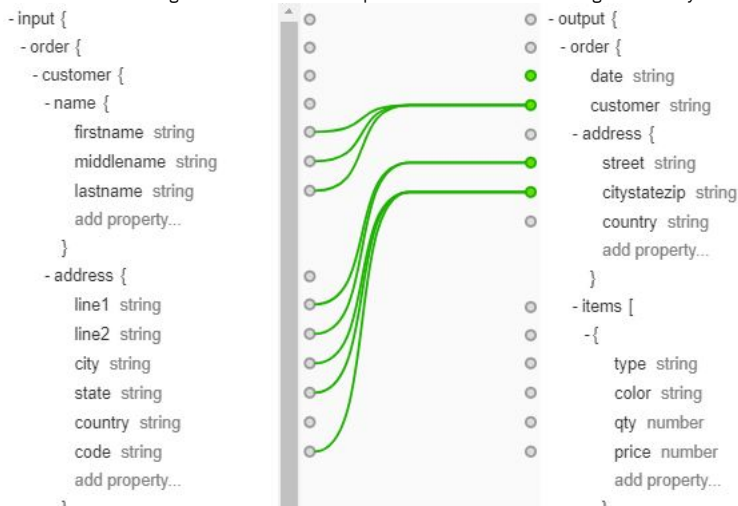
Default

A default value which will be used if the inputs to the map are not defined.

- Click the handle alongside the *city* field in Input and click the handle alongside the *citystatezip* field in Output.



33. Click the handle alongside the *state* field in Input and click the handle alongside the *citystatezip* field in Output.  
 34. Click the handle alongside the *code* field in Input and click the handle alongside the *citystatezip* field in Output.



35. Click the handle alongside the *citystatezip* field in Output. A dialog box opens.  
 36. Enter the following code in the box. Note that the variable references are positional; \$(1) refers to the first mapped value (input.order.customer.city), and so on. Click ok.

`$(1) + " " + $(2) + " " + $(3)`

**Configure mapping**

Mapped from:

- input.order.customer.address.city
- input.order.customer.address.state
- input.order.customer.address.code

Value

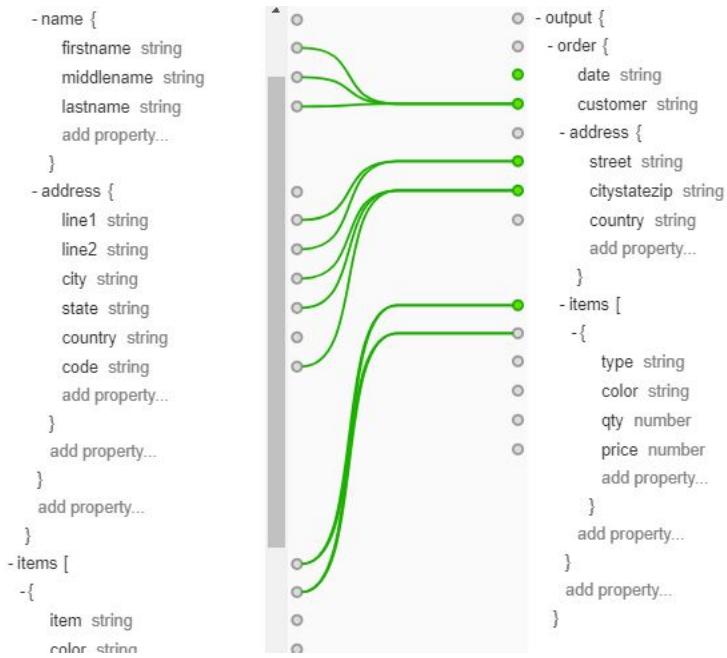
```
1 $(1) + " " + $(2) + " " + $(3)
```

Default

---

A default value which will be used if the inputs to the map are not defined.

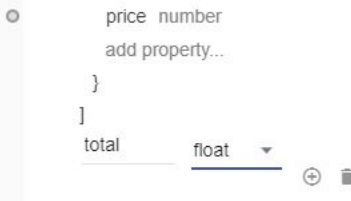
37. Click the handle alongside the *items* field in Input and click the handle alongside the *items* field in Output. An additional wire is drawn automatically.



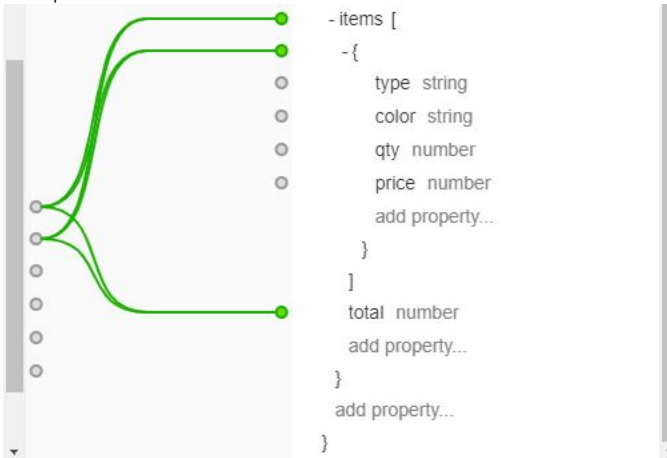
38. Click add property after the closing array bracket - ] - of the Output items array.



39. Enter total in the field that is labeled property. Set the type to float. Click the + icon.



40. Click the handle alongside the items field in Input and click the handle alongside the total field in Output. Note that the system automatically adds a connector to the map.




41. Click the handle alongside total in Output. This opens a dialog box.

42. Enter the following code in the box. This code sums the items for each order. Click ok.

```
$(0) + ($(input.items.price) * $(input.items.qty))
```

## Configure mapping

Mapped from:

 input.items

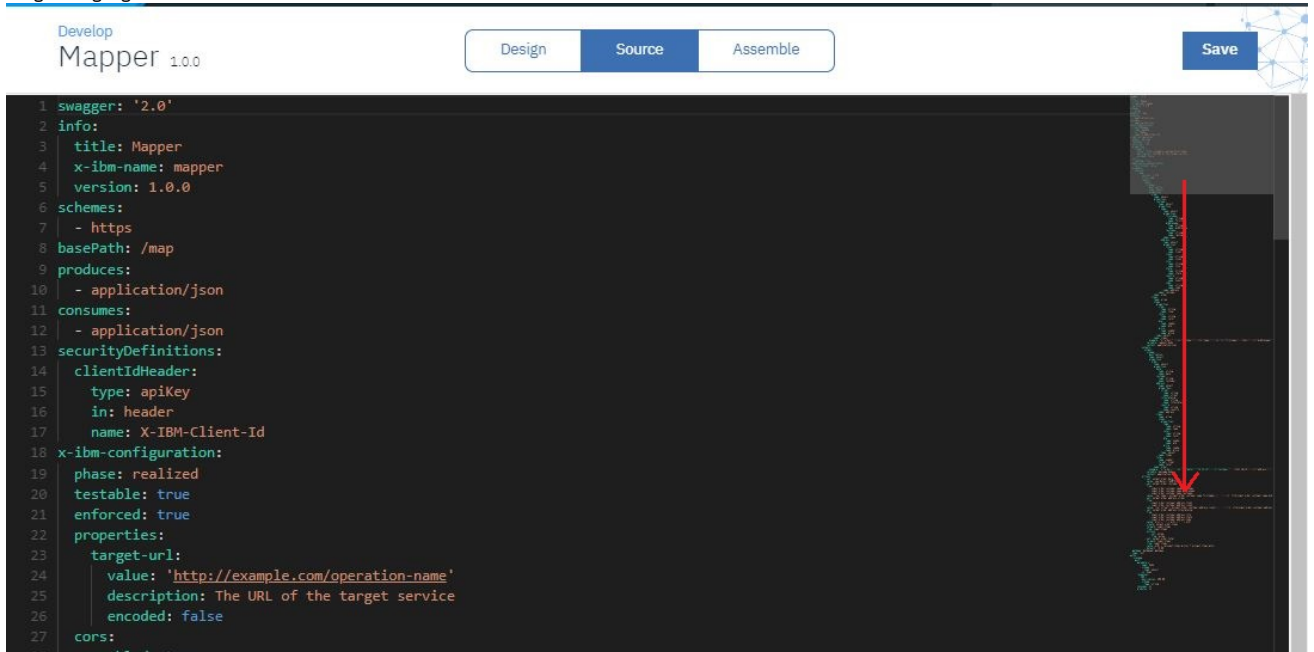
Value

```
1 ${0} + ((${input.items.price} * ${input.items.qty})
```

Default

A default value which will be used if the inputs to the map are not defined.

43. Click the Save icon to save these changes.
44. Click Source.
45. Drag the highlighter to the end of the Source.



```
1 swagger: '2.0'
2 info:
3   title: Mapper
4   x-ibm-name: mapper
5   version: 1.0.0
6 schemes:
7   - https
8 basePath: /map
9 produces:
10  - application/json
11 consumes:
12  - application/json
13 securityDefinitions:
14  clientIdHeader:
15    type: apiKey
16    in: header
17    name: X-IBM-Client-Id
18 x-ibm-configuration:
19  phase: realized
20  testable: true
21  enforced: true
22  properties:
23    target-url:
24      value: 'http://example.com/operation-name'
25      description: The URL of the target service
26      encoded: false
27  cors:
28    enabled: true
```

46. Locate the code where total is calculated.

```
- create: output.order.items
  foreach: input.items
  from: input.items
  actions:
    - set: $item
      from: $item
    - set: output.order.total
      from: input.items
      value: '${0} + (${Input.items.price} * ${Input.items.qty})'
      version: 1.0.0
catch: []
phase: realized
```

47. Insert a new line with the code `foreach: input.items`.

```
- set: output.order.total
  foreach: input.items
  from: input.items
  value: '${0} + (${Input.items.price} * ${Input.items.qty})'
  version: 1.0.0
```

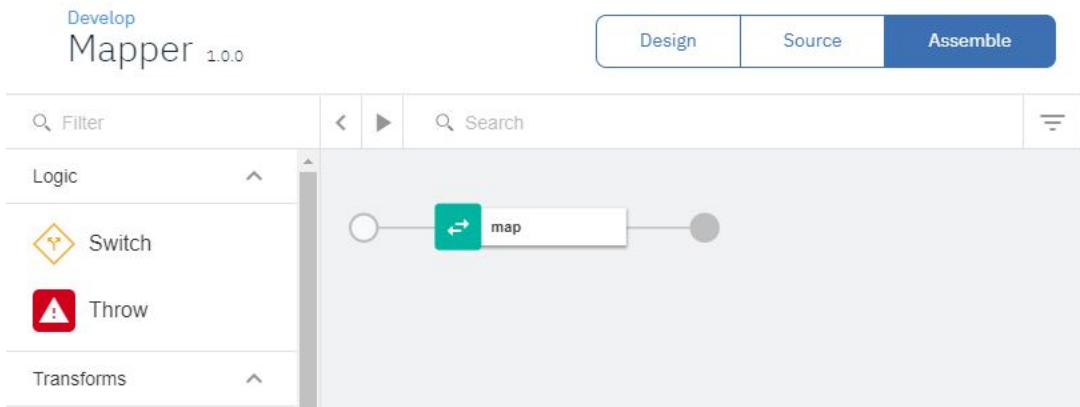
48. Click the Save icon to save these changes.

## Testing your API definition

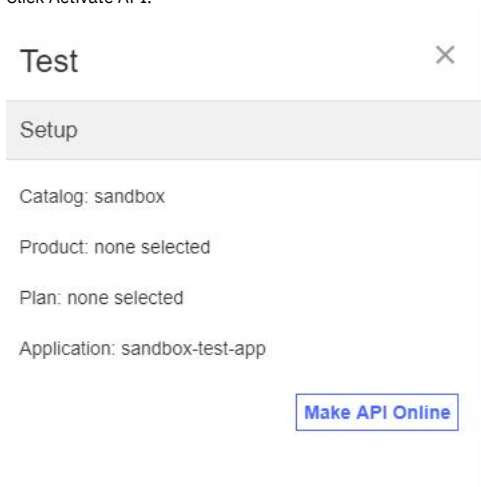
Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test your API definition by using the API Manager test tool, complete the following steps:

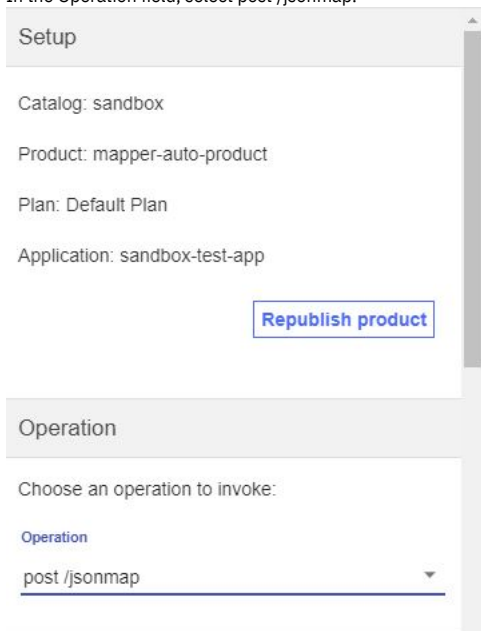
1. Click the Assemble.



2. Click the Test icon . The test tool opens.  
3. Click Activate API.



4. In the Operation field, select post /jsonmap.



5. In the body field, enter the following JSON text.

```
{ "order":  
  { "customer":  
    { "name":  
      {  
        "firstname": "John",  
        "middlename": "Q",  
        "lastname": "Smith"  
      }  
    }  
  }  
}
```

```
"address":
{
"line1": "550 King St",
"line2": "Dept 5",
"city": "Littleton",
"state": "MA",
"country": "USA",
"code": "01460"
}
},
"items":
[{"item": "shoes", "color": "black", "qty": 2, "price": 23.50},
{"item": "socks", "color": "argyle", "qty": 2, "price": 3.95},
{"item": "pants", "color": "grey", "qty": 1, "price": 48.00}]
}
```



6. Click Invoke. The response is displayed.

Note: If you are using a self-signed certificate, you might be prompted to visit a provided URL. Click the link and then accept the certificate before returning to API Manager and clicking Invoke again.

Test
×

Invoke

**Response**

Status code:  
200 OK

Response time:  
1154ms

Headers:  
apim-debug-trans-id: -f7a2d4e1-39f3-4862-93f9-65556c1913fc  
x-global-transaction-id: 196c55655b3e745800014a90  
content-type: application/json

Body:

```

{
  "order": {
    "date": "Thu, 05 Jul 2018 19:41:12 GM
T",
    "customer": "John Q Smith",
    "address": {
      "street": "550 King St Dept 5",
      "citystatezip": "Littleton MA 0146
0"
    },
    "items": [
      {

```

Test
×

```

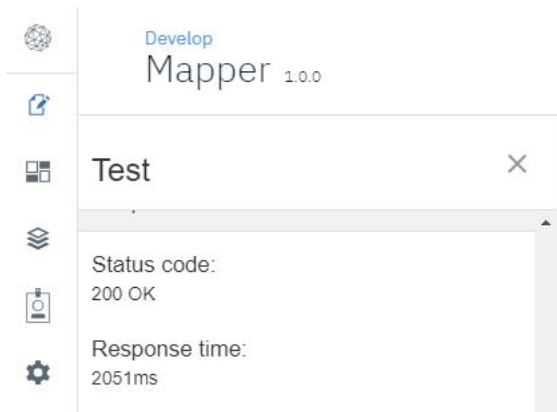
{
  "order": {
    "date": "Fri, 06 Jul 2018 13:18:23 GM
T",
    "customer": "John Q Smith",
    "address": {
      "street": "550 King St Dept 5",
      "citystatezip": "Littleton MA 0146
0"
    },
    "items": [
      {
        "item": "shoes",
        "color": "black",
        "qty": 2,
        "price": 23.5
      },
      {
        "item": "socks",
        "color": "argyle",
        "qty": 2,
        "price": 3.95
      },
      {
        "item": "pants",
        "color": "grey",
        "qty": 1,
        "price": 48
      }
    ],
    "total": 102.9
  }
}

```

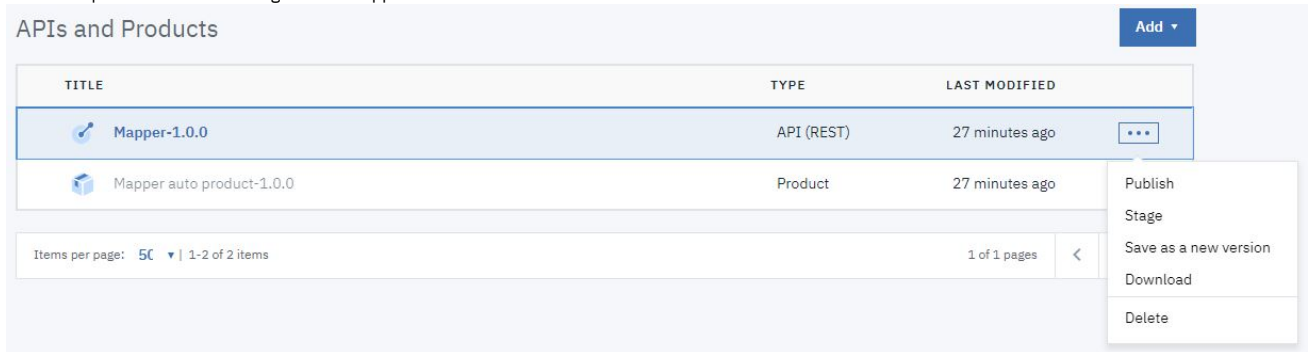
## Manage your API definition

Now that your new API works correctly, you can manage this API. To see your immediate options, take the following steps.

1. Click the Develop icon  on the navigation bar.



2. Click the Options icon  alongside the Mapper API.



3. Select Download.

## What you did in this tutorial

- Created a new API
- Mapped content from one JSON schema to another schema.

## Related information

- [Tutorials home](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Importing an API

This tutorial shows you how to import an existing OpenAPI 2.0 definition.

### About this tutorial

In this tutorial you will complete the following lessons:

1. [Importing an API](#)
2. [Testing the imported API](#)

### Before you begin

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

If your Sandbox catalog uses a DataPower Gateway (v5 compatible) (as one of the gateways, or as the only gateway), download the [findbranch.txt](#) file to your local filesystem. Rename this file `findbranch.yaml`.

If your Sandbox catalog uses a DataPower API Gateway, download the [findbranch\\_v6.txt](#) file to your local filesystem. Rename this file `findbranch.yaml`.

### Importing an API

To import an API, complete the following steps:

1. Log in to API Manager.
2. In the Welcome page, click the Develop APIs and Products tile.

The screenshot shows the IBM API Manager interface. At the top, there is a dark blue header with the text "IBM API Connect API Manager" on the left and "Organization Tutorial" on the right. Below the header, the main content area is titled "Welcome to the API Manager" with the subtitle "Choose an option to get started". A vertical sidebar on the left contains several icons: a globe, a document, a grid, a stack of layers, a clipboard, and a gear. The main area features six white tiles with blue icons and text:

- Develop APIs and Products**: Edit, assemble, secure and test APIs. Package APIs using products for publishing to consumers.
- Manage Catalogs**: Manage active APIs and consumers.
- Manage Resources**: Configure user registries, OAuth providers and TLS.
- Manage Settings**: Edit settings for roles, notifications and more.
- Learn more**: Documentation and tutorials with step-by-step instructions.
- Connect**: Find expert answers in the API Connect community forum.

3. Click Add > API.

The screenshot shows the "Develop" section of the IBM API Manager. The page title is "Develop" and the sub-section is "APIs and Products". A vertical sidebar on the left contains the same icons as the previous screenshot. The main content area features a large white box with a blue icon of an open cardboard box and the text "You haven't added any APIs or Products". In the top right corner of this box, there is a blue "Add" button with a dropdown arrow. The dropdown menu is open, showing two options: "API" (highlighted in blue) and "Product".

4. In the Import section, select Existing OpenAPI. Click Next.



## Add API

### Create

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations

### Import

- Existing OpenAPI**  
Use an existing definition of a REST proxy

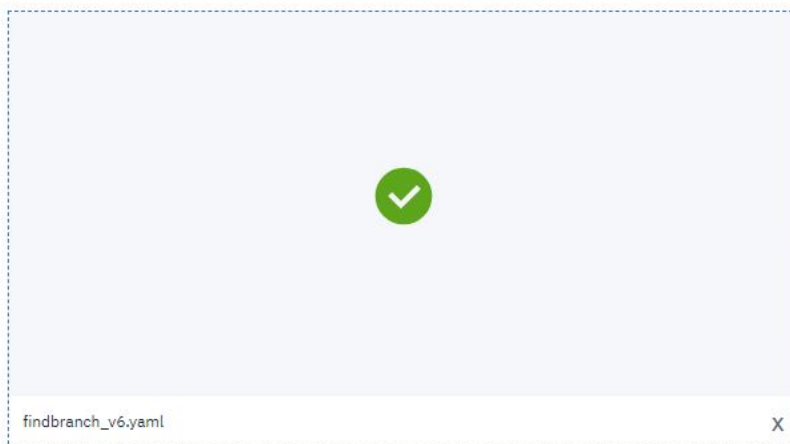
Cancel

Next

- Click Browse. Select the `findbranch.yaml` file on your local filesystem.
- Click Next.

### Import from file

Select the API definition file to import from



Next

Cancel

7. Check Activate API. Click Next.

## Import API

### Activate API

This API will be available to be invoked when the following option is enabled.

Activate API


[Cancel](#) [Back](#) [Next](#)

8. The new API is created, along with a product. Click Edit API.



### Summary

- ✓ Generated OpenAPI 2.0 definition
- ✓ Your API is online!

API Base URL  
URLs for all operations in the API begin with this value.

`https://apimdev .com:9443/tutorial/sandbox/findbranch` 

API Subscription

<b>Client ID</b>	<code>5829ed481ad3592e3fa873d5f250576c</code> 
<b>Client Secret</b>	<code>JORsFyuz6n1g/3lbkQ2hsKL3rK4/W4tDGAomDnJtdPg=</code> 

[Edit API](#)

## Testing the imported API

Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test the imported API, complete the following steps:

1. Click Assemble.

API Setup

Security  
Definitions

Security

Paths

Definitions

Properties

Target Services

Categories

### Info


Enter the API summary details

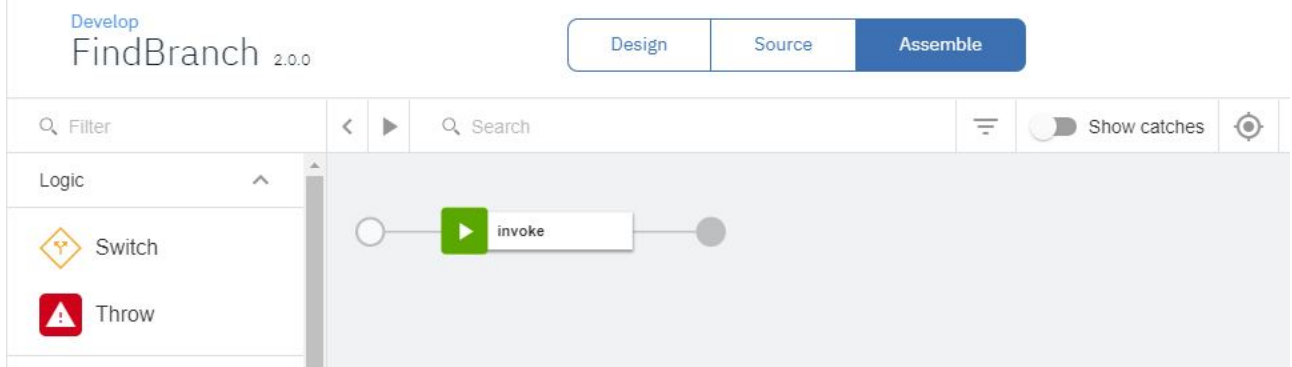
#### Title

FindBranch

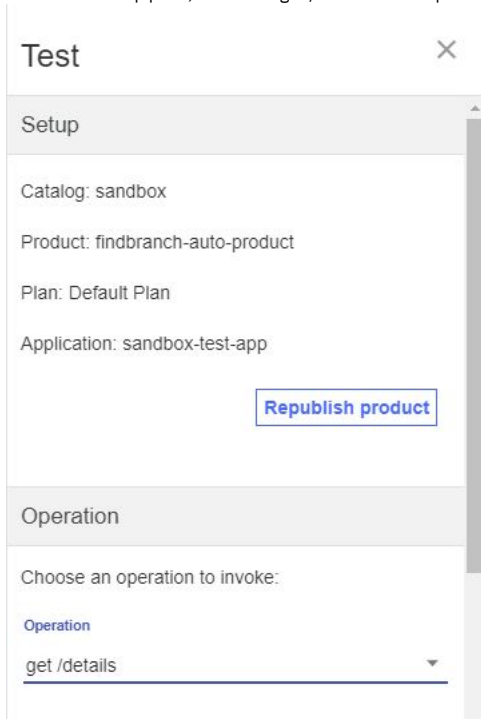
#### Name

findbranch

2. Click the Test icon .



3. On the test Setup pane, select the get /details in the Operations section.



4. Scroll down. Click Invoke. You may encounter a yellow error box with a URL embedded in it. Click this URL to override a browser certificate error.

Response

Status code:  
-1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.

[https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client\\_id=untrusted](https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client_id=untrusted)

5. Click Invoke again. The response contains branch data.

Test ×

Invoke

Response

Status code:  
200 OK

Response time:  
4292ms

Headers:  
apim-debug-trans-id: -b5659ae3-1d12-49d5-b984-65556c192fcc  
x-global-transaction-id: 196c55655ade11d600021501  
content-type: application/json; charset=utf-8

Body:

```
[
  {
    "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
    "type": "atm",
    "address": {
      "street1": "600 Anton Blvd.",
      "street2": "Floor 5",
```

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Imported an API definition.
- Tested an API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Supersede a Product

This tutorial shows you how to supersede a published product with another product.

### Before You Begin

This task can be completed by users who are assigned one of the following roles:

- Catalog Owner
- Catalog Administrator
- Portal User

You must also do the following steps.

- Complete the [Tutorial: Importing an API](#) tutorial.
- You must have a Developer Portal activated for the Sandbox catalog. If you do not, complete the [Tutorial: Creating the Developer Portal](#) tutorial.
- You must have created an application in the Developer Portal and subscribed it to a product. If you do not, complete the [Tutorial: Creating Accounts and Applications on the Developer Portal](#) tutorial.

## About this tutorial

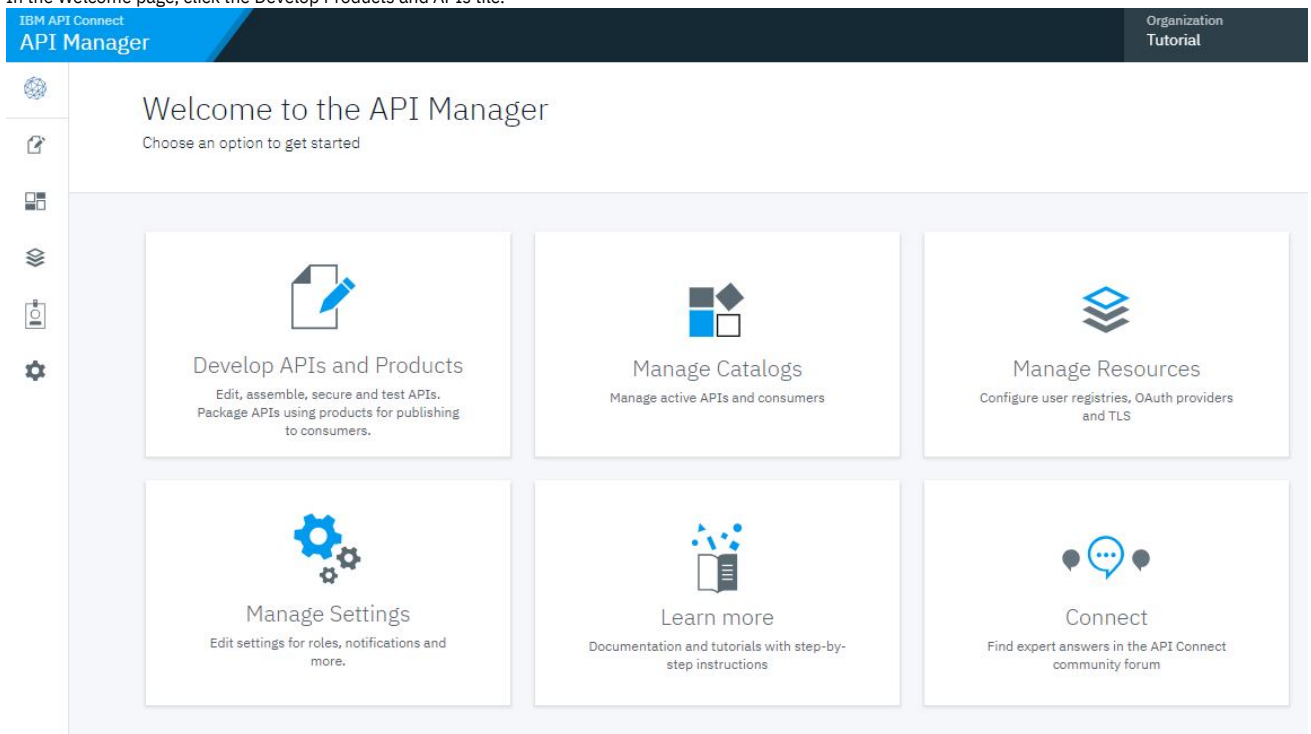
In this tutorial you are going to complete the following lessons:


- [Modify an Existing API](#)
- [Duplicate and Change a Product](#)
- [Supersede a Product](#)
- [Migrate a Product](#)

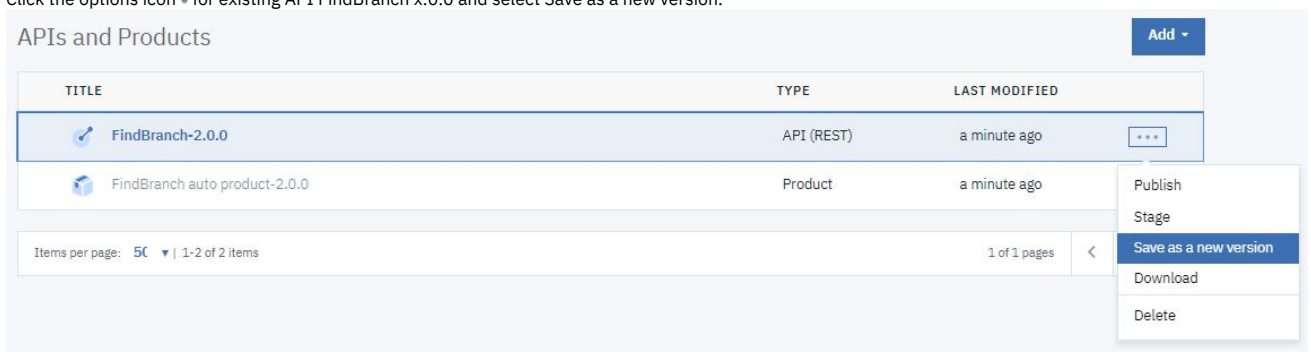
## Modify an Existing API

Take the following steps to modify an existing API.

1. Log in to API Manager.
2. In the Welcome page, click the Develop Products and APIs tile.



3. Click the options icon  for existing API FindBranch x.0.0 and select Save as a new version.



4. In the Version field, enter 3.0.0.
5. Click Save field. The dialog box closes.

## Save API as a new Version

### Version

3.0.0

Cancel

Save

6. Click the arrow to expand the findbranch API listing. Click FindBranch 3.0.0 to edit the new API.

APIs and Products Add

TITLE	TYPE	LAST MODIFIED
<span>findbranch</span>	API (REST)	
<span>FindBranch-2.0.0</span>	API (REST)	10 minutes ago
<span>FindBranch-3.0.0</span>	API (REST)	a few seconds ago
<span>FindBranch auto product-2.0.0</span>	Product	10 minutes ago

7. Click Paths in the navigation bar.  
8. Click /details in the Path list.

Develop  
FindBranch 2.0.0

Design Source Assemble

API Setup  
Security Definitions  
Security  
**Paths**  
Definitions

Paths

NAME
/details

9. Click GET in the Operations section.

## Operations

NAME
GET

10. Scroll down. Click Add in the Parameters section.

11. Take the following actions.

- Enter `loc` in the NAME field.
- Select `query` in the LOCATED IN field.
- Select `string` in the TYPE field.
- Enter `Location hint` in the DESCRIPTION field.

Parameters Add

REQUIRED	NAME	LOCATED IN	TYPE	DESCRIPTION	DELETE
<input type="checkbox"/>	loc	query	string	Location hint	



12. Scroll down. Click Save.

## Duplicate and Change a Product

Take the following steps to duplicate and then modify an existing product.

1. Click the Develop icon  in the navigation bar.
2. Click Add > Product.

APIs and Products

TITLE	TYPE	LAST MODIFIED	API
>  findbranch	API (REST)		API
 FindBranch auto product-2.0.0	Product	14 minutes ago	Product

3. Select New product. Click Next.

Create

**New product**  
Compose a new product by adding rate limits and plans

Import

**Existing product**  
Use an existing definition of a product

4. Enter FindBranches in the Title field. Click Next.

Info

Enter details of the product

**Title**  
FindBranches

**Name**  
findbranches

**Version**  
1.0.0

**Summary (optional)**

5. Select FindBranch 3.0.0 in the APIs field. Click Next.

## APIs

Select the APIs to add to this product

<input type="checkbox"/>	TITLE	VERSION	DESCRIPTION
<input type="checkbox"/>	FindBranch	2.0.0	
<input checked="" type="checkbox"/>	FindBranch	3.0.0	

Cancel

Back

Next

6. Accept the defaults for Plans. Click Next.

## Plans

Add plans to this product

Add

### Default Plan

#### Title

Default Plan

#### Description (optional)

Default Plan

#### Rate Limit

100

/ 1

hour

Cancel

Back

Next

7. Accept the defaults for Publish, Visibility and Subscribability. Click Next.

## Publish

Enable publishing of this product

Publish product

## Visibility

Select the organizations or groups you would like to make this product visible to

- Public  
 Authenticated  
 Custom

## Subscribability

Select the organizations or groups you would like to subscribe to this product

- Authenticated  
 Custom

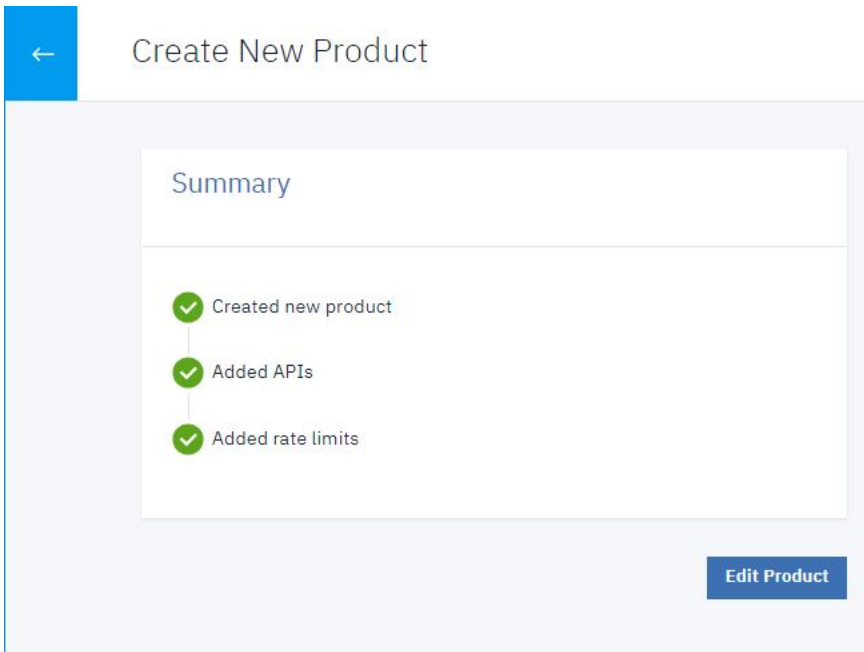
Cancel


Back

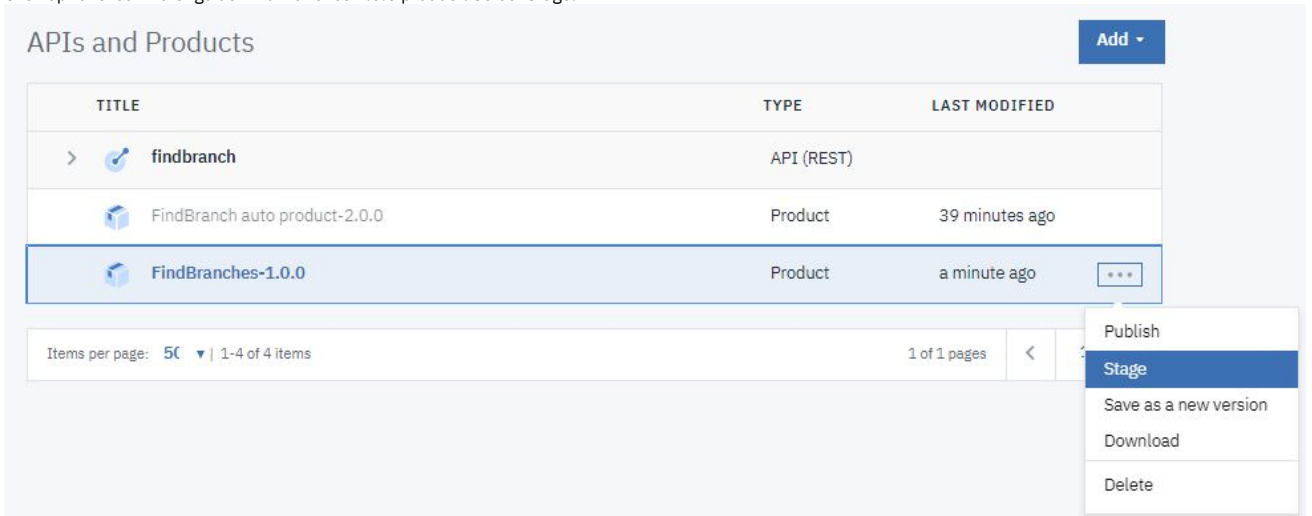
Next

8. Click Back Arrow.

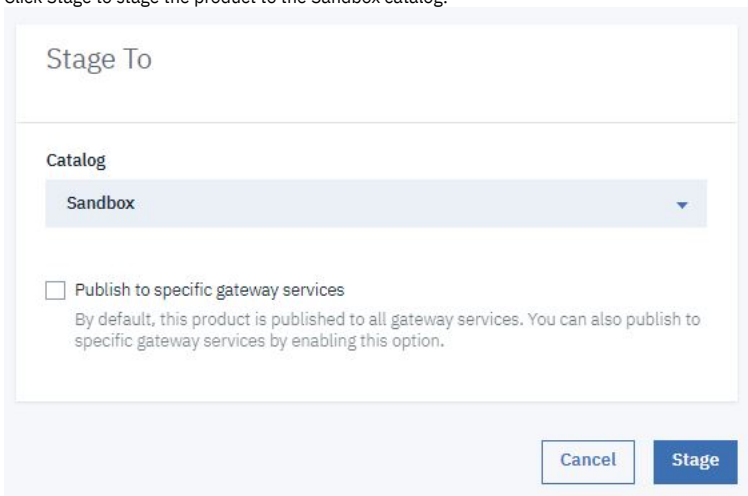




9. Click options icon  alongside FindBranches 1.0.0 product. Select Stage.



10. Click Stage to stage the product to the Sandbox catalog.



## Supersede a Product

1. Click the Manage icon  in the navigation bar.
2. Click the Sandbox catalog icon.

3. Click the options icon  alongside the product FindBranch auto product x.0.0 and select Supersede.

TITLE	NAME	STATE	
> FindBranch auto product	findbranch-auto-product 2.0.0	Published	⋮
> FindBranches	findbranches 1.0.0	Staged	

- Deprecate
- Retire
- Replace
- Supersede**
- Set migration target
- Update gateway services
- Edit visibility

4. Select FindBranches.

Select Product

Select a product to supersede findbranch-auto-product 2.0.0:

Title	Name	State
<input checked="" type="checkbox"/> FindBranches	findbranches 1.0.0	Staged

5. Click Next.

6. Select Default plan in the TARGET column.

7. Click Supersede.

**Supersede**  
FindBranch auto product (findbranch-auto-product:2.0.0)

**With**  
FindBranches (findbranches:1.0.0)

Migrate Plans

SOURCE	TARGET
default	default-plan

Notice the state of both products change. FindBranch auto product x.0.0 is now Deprecated, and FindBranches 1.0.0 is Published. All external applications subscribed to FindBranch auto product x.0.0 remain subscribed but should migrate to FindBranches 1.0.0. The external application developer completes the migration in the Developer Portal.

TITLE	NAME	STATE
> FindBranch auto product	findbranch-auto-product 2.0.0	Deprecated
> FindBranches	findbranches 1.0.0	Published

## Migrate a Product

1. In a new browser window, sign in to the Developer Portal using the account you created in [Tutorial: Creating Accounts and Applications on the Developer Portal](#).
2. Click Apps.

3. Click AppOne.

4. Click Subscriptions.  
Applications



Dashboard Subscriptions

5. Click Migrate this subscription to plan 'default' in product 'FindBranch' at version '2.0.0'.

Subscriptions		
PRODUCT	PLAN	
FindBranch auto product (2.0.0)	Default Plan	Migrate this subscription to plan 'default-plan' in product 'FindBranches' at version '1.0.0' <span>⋮</span>

6. Click Migrate subscription.

## Migrate the subscription for *AppOne*?

Are you sure you want to migrate this subscription? This action cannot be undone.

Cancel

Migrate subscription

7. Notice the subscription has changed to the target product.

Subscriptions		
PRODUCT	PLAN	
FindBranches (1.0.0)	Default Plan	<span>⋮</span>

## What you did in this tutorial

- Created a new API
- Created a new Product
- Superseded an existing published product with the new product
- Migrated a subscribed external application to the new product

## Related information

- [Tutorials home](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Implementing OAuth Security

This tutorial shows you how to create a native OAuth provider using API Manager.

## About this tutorial

In this tutorial you will complete the following lessons:

1. [Create a Native OAuth Provider](#)
2. [Adding OAuth Security to an API](#)
3. [Test OAuth Security](#)

## Before you begin


In this tutorial you will implement and test OAuth security. To complete this tutorial, you must have the following available:

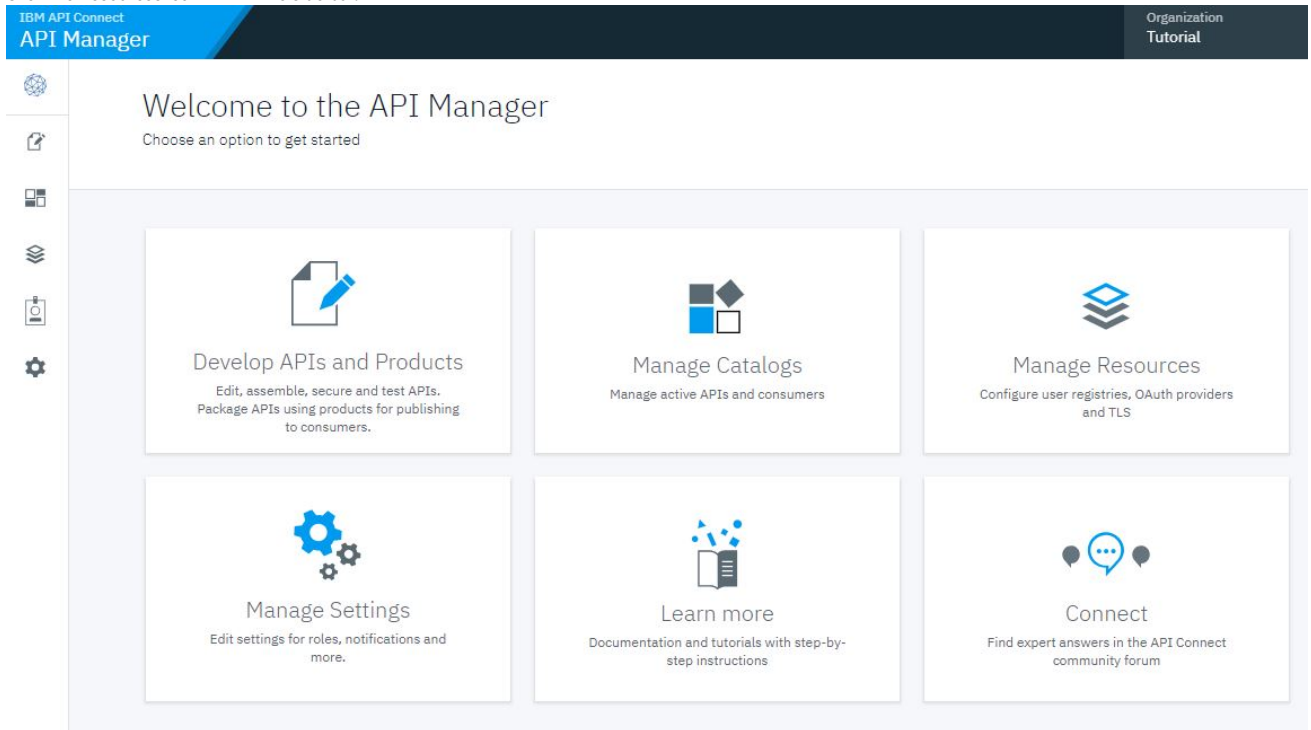
1. An existing published API. If you do not have an API available, complete the [Tutorial: Importing an API](#) tutorial.
2. An existing client application. If you do not have a client application available, complete [Tutorial: Creating a Client Application](#).
3. An external application, such as cURL, used to send requests to the OAuth token provider endpoint.

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

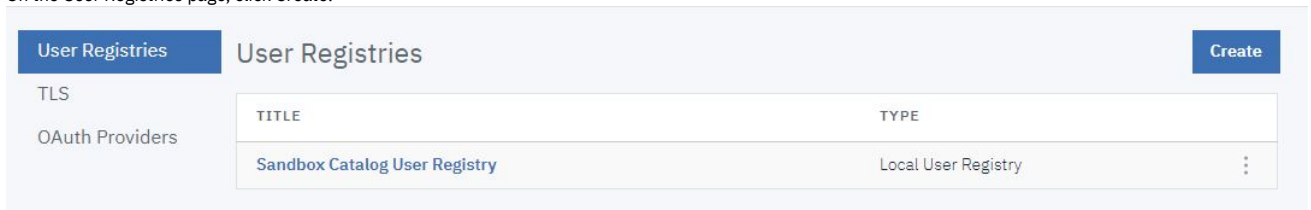
## Create a Native OAuth Provider

To create a native OAuth provider by using API Manager, complete the following steps:

1. Log in to API Manager.
2. Click the Resources icon  in the side bar.

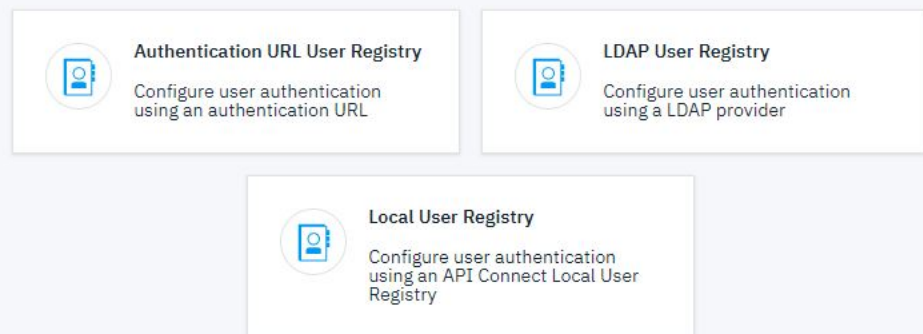


3. On the User Registries page, click Create.



4. Click Authentication URL User Registry.

### Select the user registry type



5. Take the following steps:
  - a. Enter AuthURL in the Title and Display Name fields.
  - b. Enter `https://httpbin.org/basic-auth/user/pass` in the URL field.

- c. Select `No TLS client profile` in the `TLS Client Profile` field.
6. Click `Save`.

### Authentication URL User Registry

**Title**  
AuthURL

**Name**  
authurl

**Display Name**  
AuthURL

**Summary (optional)**

**Url**  
https://httpbin.org/basic-auth/user/pass

**TLS Client Profile (optional)**  
No TLS Profile

Case Sensitive

7. Click `Save`.
8. Click `OAuth Providers`.

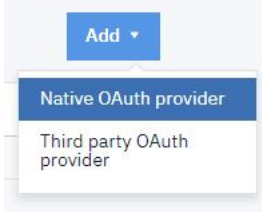
User Registries Create

TLS

OAuth Providers

TITLE	TYPE
AuthURL	Authentication URL
Sandbox Catalog User Registry	Local User Registry

9. Click `Add` Native OAuth provider.



10. Enter `MainProviderOA` in the `Title` field.
11. In the `Gateway Type` section, select the gateway type in use by the catalog. This defaults to `DataPower Gateway (v5 compatible)`. You may need to change this selection.  
 Note: The gateway type selected here must match the gateway type used by the API that uses this provider.  
 Click `Next`.

### Native OAuth Provider

**Title**  
MainProviderOA

**Name**  
mainprovideroa

**Description (optional)**

**Base path (optional)**

---

### Gateway Type

Select the gateway type for this OAuth provider

DataPower Gateway (v5 compatible)  
 DataPower API Gateway

12. Select Resource owner - Password in the Supported grant types field. Deselect any others if necessary.

### Configuration

**Authorize path**  
/oauth2/authorize

**Token path**  
/oauth2/token

**Supported grant types**

Implicit  
 Application  
 Access code  
 Resource owner - Password  
 Resource owner - JWT

**Supported client types**

Confidential  
 Public

13. Click Next.

14. In the Scopes section, enter details in the first NAME field. Enter Branch details in the first DESCRIPTION field.

### Scopes

Add scopes for this OAuth Provider **Add**

NAME	DESCRIPTION	DELETE
details	Branch details	

Back Cancel **Next**

- 15. Click Next.
- 16. Under Authentication, select the AuthURL Authentication method.

### Authentication

Authenticate application users using

AuthURL

---

### Authorization

Authorize application users using

Authenticated

Cancel Back **Next**

- 17. Do not alter the default Authorization setting (Any Authenticated). Click Next.

### Identity Extraction

Collect credentials using

Basic Authentication

---

### Authentication

Authenticate application users using

SampleAuthURL

---

### Authorization

Authorize application users using

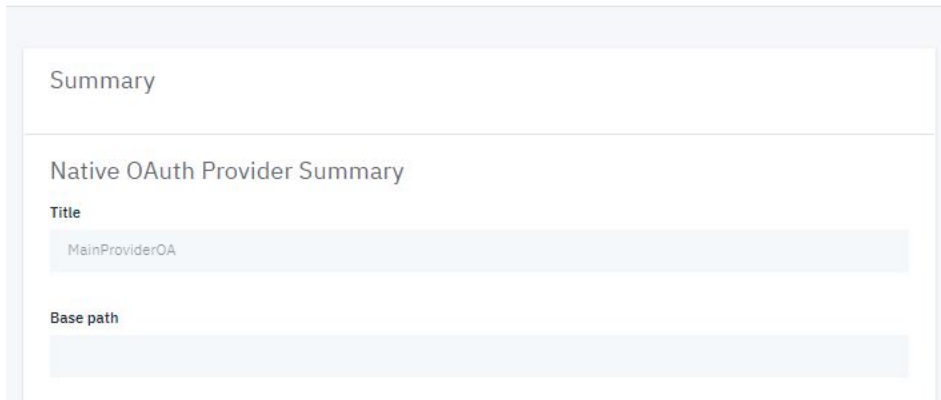
Authenticated

Cancel Back **Next**



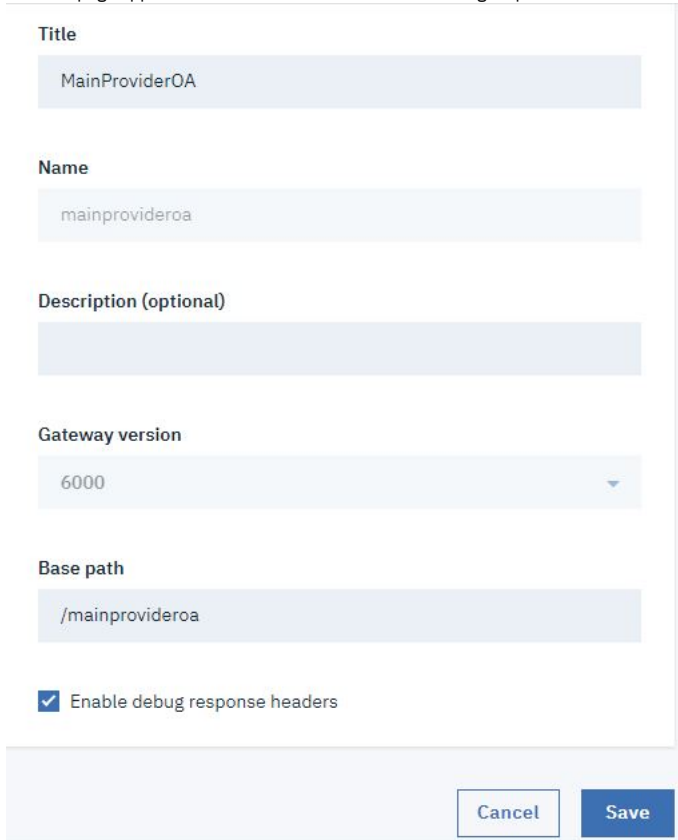
18. A Summary page appears. Scroll down and click Finish.

## Create Native OAuth Provider




The screenshot shows a 'Summary' page for creating a Native OAuth Provider. The page has a light blue header with the word 'Summary'. Below the header, the title 'Native OAuth Provider Summary' is displayed. There are two input fields: 'Title' with the value 'MainProviderOA' and 'Base path' which is currently empty.

19. The Info page appears. Scroll down and select Enable debug response headers. Click Save.

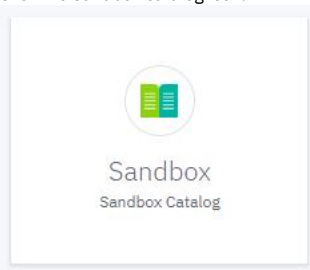



The screenshot shows an 'Info' page for creating a Native OAuth Provider. The page contains several fields: 'Title' (MainProviderOA), 'Name' (mainprovideroa), 'Description (optional)' (empty), 'Gateway version' (6000), and 'Base path' (/mainprovideroa). At the bottom, there is a checkbox labeled 'Enable debug response headers' which is checked. At the bottom right, there are two buttons: 'Cancel' and 'Save'.

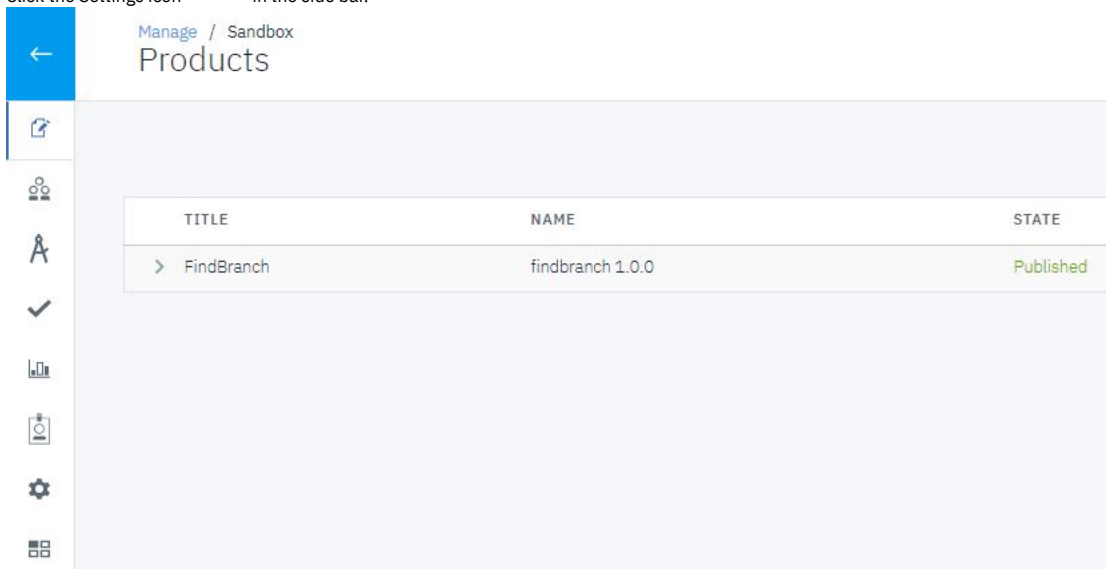
20. Click the Manage icon  in the side bar.



21. Click the Sandbox catalog icon.



22. Click the Settings icon  in the side bar.



23. Click API User Registries.

24. Click Edit.

Overview

Gateway Services

Lifecycle Approvals

Roles

Onboarding

**API User Registries**

OAuth Providers


API Endpoints

TLS Client Profiles

Portal

Properties

### API User Registries Edit

TITLE	TYPE	SUMMARY
 No items found		

- 25. Select AuthURL. Click Save.
- 26. Click OAuth Providers.

Overview

Gateway Services

Lifecycle Approvals

Roles

Onboarding

**API User Registries**

OAuth Providers

API Endpoints

TLS Client Profiles

Portal

Properties

### API User Registries

TITLE	TYPE
AuthURL	Authentication URL

- 27. Click Edit.

Overview

Gateway Services

Lifecycle Approvals

Roles

Role Defaults


Onboarding

API User Registries

**OAuth Providers**

### OAuth Providers Edit

Manage the OAuth Providers configured for API Manager

TITLE	TYPE
 No items found	

- 28. Select MainProviderOA. Click Save.
- 29. Click API Endpoints .

Manage / Sandbox  
Settings

- Overview
- Gateway Services
- Lifecycle Approvals
- Roles
- Role Defaults
- Onboarding
- API User Registries
- OAuth Providers**
- API Endpoints
- TLS Client Profiles
- Portal
- Properties

### OAuth Providers

Manage the OAuth Providers configured for API Manager

TITLE	TYPE
MainProviderOA	Native

30. Make a note of the gateway URL. You will need this to obtain an OAuth token.

## Settings

- Overview
- Gateway Services
- Lifecycle Approvals
- Roles
- Role Defaults
- Onboarding
- API User Registries
- OAuth Providers
- API Endpoints**
- TLS Client Profiles

### Vanity API Endpoint


Configure how vanity endpoints are displayed in the developer portal

- Use gateway URLs  
https://apimdev.ibm.com:9443/tutorial/sandbox


## Verify or create test application credentials

You will need the client ID and client secret for an application to test the OAuth functionality. You obtained this information during the completion of the [Tutorial: Creating a Client Application](#) tutorial listed in the Prerequisites.

Note: These steps are necessary only if you did not make note of the credentials for the client application you previously created.

1. Click the Applications icon  in the side bar.
2. Click the Down arrow icon alongside the AppOne application. The findbranch product should be listed.

TITLE	APPLICATION TYPE	CONSUMER ORGANIZATION	STATE						
AppOne	Production	sandbox-test-org	Enabled						
<table border="1"> <thead> <tr> <th>TITLE</th> <th>NAME</th> <th>PLAN</th> </tr> </thead> <tbody> <tr> <td>FindBranch auto product</td> <td>findbranch-auto-product:2.0.0</td> <td>Default Plan</td> </tr> </tbody> </table>				TITLE	NAME	PLAN	FindBranch auto product	findbranch-auto-product:2.0.0	Default Plan
TITLE	NAME	PLAN							
FindBranch auto product	findbranch-auto-product:2.0.0	Default Plan							
Test App Title	Production	sandbox-test-org	Enabled						

3. Click the Options icon  alongside AppOne. Select Credentials.

TITLE	APPLICATION TYPE	CONSUMER ORGANIZATION	STATE						
AppOne	Production	sandbox-test-org	Enabled						
<table border="1"> <thead> <tr> <th>TITLE</th> <th>NAME</th> <th>PLAN</th> </tr> </thead> <tbody> <tr> <td>FindBranch auto product</td> <td>findbranch-auto-product:2.0.0</td> <td>Default</td> </tr> </tbody> </table>				TITLE	NAME	PLAN	FindBranch auto product	findbranch-auto-product:2.0.0	Default
TITLE	NAME	PLAN							
FindBranch auto product	findbranch-auto-product:2.0.0	Default							
Test App Title	Production	sandbox-test-org							

- Edit
- Credentials**
- Create subscription
- Disable
- Delete

4. Click Add.

### Credentials

TITLE	CLIENT ID
Credential for AppOne	13ab76a200d62678567970070d829a9e

5. Copy both the Client ID and client Secret. You will need these values to obtain an OAuth token. Click Create.

## Credentials ✕

Save the client secret (it will no longer be retrievable for security purposes)

**Title**

cad72a92-f51c-4c52-bcc6-a0070ea19820

**Client ID**

a79c4a3b6602f9acdac59ca487a61e60 Copy

**Client Secret**

6efdf00a9ef9491237f59342f9085f5e Copy

Cancel
Create

6. Click the Back arrow until you see the Manage page.

←

[Manage](#) / [Sandbox](#)

## Credentials

Add

TITLE	CLIENT ID	
Credential for AppOne	13ab76a200d62678567970070d829a9e	⋮
fb1c10f4-d743-458a-89e5-ca1a363879d5	e6262cf13be9a67946219a6ca820abd2	⋮

7. Click Develop in the side bar.

## Manage

A catalog hosts a collection of API products that are visible in the associated develop

**Sandbox**

Sandbox Catalog

## Adding OAuth Security to an API



You add OAuth security enforcement to an existing API by creating a security definition for OAuth and then including that definition in the security applied to the API. Follow these steps.

1. Click the existing API to which you want to add OAuth security. This tutorial uses the FindBranch API.

Note: The gateway type used by this API must match the gateway type used by the OAuth Provider you just created. The gateway type for the API is shown at the foot of the API Setup page.

## Develop

APIs and Products Add ▾

TITLE	TYPE	LAST MODIFIED
 FindBranch-2.0.0	API (REST)	an hour ago
 FindBranch auto product-2.0.0	Product	2 hours ago

2. Click Security Definitions.
3. Click Add.

API Setup Security Definitions

Security Definitions Add

Security definitions control client access to API endpoints, including API key validation, application user authentication, and OAuth. Learn more


NAME	TYPE	LOCATED_IN
clientID	apiKey	header

API Setup

- Security Definitions
- Security
- Paths
- Definitions
- Properties
- Target Services

4. Enter `ProviderOA` in the Name field.
5. Under Type, click OAuth2.
6. Select `MainProviderOA` in the OAuth Provider field.
7. Select `Resource owner` in the Flow field.
8. Click Save.

Develop

findb... 2.0.0 ▾ Design\* Source Assemble ● Running ⚙️ ✔ No Errors Save ⋮ 

### API Security Definition

**Name**

ProviderOA

**Description (optional)**

**Type**

API Key  Basic  OAuth2

**OAuth Provider**

MainProviderOA ▾

**Flow**

Resource owner ▾

**Token URL**

https://\$(catalog.uri)/mainprovideroa/oauth2/token

9. Click Security.

API Setup

## Security Definitions

Security definitions control client access to API endpoints, including API key validation, application authentication, and OAuth. [Learn more](#)

NAME	TYPE	LOCATED_IN
ProviderOA	oauth2	
clientID	apiKey	header

Security Definitions

Security

Paths

Definitions

Properties

Target Services

Categories

10. Under Security Definitions, select ProviderOA, and select details.

Develop  
findb... 2.0.0

Design\* Source Assemble

Running No Errors Save

API Setup

## Security

Security definitions selected here apply across the API, but can be overridden for individual operations. [Add](#)

Learn more

SECURITY DEFINITIONS

<input checked="" type="checkbox"/> ProviderOA oauth2	<input checked="" type="checkbox"/> details
<input checked="" type="checkbox"/> clientID apiKey	

Security

Paths

Definitions

Properties

Target Services

Categories

Activity Log

11. Click Save.

## Test OAuth Security

Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test the new OAuth security added to the API, complete the following steps:

1. Click Assemble.

2. Click the Test icon .

Develop  
FindBranch 2.0.0

Design Source Assemble

Filter Search Show catches

Logic

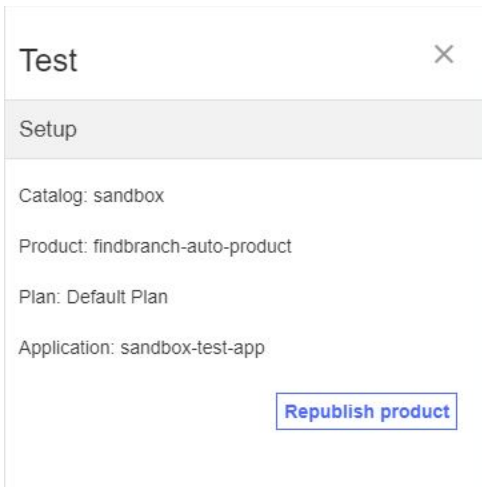
Switch


Throw

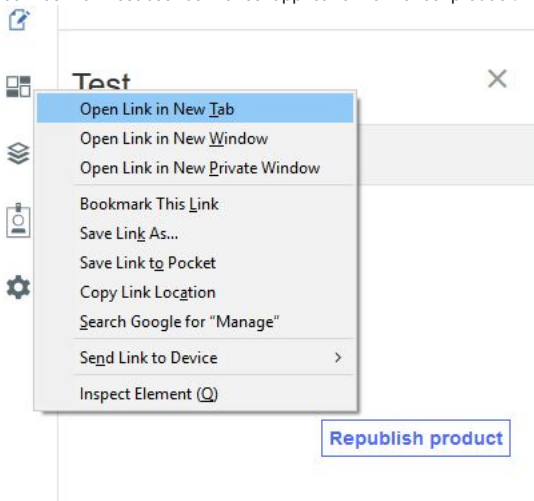
invoke

3. On the test Setup pane, click Republish product.



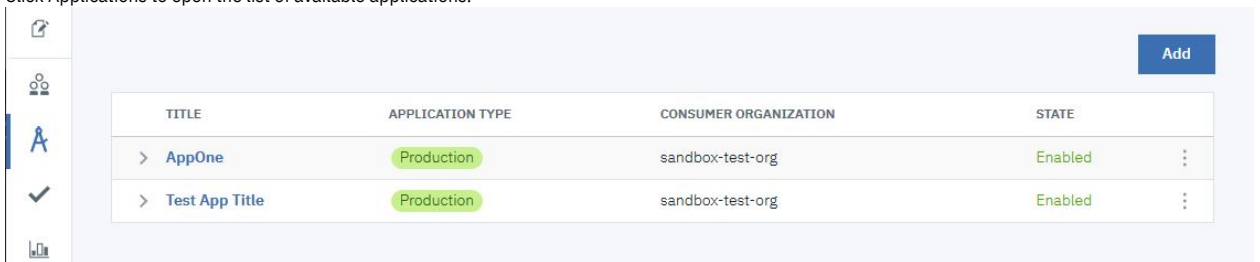


4. You must now resubscribe the test application to the test product. Right-click the Manage icon , and select Open Link in New Tab.



Take the following steps to navigate to the catalog applications page.

- a. Log in to the Manager page again.
- b. On the home page, click Manage.
- c. Click Sandbox to open the catalog.
- d. Click Applications to open the list of available applications.



e. Click the Options icon  alongside the AppOne application. Select Create Subscription .



f. Select FindBranch. Click Create .

### Product/Plan

Select the product and plan for the application subscription

- AccountService auto product:1.0.0/Default Plan
- Branches auto product:1.0.0/Default Plan
- FindBranch auto product:2.0.0/Default Plan

You must now return to testing the republished API. Close the currently open browser tab to return to testing the API you just republished.

5. Select `get /details` in the Operation field.

### Test

Operation

Choose an operation to invoke:

Operation

`get /details`

Identification

6. Enter the client ID you created previously in the `clientId` field.

7. Enter the client secret you created previously in the `clientSecret` field.

### Test

Choose an operation to invoke:

Operation

`get /details`

Identification

`clientId`

`a79c4a3b6602f9acdac59ca487a61e60`

`clientSecret`

.....

8. Enter `user` in the Username field. Enter `pass` in the Password field.

## Test ✕

Authorization

This operation is secured with password flow OAuth

Username

user

---

Password

....

---

explorer\_scopes

details

api\_security\_oauth\_token\_url:

https://\$(catalog.url)/oauth2/token

explorer\_authorize
explorer\_access\_token

9. Obtain an OAuth token. In this case, cURL is used to obtain the token using the following command.
- ```
curl -k https://gateway_url/org_name/sandbox/mainprovideroa/oauth2/token -d "grant_type=password&scope=details&username=user&password=pass&client_id=app_client_id&client_secret=app_client_secret"
```

```
C:\Users\David >curl -k https://apimdev.ibm.com:9443/tutorial/sandbox/mainprovideroa/oauth2/token -d "grant_type=password&scope=details&username=user&password=pass&client_id=35824f2646f799ef4c3f79e5d40df0ac&client_secret=57b2d3882aa6062b546a15ae704981f8"
{"token_type":"Bearer", "access_token":"AAIgMzU4MjRmMjY0NmY3OT11ZjRjM2Y3OWU1ZDQwZGYwYW0xkwNYTnIxWaHu8Htf10UAQUEG13TLJVHayXjPJE5Rxd7c1NdBEYRAEkuHIWX8hR2KF4AA9_SuOCNxJsETDaiJ", "expires_in":3600, "consented_on":1524503117, "scope":"details" }
C:\Users\David >
```

10. Enter or paste the access token in the explorer\_access\_token field. Here is an example token.

```
AAIgMzU4MjRmMjY0NmY3OT11ZjRjM2Y3OWU1ZDQwZGYwYW0xkwNYTnIxWaHu8Htf10UAQUEG13TLJVHayXjPJE5Rxd7c1NdBEYRAEkuHIWX8hR2KF4AA9_SuOCNxJsETDaiJ
```

# Test >

....

explorer\_scopes

details

api\_security\_oauth\_token\_url:

https://\$(catalog.url)/oauth2/token

explorer\_access\_token

.....

Repeat

Repeat the API invocation a set number of times, or until the stop button is clicked

Stop after:

10

Stop on error



11. Click Invoke. You may encounter a yellow error box with a URL embedded in it. Click this URL to override a browser certificate error.

### Response

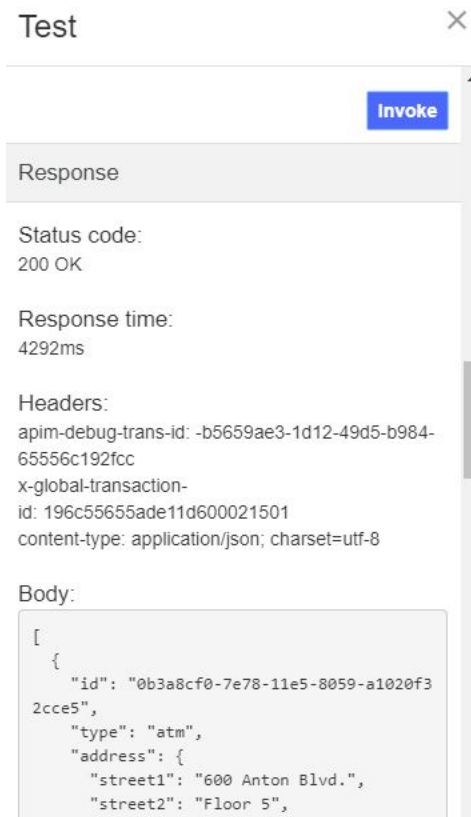
Status code:  
-1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.

[https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client\\_id=untrusted](https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client_id=untrusted)

12. Click Invoke again. The response contains branch data.



---

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created an OAuth Provider.
- Made the OAuth provider available within the catalog.
- Added OAuth security to an existing API.
- Tested the OAuth security.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Implementing OpenID Connect Security

This tutorial shows you how to add OpenID Connect capability to an existing native OAuth provider using API Manager.

---

### About this tutorial

In this tutorial you will complete the following lessons:

1. [Add OIDC capability to an OAuth native provider](#)
2. [Test OIDC Security](#)

---

### Before you begin

In this tutorial you will implement and test OpenID Connect security. To complete this tutorial, you must have the following available:

1. An existing published native OAuth provider. If you do not have an API available, complete the [Tutorial: Implementing OAuth Security](#) tutorial.
2. An external application, such as cURL, used to send requests to the OAuth token provider endpoint.


Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

---

### Add OIDC capability to an OAuth native provider

To add OIDC security to a native OAuth provider by using API Manager, complete the following steps:

1. Log in to API Manager.


2. Click the Resources icon  in the side bar.

IBM API Connect  
API Manager

Organization  
Tutorial


## Welcome to the API Manager

Choose an option to get started




### Develop APIs and Products

Edit, assemble, secure and test APIs.  
Package APIs using products for publishing to consumers.




### Manage Catalogs

Manage active APIs and consumers




### Manage Resources

Configure user registries, OAuth providers and TLS




### Manage Settings

Edit settings for roles, notifications and more.



### Learn more

Documentation and tutorials with step-by-step instructions



### Connect

Find expert answers in the API Connect community forum

3. Click OAuth Providers.

User Registries

TLS

OAuth Providers

Create

| TITLE                         | TYPE                |   |
|-------------------------------|---------------------|---|
| AuthURL                       | Authentication URL  | ⋮ |
| Sandbox Catalog User Registry | Local User Registry | ⋮ |

4. Click the name of an available OAuth provider, such as `MainProviderOA`.

5. The Info page appears. Click Introspection. Note that introspection is not required, but provides helpful debugging capabilities.

**Info**

- Configuration
- Scopes
- User Security
- Tokens
- Token Management
- Introspection
- Metadata
- OpenID Connect
- API Editor

## Native OAuth Provider

**Title**

MainProviderOA

**Name**

mainprovideroa

**Description (optional)**

**Gateway version**

6000

**Base path (optional)**

/mainprovideroa

Enable debug response headers

Cancel Save

6. Select Introspection.

**Info**

- Configuration
- Scopes
- User Security
- Tokens
- Token Management
- Introspection**
- Metadata
- OpenID Connect
- API Editor

## Introspection

When you enable token introspection, access token can be examined through the introspection path.

Introspection

**Introspection Path**

/oauth2/introspect

Cancel Save

7. Click Save.

8. Click OpenID Connect on the side bar.

9. Select Enable OIDC.

### OpenID Connect

Enable OpenID connect template to generate ID tokens.

Enable OIDC

**Support hybrid response types (optional)**

code id\_token

code token

code id\_token token

Auto Generate OIDC API Assembly

10. Select Auto Generate OIDC API Assembly.

### OpenID Connect

Enable OpenID connect template to generate ID tokens.

Enable OIDC

**Support hybrid response types (optional)**

code id\_token

code token

code id\_token token

Auto Generate OIDC API Assembly

**ID token issuer**

IBM APIConnect

**ID token signing key**

**ID token signing key identifier (optional)**

**Sample ID token signing key (RS256)**

```
{ "kty": "RSA", "d": "QcdNmaSVuRjY9G4QKxNhtTVcAunf99-Z9J0kji5JebIZNG_BFmAFeV1pAUtYI
```

**Sample ID token validation key (RS256)**

```
{ "kty": "RSA", "e": "AQAB", "use": "sig", "kid": "APIConnect", "alg": "RS256", "n": "jg8lWlVlUldMah4q-fsi
```

**ID token signing algorithm**

RS256

11. Enter a JWK in the ID token signing key field. Here is an example. { "alg": "HS256", "kty": "oct", "use": "sig", "k": "o5yErLaE-dbgVpSw65Rq570A9dHyaF66Q\_Et5azPa-XUjbyP0w9iRWhR4kr09aFfQLXeIODIN4uhjE1YKXt8n76jt0Pjkd2pqk4t9abRF6tnL19GV4pf1fL6uvVKkP4weOh39tqHt4TmkBgF2P-gFhgssZpjwq6182fz3dUhQ2nkzoLA\_CnyDGLZLd7SZ1yv73uzfE20t813zmig8KTMEMWvcWSDvy61F06vs\_6LURcq\_IEEevUiubBxG5S2akNnWigfphWYjMI5M22FOCpdcDBt4L7K1-yHt95Siz0Qub0MN1T\_X8F76wH7\_A37GpKKJGqeaINWmHkgWdE8QWDQ", "kid": "hs256-key" }

12. Select HS256 in the ID token signing algorithm field.



### OpenID Connect

Enable OpenID connect template to generate ID tokens.

Enable OIDC

**Support hybrid response types (optional)**

code id\_token  
 code token  
 code id\_token token

Auto Generate OIDC API Assembly

**ID token issuer**

IBM APIConnect

**ID token signing key**

{t95Siz0QUb0MNIIT\_X8F76wH7\_A37GpKKJGqeiNWmHkgWdE8QWDQ", "kid": "hs256-key" }

**ID token signing key identifier (optional)**

**Sample ID token signing key (RS256)**

{ "kty": "RSA", "d": "QcdNmaSVuRJY9G4QKxNhtTVcAunf99-Z9JOkji5JebIZNG\_BFmAFeV1pAUTYI

**Sample ID token validation key (RS256)**


{ "kty": "RSA", "e": "AQAB", "use": "sig", "kid": "APIConnect", "alg": "RS256", "n": "jg8IWivUldMah4q-fs

**ID token signing algorithm**

HS256

Cancel Save

13. Click Save.

14. Click the Manage icon  in the side bar.

← Edit Native OAuth Provider

- Info
- Configuration
- Scopes
- User Security
- Tokens
- Token Management
- Introspection
- Metadata
- OpenID Connect
- API Editor

**Native OAuth Provider**

**Title**

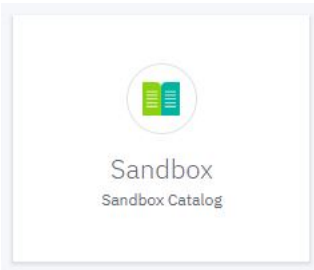
MainProviderOA


**Name**

mainprovideroa

**Description (optional)**

15. Click the Sandbox catalog icon.



16. Click the Settings icon  in the side bar.

The screenshot shows the "Products" page in the API Manager console. The breadcrumb navigation is "Manage / Sandbox". The page title is "Products". On the left is a vertical sidebar with icons for home, products, search, checkmark, bar chart, API endpoints, settings (highlighted), and a grid icon. The main content area contains a table with the following data:

| TITLE        | NAME             | STATE     |
|--------------|------------------|-----------|
| > FindBranch | findbranch 1.0.0 | Published |

17. Click API Endpoints .

The screenshot shows the "Settings" page in the API Manager console. The breadcrumb navigation is "Manage / Sandbox". The page title is "Settings". On the left is a vertical sidebar with icons for home, products, search, checkmark, bar chart, API endpoints, settings (highlighted), and a grid icon. The main content area is titled "OAuth Providers" and includes the subtitle "Manage the OAuth Providers configured for API Manager". Below this is a table with the following data:

| TITLE          | TYPE   |
|----------------|--------|
| MainProviderOA | Native |

On the left side of the main content area, there is a list of settings categories: Overview, Gateway Services, Lifecycle Approvals, Roles, Role Defaults, Onboarding, API User Registries, **OAuth Providers** (highlighted), API Endpoints, TLS Client Profiles, Portal, and Properties.

18. Make a note of the gateway URL. You will need this to obtain an OAuth token.



**Credentials** **Add**

| TITLE                 | CLIENT ID                        |
|-----------------------|----------------------------------|
| Credential for AppOne | 13ab76a200d62678567970070d829a9e |

5. Copy both the Client ID and Client Secret (you need these values to obtain an OAuth token). Click Create to add the new credentials to the AppOne application.

**Credentials** ✕

Save the client secret (it will no longer be retrievable for security purposes)

**Title**

cad72a92-f51c-4c52-bcc6-a0070ea19820

**Client ID**

a79c4a3b6602f9acdac59ca487a61e60 **Copy**

**Client Secret**

6efd00a9ef9491237f59342f9085f5e **Copy**

Cancel **Create**

6. Click the Back arrow until you see the Manage page.

← Manage / Sandbox

**Credentials** **Add**

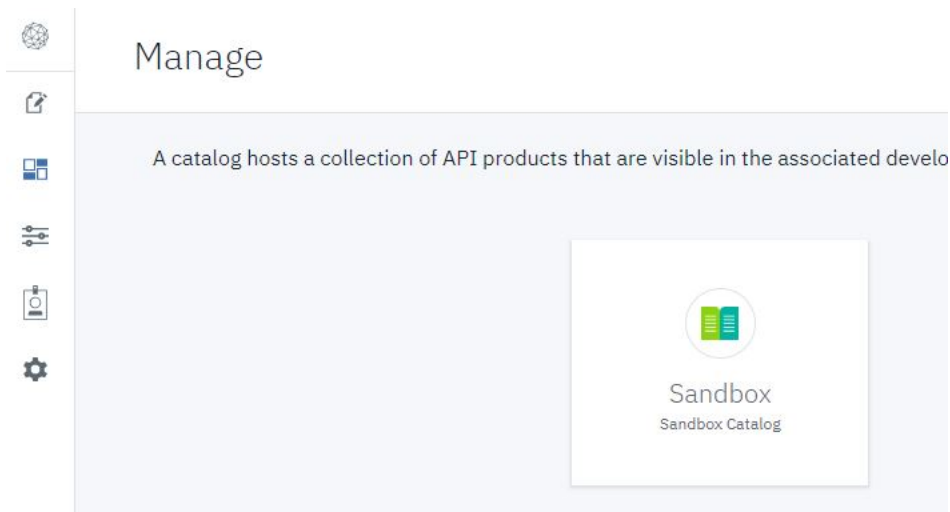
| TITLE                                | CLIENT ID                        |
|--------------------------------------|----------------------------------|
| Credential for AppOne                | 13ab76a200d62678567970070d829a9e |
| fb1c10f4-d743-458a-89e5-ca1a363879d5 | e6262cf13be9a67946219a6ca820abd2 |

## Test OIDC Security

Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

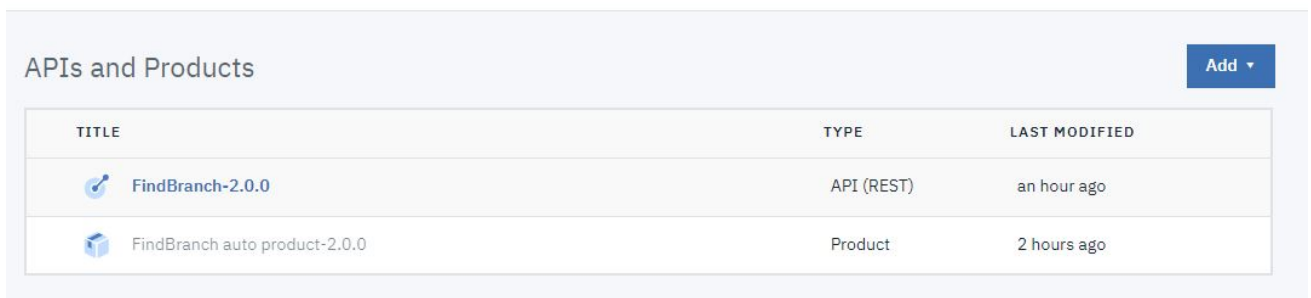
To test the new OIDC security added to the API, complete the following steps:

1. Click Develop  in the side bar.



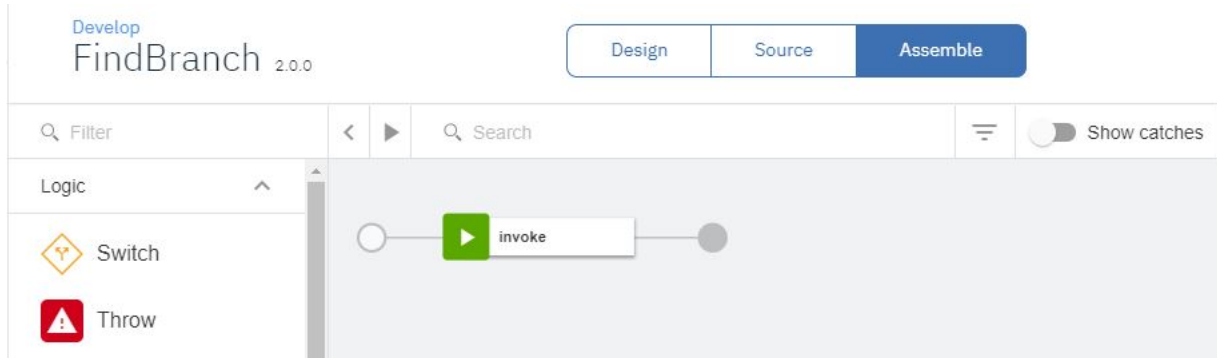
2. On the Develop page, click the name of the API that uses the OAuth provider to which you added OIDC. This tutorial uses the FindBranch API.

## Develop

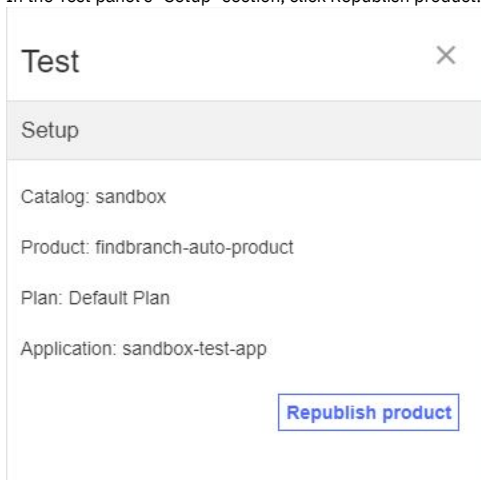


3. Click Assemble in the page header to open the Test panel.

4. Click the Test icon .

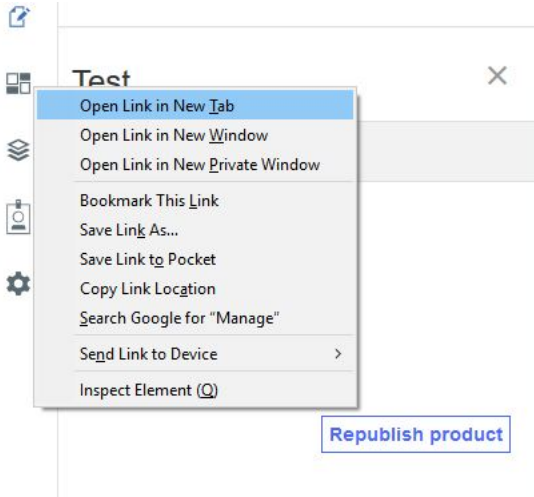


5. In the Test panel's "Setup" section, click Republish product.



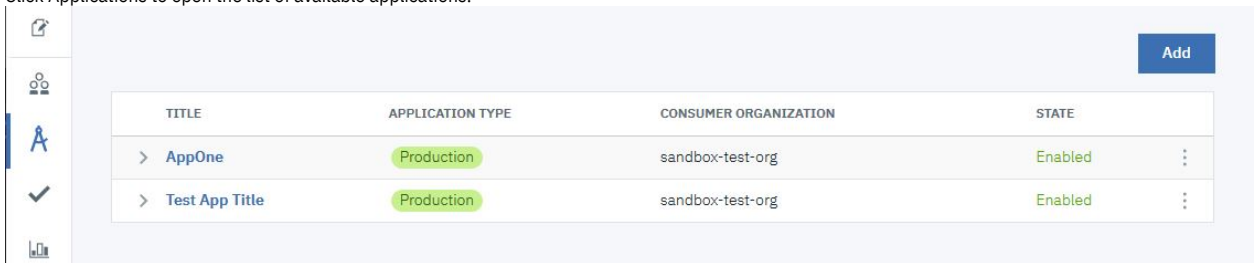
You must now resubscribe the test application to the updated product.

6. Open a copy of the API Manager in a new browser tab.  
For example, in Windows, right-click API Manager in the page header, and select Open Link in New Tab.

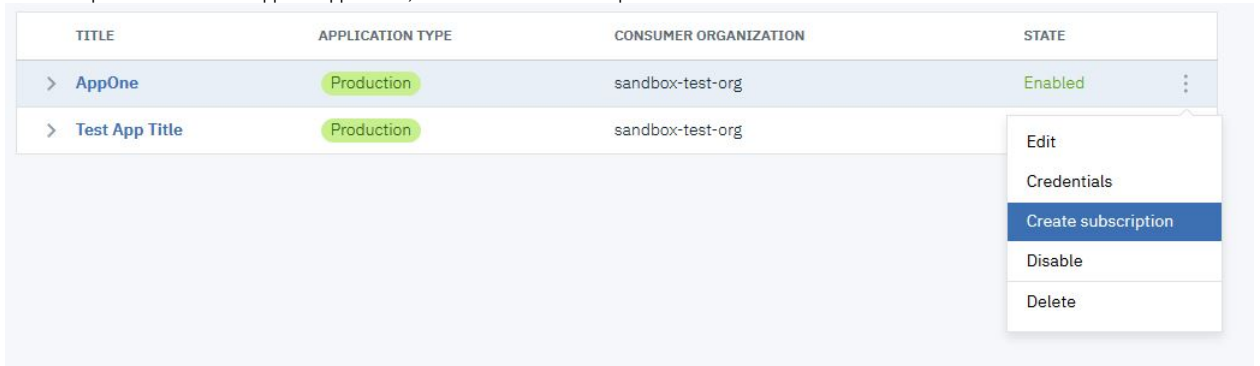


Complete the following steps to subscribe the client app to the updated API:

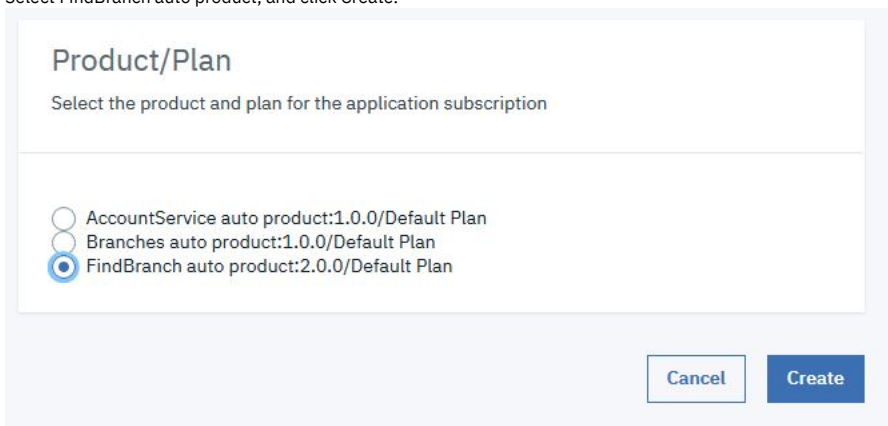
- a. In the newly opened browser tab, log in to API Manager.
- b. On the Home page, click Manage in the navigation list.
- c. Click the Sandbox catalog.
- d. Click Applications to open the list of available applications.



- e. Click the Options icon (three dots) for the AppOne application, and select Create subscription.

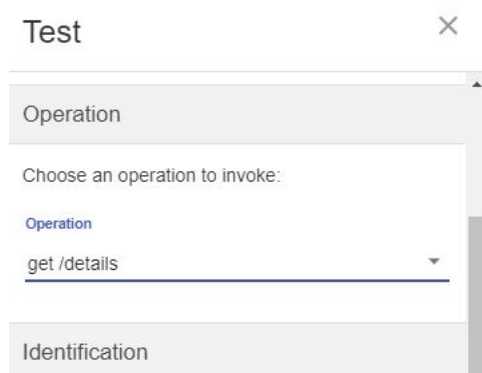


- f. Select FindBranch auto product, and click Create.



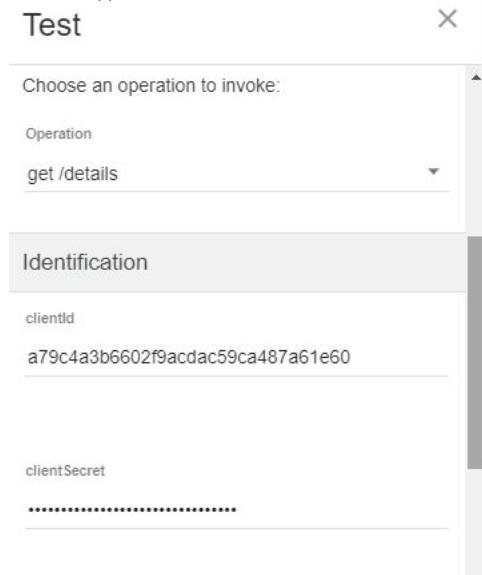
Return to the browser tab where you opened the Test panel.

7. Select `get /details` in the Operation field.



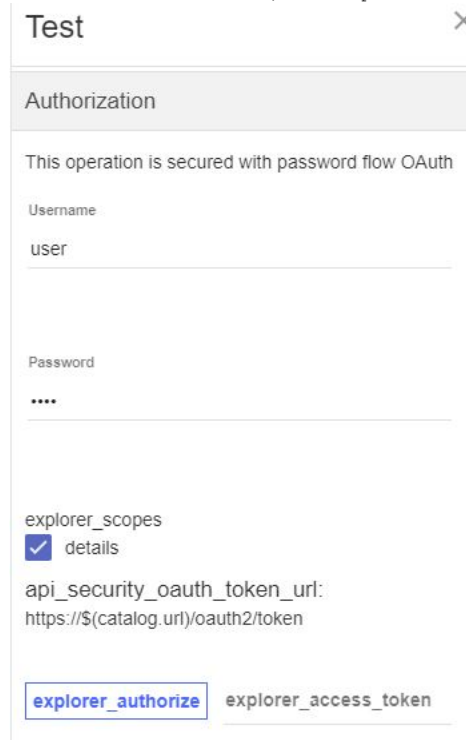
The screenshot shows a 'Test' panel with a close button (X) in the top right. Below the title bar, there is a section titled 'Operation'. Underneath, it says 'Choose an operation to invoke:'. There is a dropdown menu labeled 'Operation' with 'get /details' selected and highlighted. Below the dropdown is a section titled 'Identification'.

8. Paste the application's client ID in the `clientId` field.
9. Paste the application's client secret in the `clientSecret` field.



The screenshot shows the 'Test' panel with the 'Operation' dropdown still set to 'get /details'. The 'Identification' section is expanded, showing two input fields. The 'clientId' field contains the value 'a79c4a3b6602f9acdac59ca487a61e60'. The 'clientSecret' field contains a series of dots representing a masked password.

10. Enter `user` in the Username field, and enter `pass` in the Password field.



The screenshot shows the 'Test' panel with the 'Authorization' section expanded. It contains the text 'This operation is secured with password flow OAuth'. Below this, there are two input fields: 'Username' with the value 'user' and 'Password' with a series of dots. At the bottom, there is a section for 'explorer\_scopes' with a checked checkbox for 'details'. Below that, the 'api\_security\_oauth\_token\_url' is shown as 'https://\$(catalog.url)/oauth2/token'. At the very bottom, there is a button labeled 'explorer\_authorize' and an input field labeled 'explorer\_access\_token'.

- Obtain an OAuth token. For example, run the following cURL command in a command window to obtain the token:

```
curl -k https://gateway_url/org_name/sandbox/mainprovider/oa/oauth2/token -d
"grant_type=password&scope=details
openid&username=user&password=pass&client_id=app_client_id&client_secret=app_client_secret"
```

```
C:\Users\David >curl -k https://apimdev.ibm.com:9443/tutorial/sandbox/mainprovider/oa/oauth2/token -d
"grant_type=password&scope=details&username=user&password=pass&client_id=35824f2646f799ef4c3f79e5d40df0ac&client_secret=57b2d3882aa6062b546a15ae704981f8"
{"token_type":"Bearer", "access_token":"AAIgMzU4MjRmMjY0NmY3OT1lZjRjM2Y3OWU1ZDQwZGYwYW0xkwNYTnIxWwHu8Htf10UAQUEG13TLJVHayXjPJE5Rxd7c1NdBEYRAEKuHIWx8r2KF4AA9_Su0CnxJsETDaij", "expires_in":3600, "consented_on":1524503117, "scope":"details" }
C:\Users\David >
```

- Introspect the token. In this case, cURL is used to obtain the data returned from the introspection endpoint. Use a command similar to the following command.

```
curl -k
https://gateway_url/org_name/sandbox/mainprovider/oa/oauth2/introspect -d
"username=user&password=pass&client_id=app_client_id&client_secret=app_client_secret&token=Atokenstring"
```

```
C:\Users\David >curl -k https://apimdev.ibm.com:9443/tutorial/sandbox/mainprovider/oa/oauth2/introspect -d
"client_id=13ab76a200d62678567970070d829a9e&client_secret=59f7587cf05778f0c1229b5c45fd70c5&token=AAIgMTNhYjc2YTlWmgQ2MjY3ODU2Nzk3MDA3MGQ4Mj1hOWVb6S-C-uB9MEh0fTFeBHneCvKiDtIpFjggqIbQ2jx64CVhWbgaOQdCS16IpskLfIjFqOn8CGheVM9vFWVDI fHLv4u0HhVhGdCNu7xFWgkQ13w"
{"active":true,"scope":"details openid","client_id":"13ab76a200d62678567970070d829a9e","username":"user","token_type":"Bearer", "grant_type":"password", "ttl":1139, "exp":1550606219, "expstr":"2019-02-19T19:56:59Z", "iat":1550602619, "nbf":1550602619, "nbfstr":"2019-02-19T18:56:59Z", "consented_on":1550602619, "consented_on_str":"2019-02-19T18:56:59Z", "one_time_use":false}
C:\Users\DavidShute>
```

- Enter or paste the access token in the explorer\_access\_token field, and then click explorer\_authorize.

Here is an example token: **AAIgMTNhYjc2YTlWmgQ2MjY3ODU2Nzk3MDA3MGQ4Mj1hOWVb6S-C-uB9MEh0fTFeBHneCvKiDtIpFjggqIbQ2jx64CVhWbgaOQdCS16IpskLfIjFqOn8CGheVM9vFWVDI fHLv4u0HhVhGdCNu7xFWgkQ13w**

### Test >

---

....

explorer\_scopes  
 details

api\_security\_oauth\_token\_url:  
 https://\$(catalog.url)/oauth2/token

explorer\_access\_token

Repeat

Repeat the API invocation a set number of times, or until the stop button is clicked

Stop after: 10  Stop on error

- Click Invoke to execute the API call.

You might see a yellow error box with a URL embedded in it. Click this URL to override the indicated browser certificate error.



Response

Status code:  
-1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.

[https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client\\_id=untrusted](https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client_id=untrusted)

15. Click Invoke again. The response contains branch data and the HTTP status 200 OK.

Test ×

[Invoke](#)

Response

Status code:  
200 OK

Response time:  
4292ms

Headers:  
apim-debug-trans-id: -b5659ae3-1d12-49d5-b984-65556c192fcc  
x-global-transaction-id: 196c55655ade11d600021501  
content-type: application/json; charset=utf-8

Body:

```
[
  {
    "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
    "type": "atm",
    "address": {
      "street1": "600 Anton Blvd.",
      "street2": "Floor 5",
```

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Added OpenID Connect security to an existing API.
- Tested the security.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Generate a JSON Web Token (JWT)

This tutorial shows you how to define and implement a REST API definition that generates a JSON Web Token (JWT).

### About this tutorial

In this tutorial you will complete the following lessons:

1. [Generate a JWT](#)
2. [Testing the REST API](#)

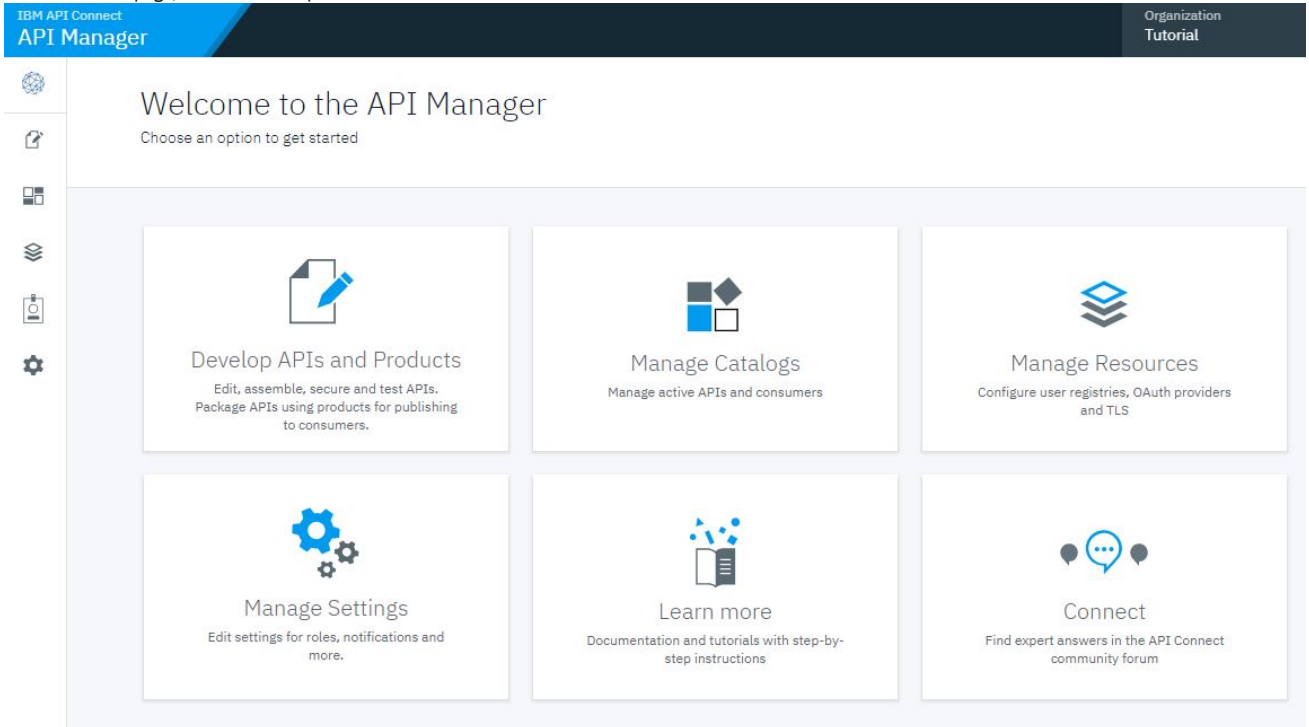
Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

## Generate a JWT

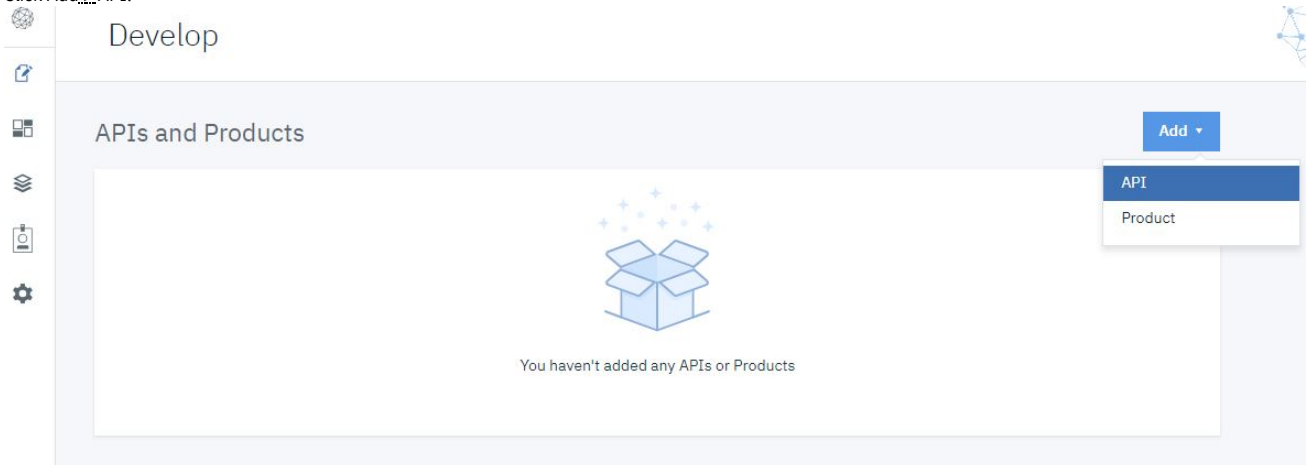
Create a REST API to generate and return a JSON Web Token (JWT).

To add and define this REST API, complete the following steps:

1. Log in to API Manager.
2. In the Welcome page, click the Develop APIs and Products tile.



3. Click Add > API.



4. Select New OpenAPI. Click Next.

**Create**

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations

**Import**

- Existing OpenAPI**  
Use an existing definition of a REST proxy

5. Enter the appropriate information to create a REST API definition.
  - a. In the Title field, enter `JWT`.
  - b. The Name and Base Path fields autopopulate with the terms `jwt` and `/jwt` respectively.
  - c. Enter `1.0.0` in the Version field.

**Info**  
Enter details of this API

**Title**  
JWT

**Name**  
jwt

**Version**  
1.0.0

**Base path (optional)**  
/jwt

**Description (optional)**

6. Click Next.
7. Make no changes on the Secure screen. Click Next.

## Secure

Configure the security of this API

- Secure using Client ID
- CORS

8. You see the progress as the new API gets created. When it is done, you see a Summary. Click Edit API.

## Summary

- Generated OpenAPI 2.0 definition
- Applied security

9. In the side bar of the Design page, select Paths to display the Paths panel.

Develop  
JWT 1.0.0

- API Setup
- Security Definitions
- Security
- Paths
- Definitions
- Properties
- Target Services
- Categories
- Activity Log

### Info

Enter the API summary details

---

**Title**

JWT

**Name**

jwt

**Version**

1.0.0

10. Click Add.

11. In the Path name field, enter /gen.

12. In the Operations section, click Add.

13. Select GET and click Add.

**Path**  
Paths identify the REST resources exposed by the API. An operation combines a path with an HTTP verb, parameters, and definitions for requests and responses. [Learn more](#)

**Path name**  
/gen

**Path Parameters** Add

| REQUIRED | NAME | LOCATED_IN | TYPE | DESCRIPTION | DELETE |
|----------|------|------------|------|-------------|--------|
|          |      |            |      |             |        |

**Operations** Add

| NAME |
|------|
| GET  |

Cancel Save

14. Click Save.
15. Click /gen in the list of available paths.

**Paths**

| NAME |
|------|
| /gen |

16. Click GET in the list of Operations.
17. Scroll down. In the Parameters section, click Add.
  - a. Select REQUIRED.
  - b. Enter `iss-claim` in the NAME field.
  - c. Select `header` in the LOCATED IN field.
  - d. Select `string` in the TYPE field.
  - e. Enter `Enter https://myidp.ibm.com to match` in the DESCRIPTION field.

**Parameters** Add

| REQUIRED                            | NAME      | LOCATED_IN | TYPE   | DESCRIPTION                          | DELETE |
|-------------------------------------|-----------|------------|--------|--------------------------------------|--------|
| <input checked="" type="checkbox"/> | iss-claim | header     | string | Enter https://myidp.ibm.com to match |        |

18. Click Add to add a second parameter.
  - a. Select REQUIRED.
  - b. Enter `aud-claim` in the NAME field.
  - c. Select `header` in the LOCATED IN field.
  - d. Select `string` in the TYPE field.
  - e. Enter `Enter ClientID1 to match` in the DESCRIPTION field.

**Parameters** Add

| REQUIRED                            | NAME      | LOCATED_IN | TYPE   | DESCRIPTION                          | DELETE |
|-------------------------------------|-----------|------------|--------|--------------------------------------|--------|
| <input checked="" type="checkbox"/> | iss-claim | header     | string | Enter https://myidp.ibm.com to match |        |
| <input checked="" type="checkbox"/> | aud-claim | header     | string | Enter ClientID1 to match             |        |

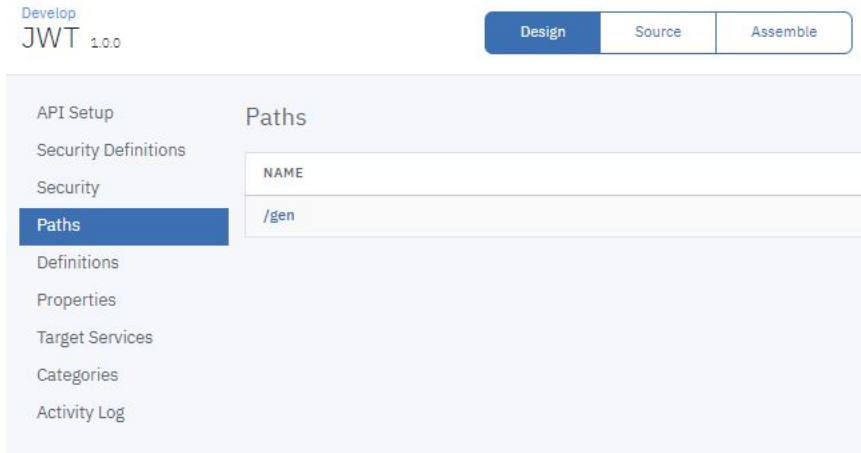
19. In the Response section, change the description of the pre-supplied 200 status code to 200 OK.

**Response** Add

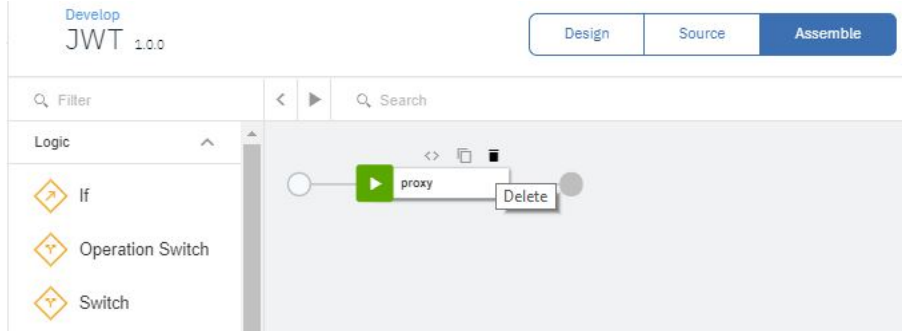
| STATUS CODE | SCHEMA | DESCRIPTION | DELETE |
|-------------|--------|-------------|--------|
| 200         | string | 200 OK      |        |

Cancel Save

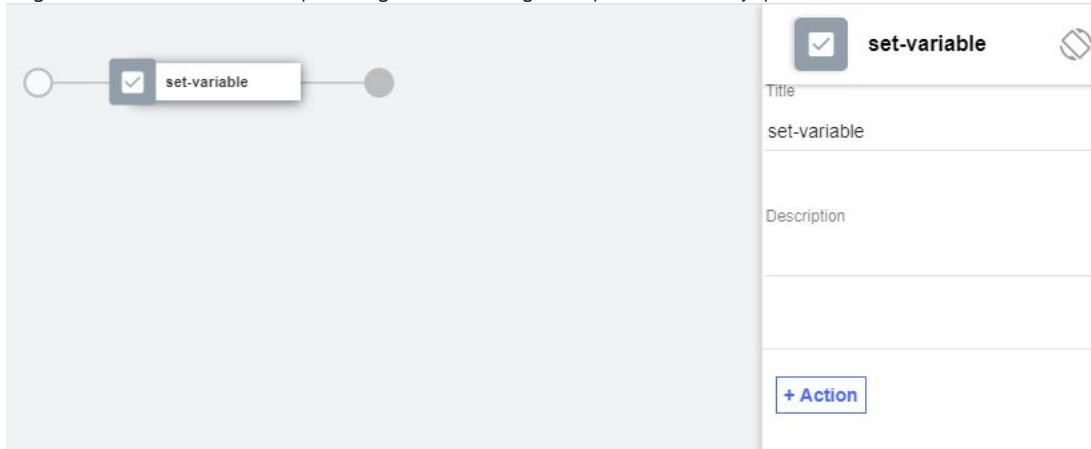
20. Click Save.
21. Click Assemble.



22. Hover the mouse over the existing Proxy or Invoke action and click the trash can icon to delete it.



23. Drag the Set Variable action onto the processing flow line. A configuration panel automatically opens.



24. Click + Action field.
25. Enter `hs256-key` in the Set field.
26. Select `string` in the Type field.
27. Enter a JWK in the Value field. Here is an example. 

```
{ "alg": "HS256", "kty": "oct", "use": "sig", "k": "o5yErLaE-dbgVpSw65Rq570A9dHyaF66Q_Et5azPa-XUjbyP0w9iRWhR4kru09aFfQLXeIODIN4uhjE1YKXt8n76jt0Pjkd2pqk4t9abRF6tnL19GV4pflfL6uvVKkP4weOh39tqHt4TmkBgF2P-gFhgssZpjwq6182fz3dUhQ2nkzoLA_CnyDGLZLd7SZ1yv73uzfE20t813zmig8KTMEMWvcWSDvy61F06vs_6LURcq_IEEEvUiubBxG5S2akNnWigfphWYjMI5M22FOCpdcDBt4L7K1-yHt95Siz0Qub0MN1T_X8F76wh7_A37GpKKJGqaiNWmHkgWdE8QWDQ", "kid": "hs256-key" }
```

Action \*

Set

Set, Add, or Clear a runtime variable.

Set

hs256-key

The name of the variable to be set.

Type \*

string

The type of the value to set. This can be string, number or boolean.

Value

```
{ "alg": "HS256", "kty": "oct", "use": "sig", "k": "o5yErLa
```

The value that the variable will be set to.

28. Close the property panel. Click Save.

29. Drag the Generate JWT action onto the processing flow line after the set-variable icon. A configuration panel automatically opens.

The screenshot shows a processing flow line with two actions: 'set-variable' (checked) and 'jwt-generate'. To the right, a configuration panel for 'jwt-generate' is open, showing fields for Title and Description.

30. Enter `request.headers.iss-claim` in the Issuer Claim field.

31. Enter `request.headers.aud-claim` in the Audience Claim field.

Title

jwt-generate

Description

JSON Web Token (JWT)

generated.jwt

Runtime variable in which to place the generated JWT. If not set, the JWT is placed in the Authorization Header as a Bearer token.

JWT ID Claim

Indicates whether a JWT ID (jti) claim should be added to the JWT. If selected, the jti claim value will be a UUID.

Issuer Claim

request.headers.iss-claim

Runtime variable from which the Issuer (iss) claim string can be retrieved. This claim represents the Principal that issued the JWT.

Subject Claim

Runtime variable from which the Subject (sub) claim string can be retrieved.

Audience Claim

request.headers.aud-claim

32. Enter `hs256-key` in the Sign JWK variable name field.

33. Select HS256 in the Cryptographic Algorithm field.

**request.headers.aud-claim**

Runtime variable from which the Audience (aud) claim string can be retrieved. Multiple variables are set via a comma-separated string.

Validity Period

3600

The length of time (in seconds), that is added to the current date and time, in which the JWT is considered valid.

Private Claims

Runtime variable from which a valid set of JSON claims can be retrieved.

Sign JWK variable name

hs256-key

Runtime variable containing the JWK to use to sign the JWT.

Cryptographic Algorithm\*

HS256

Select a cryptographic algorithm.

Sign Crypto Object

34. Close the property panel. Click Save.
35. Drag the GatewayScript action onto the processing flow line after the Generate JWT icon. A configuration panel automatically opens.
36. Enter the following code:

```
var apim = require('apim');
apim.setvariable('message.body', apim.getvariable('generated.jwt'));
```

The screenshot shows the API Gateway configuration interface. The main flow consists of three actions: 'set-variable', 'jwt-generate', and 'gatewayscript'. A configuration panel for the 'gatewayscript' action is open on the right, displaying the code:


```
1 var apim = require('apim');
2 apim.setvariable('message.body', apim
  .getvariable('generated.jwt'));
```

37. Close the property panel. Click Save.

## Testing the REST API

Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

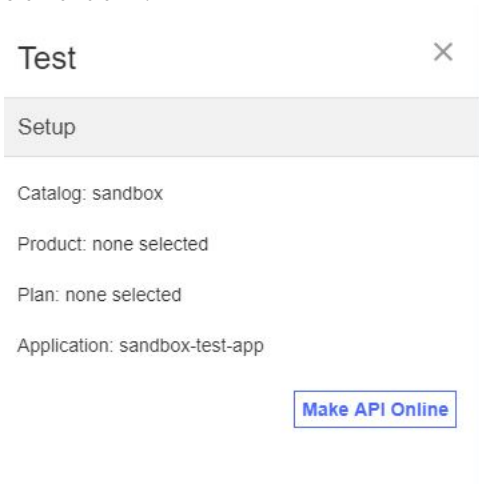
To test the REST API, complete the following steps:

1. Click the Test icon .





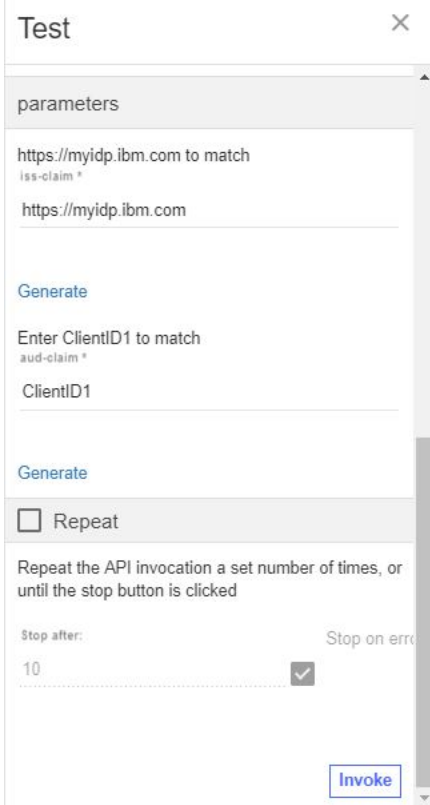
2. Click Activate API.



3. Select the get /gen Operation.

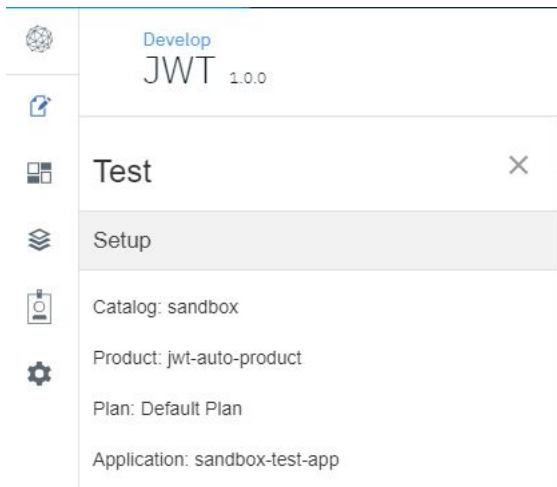
4. Enter `https://myidp.ibm.com` in the iss-claim field.

5. Enter `ClientID1` in the aud-claim field.

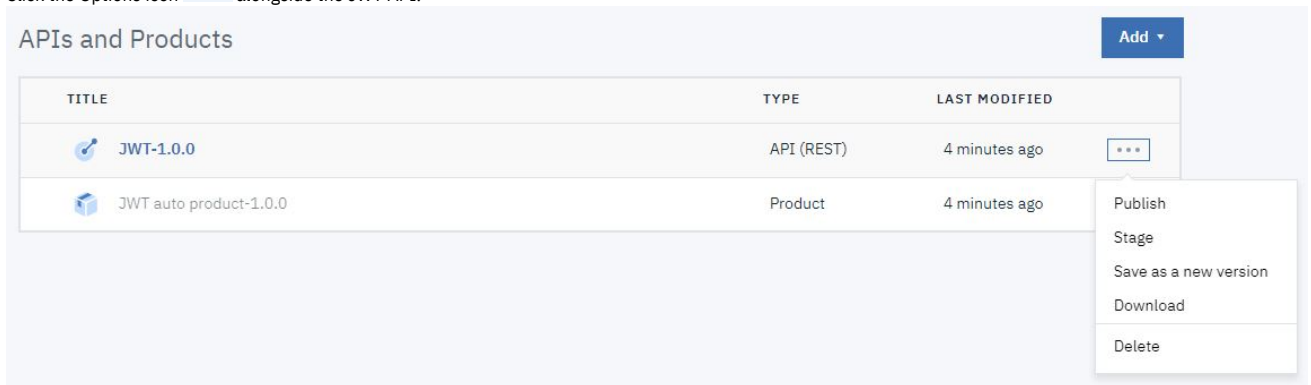


6. Click Invoke. You may encounter a yellow error box with a URL embedded in it. Click this URL to override a browser certificate error.





2. Click the Options icon  alongside the JWT API.



3. Select Download.

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a new API definition that generates a JSON Web Token (JWT).
- Tested the new API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Validate a JSON Web Token (JWT)

This tutorial shows you how to define and implement a REST API definition that validates a JSON Web Token (JWT).

### About this tutorial

In this tutorial you will complete the following lessons:

1. [Validate a JWT](#)
2. [Testing the REST API](#)

Note: The Sandbox catalog must be configured to use either a DataPower® Gateway (v5 compatible) or a DataPower API Gateway or both. See [Creating and configuring Catalogs](#).

### Before You Begin

You must also do the following steps.

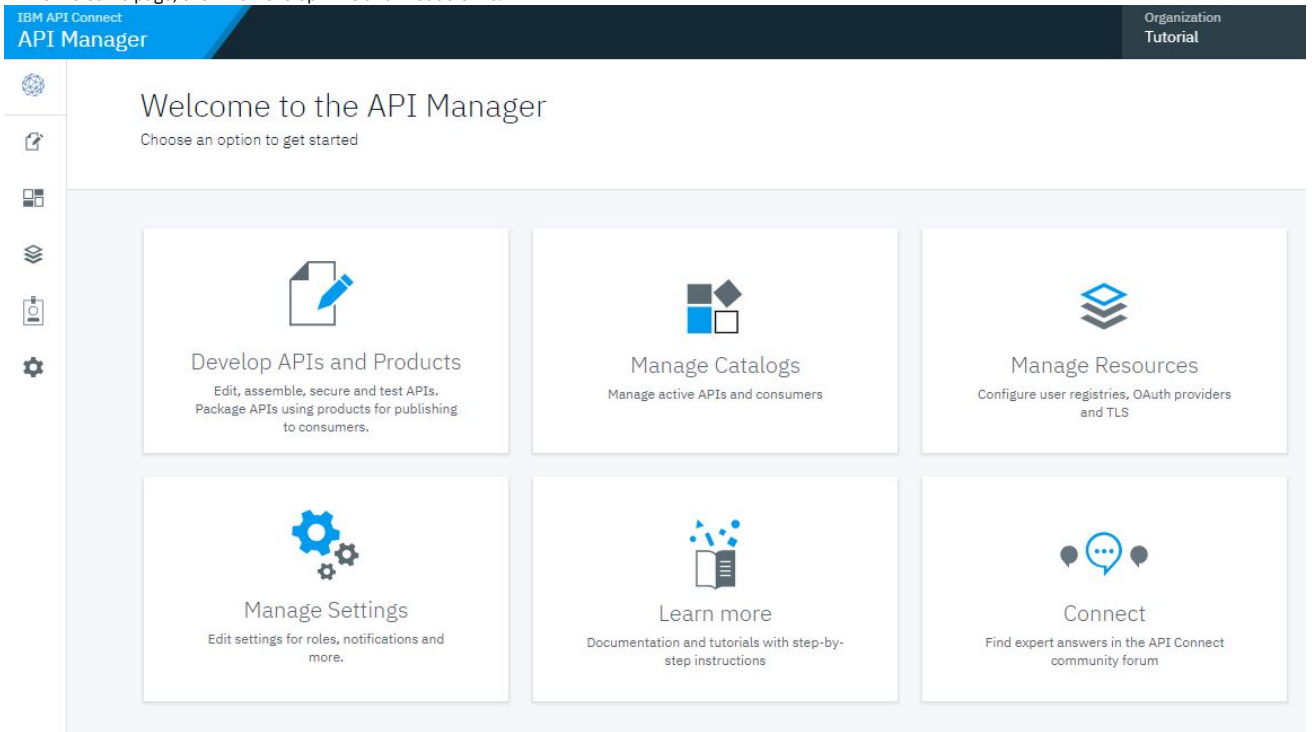
- Complete the [Tutorial: Generate a JSON Web Token \(JWT\)](#) tutorial. This tutorial generates a JSON Web Token that can be validated by this tutorial. You will need this JWT to test this validation API.

## Validate a JWT

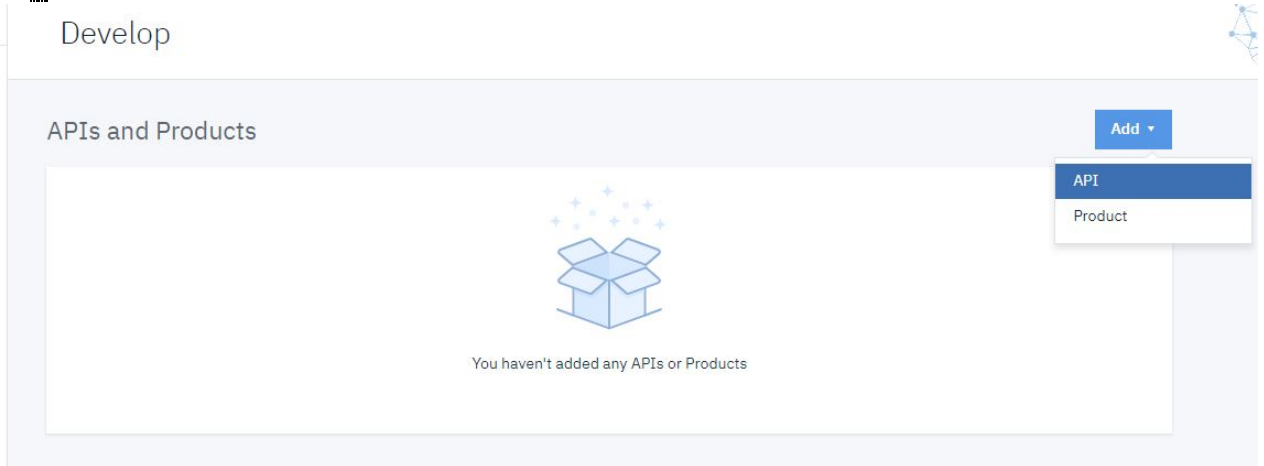
Create a REST API to validate a JSON Web Token (JWT).

To add and define this REST API, complete the following steps:

1. Log in to API Manager.
2. In the Welcome page, click the Develop APIs and Products tile.



3. Click Add > API.



4. Select New OpenAPI. Click Next.

**Create**

- From target service**  
Create a REST proxy that routes all traffic to a target API or service endpoint
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations

**Import**

- Existing OpenAPI**  
Use an existing definition of a REST proxy

5. Enter the appropriate information to create a REST API definition.
  - a. In the Title field, enter `JWTVAL`.
  - b. The Name and Base Path fields autopopulate with the terms `jwtval` and `/jwtval` respectively.
  - c. Enter `1.0.0` in the Version field.

**Info**  
Enter details of this API

**Title**  
JWTVAL

**Name**  
jwtval

**Version**  
1.0.0

**Base path (optional)**  
/jwtval

**Description (optional)**

6. Click Next.
7. Make no changes on the Secure screen. Click Next.

## Secure

Configure the security of this API

- Secure using Client ID
- CORS

8. You see the progress as the new API gets created. When it is done, you see a Summary. Click Edit API.

## Summary

- Generated OpenAPI 2.0 definition
- Applied security

9. In the side bar of the Design page, select Paths to display the Paths panel.

Develop

# JWTVAL 1.0.0

- API Setup
- Security Definitions
- Security
- Paths
- Definitions
- Properties
- Target Services
- Categories
- Activity Log

## Info

Enter the API summary details

---

**Title**

**Name**

**Version**

10. Click Add.

11. In the Path name field, enter /val.

12. In the Operations section, click Add.

13. Select GET and click Add.

### Path

Paths identify the REST resources exposed by the API. An operation combines a path with an HTTP verb, parameters, and definit more

---

**Path name**

/val

---

### Path Parameters

| REQUIRED | NAME | LOCATED_IN | TYPE | DESCRIPTION |
|----------|------|------------|------|-------------|
|----------|------|------------|------|-------------|

---

### Operations

| NAME |
|------|
| GET  |

14. Click Save.
15. Click /val in the list of available paths.

### Paths

| NAME |
|------|
| /val |

16. Click GET in the list of Operations.
17. Scroll down. In the Parameters section, click Add.
  - a. Select REQUIRED.
  - b. Enter `Authorization` in the NAME field.
  - c. Select `header` in the LOCATED IN field.
  - d. Select `string` in the TYPE field.
  - e. Enter `Enter Bearer <jwt>` in the DESCRIPTION field.

### Parameters

[Add](#)

| REQUIRED                            | NAME          | LOCATED IN | TYPE   | DESCRIPTION        | DELETE |
|-------------------------------------|---------------|------------|--------|--------------------|--------|
| <input checked="" type="checkbox"/> | Authorization | header     | string | Enter Bearer <jwt> |        |

18. In the Response section, change the description of the pre-supplied 200 status code to 200 OK.

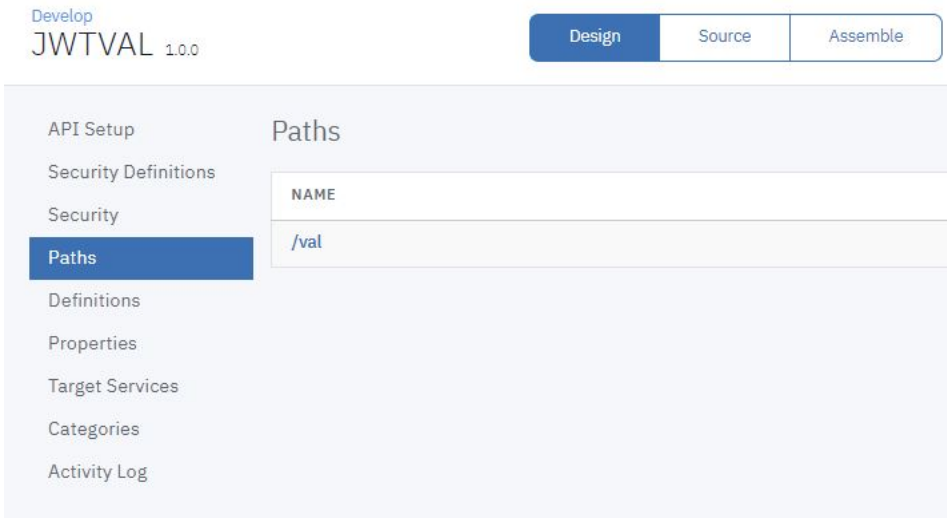
### Response

[Add](#)

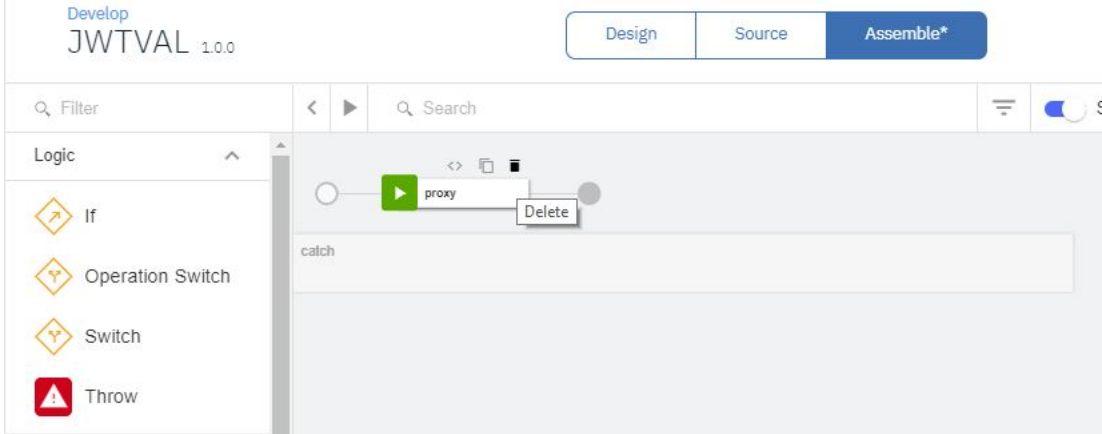
| STATUS CODE | SCHEMA | DESCRIPTION | DELETE |
|-------------|--------|-------------|--------|
| 200         | string | 200 OK      |        |

[Cancel](#) [Save](#)

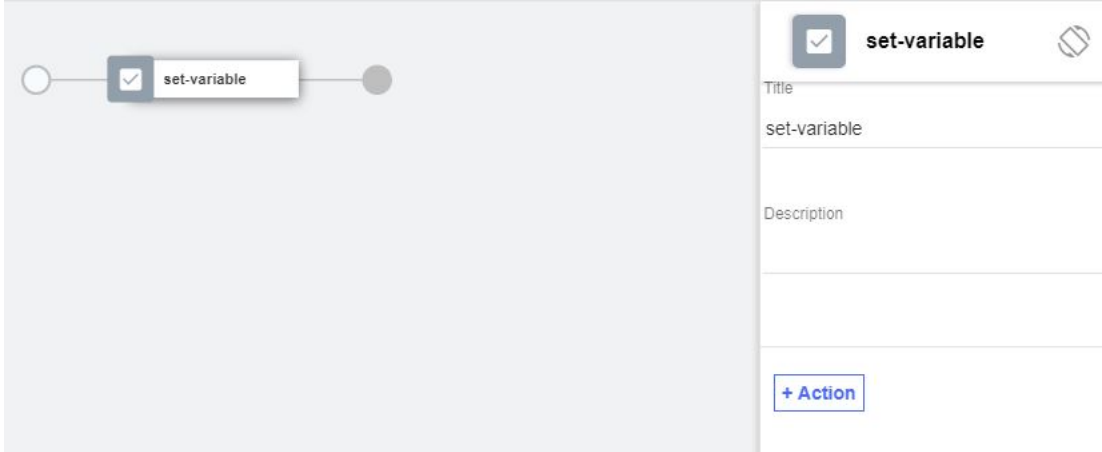
19. Click Save.
20. Click Assemble.



21. Hover the mouse over the existing Proxy or Invoke action and click the trash can icon to delete it.



22. Drag the Set Variable action onto the processing flow line. A configuration panel automatically opens.



- 23. Click + Action field.
- 24. Enter hs256-key in the Set field.
- 25. Select string in the Type field.
- 26. Enter a JWK in the Value field. Here is an example. 

```
{ "alg": "HS256", "kty": "oct", "use": "sig", "k": "o5yErLaE-dbgVpSw65Rq570A9dHyaF66Q_Et5azPa-XUjbyP0w9iRWhR4kru09aFfQLXeIODIN4uhjE1YKXt8n76jt0Pjkd2pqk4t9abRF6tnL19GV4pf1fL6uvVKkP4weOh39tqHt4TmkBgF2P-gFhgssZpjwq6182fz3dUhQ2nkzoLA_CnyDGLZLd7SZ1yv73uzfE2Ot813zmig8KTMEMWcWSDvy61F06vs_6LURcq_IEEevUiubBxG5S2akNnWigfphWYjMI5M22FOCpdcDBt4L7K1-yHt95Siz0QUB0MN1T_X8F76wh7_A37GpKKJGqeiNWmHkgWde8QWDQ", "kid": "hs256-key" }
```



Action \*

Set

Set, Add, or Clear a runtime variable.

Set

hs256-key

The name of the variable to be set.

Type \*

string

The type of the value to set. This can be string, number or boolean.

Value

```
{ "alg": "HS256", "kty": "oct", "use": "sig", "k": "o5yErLa
```

The value that the variable will be set to.

27. Close the property panel. Click Save.

28. Drag the Validate JWT action onto the processing flow line after the set-variable icon. A configuration panel automatically opens.

The screenshot shows the API Connect console interface. At the top, there are navigation icons (back, forward, search) and a 'Show catches' toggle. Below this is a flow diagram with two actions: 'set-variable' (with a checkmark icon) and 'jwt-validate' (with a green checkmark icon). A 'catch' area is visible below the flow. To the right, the configuration panel for the 'jwt-validate' action is open. It includes a title 'jwt-validate', a description, and a section for 'JSON Web Token (JWT)' with the following configuration: 'request.headers.authorization'. Under 'Output Claims', there is a field 'decoded.claims' with a description: 'Runtime variable to which the full se is assigned.'

29. Enter `hs256-key` in the Verify Crypto JWK variable name field.

Verify Crypto JWK variable name

hs256-key

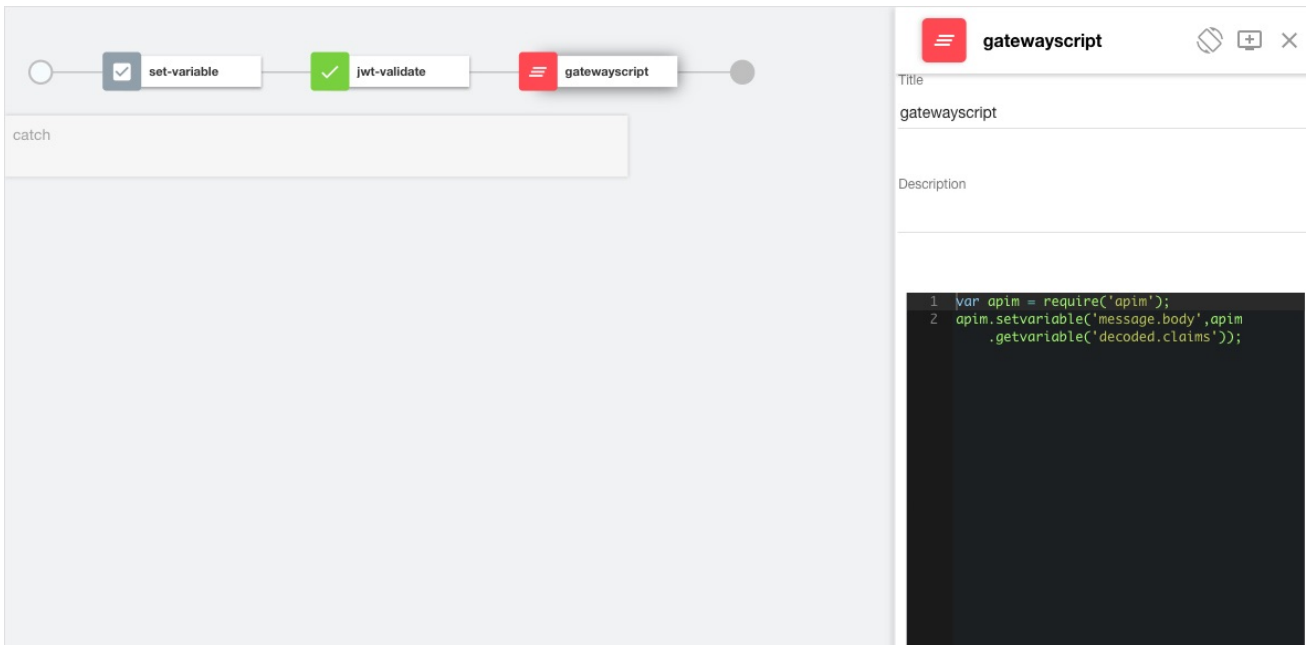
Runtime variable containing the JWK t

30. Close the property panel. Click Save.

31. Drag the GatewayScript action onto the processing flow line after the Validate JWT icon. A configuration panel automatically opens.

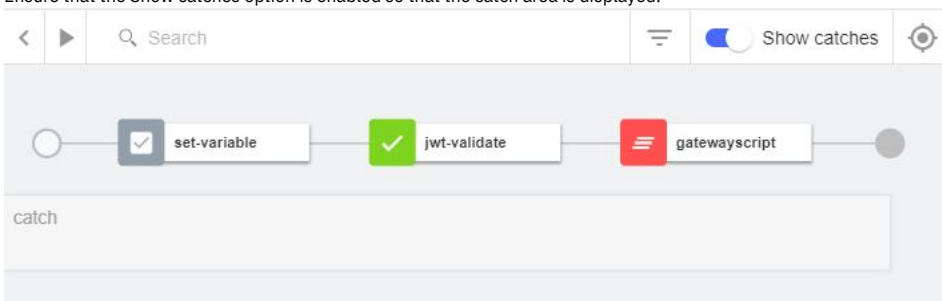
32. Enter the following code:

```
var apim = require('apim');
apim.setvariable('message.body', apim.getvariable('decoded.claims'));
```

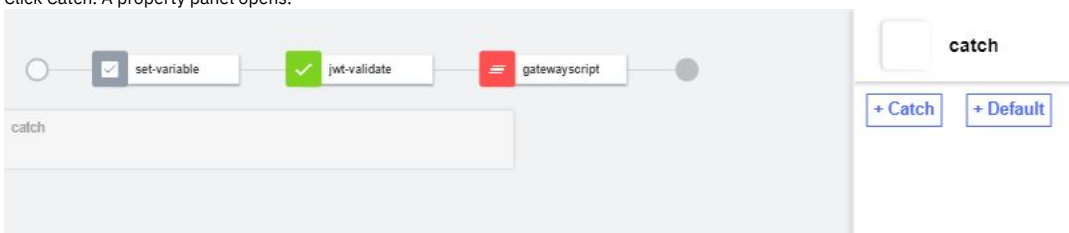


33. Close the property panel. Click Save.

34. Ensure that the Show catches option is enabled so that the catch area is displayed.



35. Click Catch. A property panel opens.

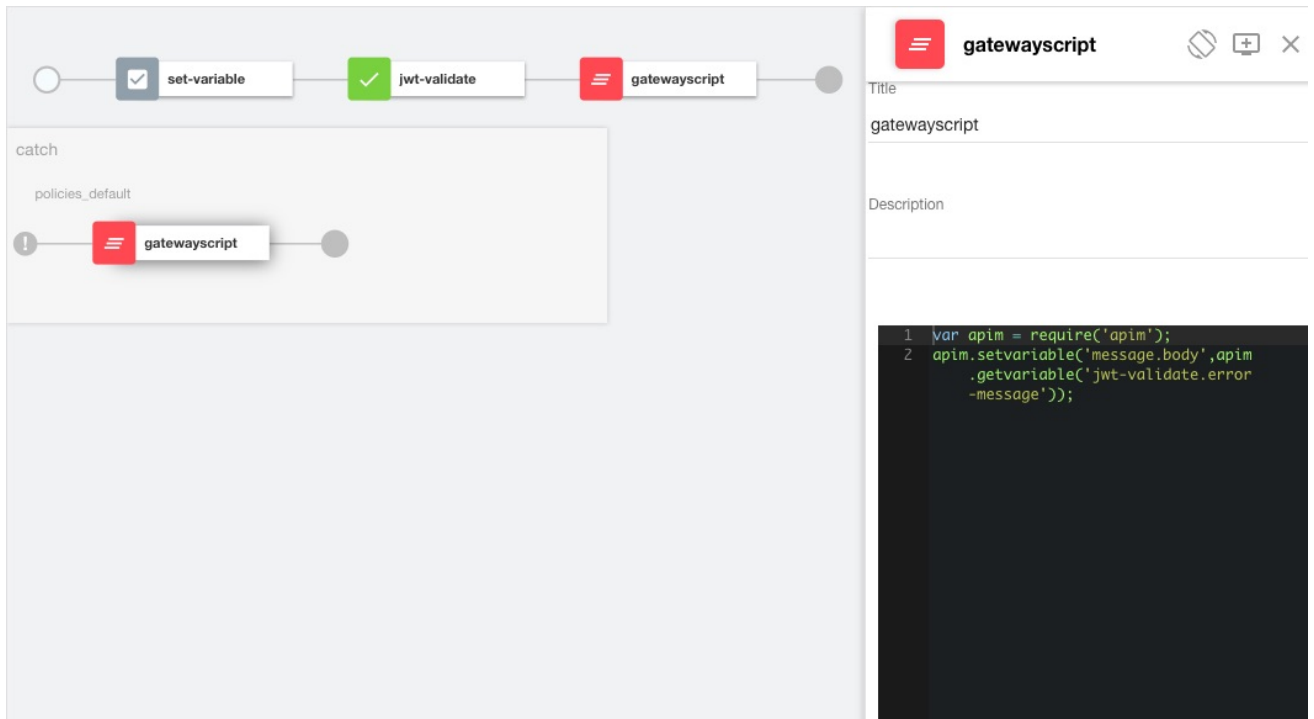


36. Click + Default.

37. Drag the GatewayScript policy action onto the catch flow line.

38. Enter the following code:

```
var apim = require('apim');
apim.setvariable('message.body', apim.getvariable('jwt-validate.error-message'));
```



39. Close the property panel. Click Save.

## Testing the REST API

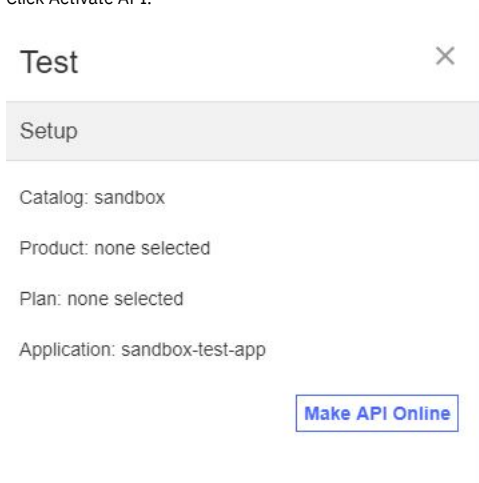
Note: Due to Cross-Origin Resource Sharing (CORS) restrictions, the assembly test tool cannot be used with the Chrome or Safari browsers on the macOS Catalina platform.

To test the REST API, you will need a valid JWT. You can obtain such a JWT by invoking the API created in the [Tutorial: Generate a JSON Web Token \(JWT\)](#). To complete testing, take the following steps:

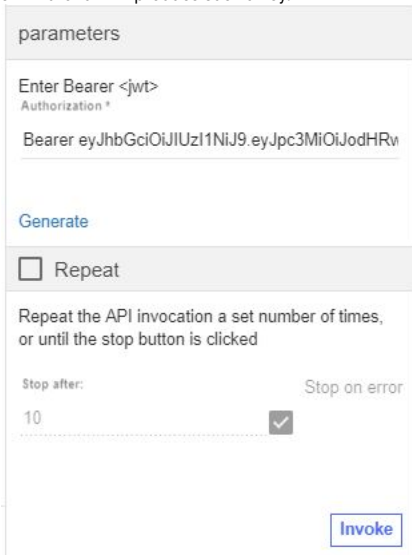
1. Click the Test icon .



2. Click Activate API.



3. Select the `get /val` Operation.
4. Enter `Bearer` followed by a space followed by a valid JWT generated with the same sign key in the Authorization field. Invoking the API created by the Generate JWT tutorial will produce such a key.



parameters

Enter Bearer <jwt>  
Authorization \*

Bearer eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRv

Generate

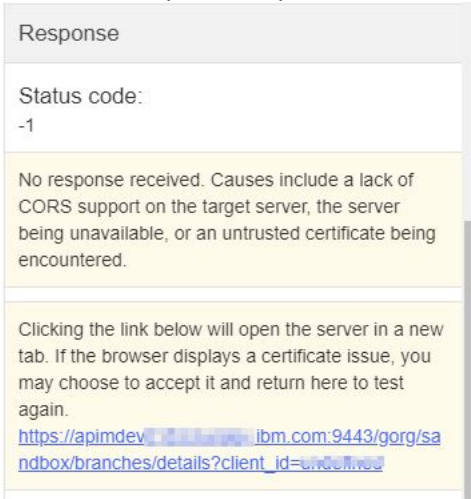
Repeat

Repeat the API invocation a set number of times, or until the stop button is clicked

Stop after: 10 Stop on error

Invoke

5. Click Invoke. You may encounter a yellow error box with a URL embedded in it. Click this URL to override a browser certificate error.



Response

Status code:  
-1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.

[https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client\\_id=...](https://apimdev.ibm.com:9443/gorg/sandbox/branches/details?client_id=...)

6. Click Invoke again. The response contains branch data.

[Invoke](#)

**Response**

Status code:  
200 OK

Response time:  
832ms

Headers:  
apim-debug-trans-id: -81b1780f-61f0-48e8-8f75-65556c19dd24  
x-global-transaction-id: 196c55655ba4e79500107543  
content-type: unknown

Body:


```
{
  "iss": "https://myidp.ibm.com",
  "aud": "ClientID1",
  "exp": 1537536977,
  "iat": 1537533377
}
```

[Debug](#)

## Manage your API definition

Now that your new API works correctly, you can manage this API. To see your immediate options, take the following steps.


1. Click the Develop icon  on the navigation bar.



Develop

JWTVAL 1.0.0


---



Test


×

---




Setup


---




Catalog: sandbox



Product: jwtval-auto-product



Plan: Default Plan







Application: sandbox-test-app

[Republish product](#)

2. Click the Options icon  alongside the Mapper API.

APIs and Products Add ▾

| TITLE   | TYPE       | LAST MODIFIED |   |
|---|------------|---------------|---|
|  JWT-1.0.0                 | API (REST) | 2 hours ago   |   |
|  JWTVAL-1.0.0              | API (REST) | 5 minutes ago | ⋮   |
|  JWT auto product-1.0.0    | Product    | 2 hours ago   | <ul style="list-style-type: none"> <li>Publish</li> <li>Stage</li> <li>Save as a new version</li> <li>Download</li> <li>Delete</li> </ul> |
|  JWTVAL auto product-1.0.0 | Product    | 5 minutes ago |   |

3. Select Download.

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a new API definition that validates a JSON Web Token (JWT).
- Tested the new API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Creating a Client Application

This tutorial shows you how to create a client application using the API Manager. The ability to create a new application using the API Manager allows members of a provider organization, such as API developers or testers, to create the environment needed to complete an API.

### About this tutorial

In this tutorial you will complete the following lessons:

1. [Create a Client Application](#)
2. [Subscribing to Products and APIs](#)

### Before you begin

In this tutorial you will create a new client application. You can then subscribe this new application to a product if one exists in your catalog. If you do not have an existing product, complete [Tutorial: Importing an API](#) first before beginning this tutorial.

### Create a Client Application

To create a client application using the API Manager, complete the following steps:

1. Log in to API Manager.
2. In the Welcome page, click the Manage Catalogs tile.

Welcome to the API Manager  
Choose an option to get started

- Develop APIs and Products**  
Edit, assemble, secure and test APIs. Package APIs using products for publishing to consumers.
- Manage Catalogs**  
Manage active APIs and consumers
- Manage Resources**  
Configure user registries, OAuth providers and TLS
- Manage Settings**  
Edit settings for roles, notifications and more.
- Learn more**  
Documentation and tutorials with step-by-step instructions
- Connect**  
Find expert answers in the API Connect community forum

3. Click the Sandbox catalog icon.

Sandbox  
Sandbox Catalog

4. Click the Application icon <sup>A</sup> in the side bar.  
5. Click Add > Create.

Manage / Sandbox  
Applications

| TITLE              | APPLICATION TYPE | CONSUMER ORGANIZATION | STATE   |   |
|--------------------|------------------|-----------------------|---------|---|
| > sandbox-test-app | Production       | sandbox-test-org      | Enabled | ⋮ |

Add ▾  
Create

6. Enter AppOne in the Title field.  
7. Enter https://example.com in the OAuth Redirect URL field.

# Create Application

## Application

**Title**  
AppOne

**Name**  
ppone

**OAuth Redirect URL (optional)**  
<https://example.com>

**Summary (optional)**

## Consumer Organization

Select the consumer organization that will own the application

8. Select the Sandbox Test Organization Consumer Organization.

## Consumer Organization

Select the consumer organization that will own the application


| Title   |
|---|
| <input checked="" type="checkbox"/> Sandbox Test Organization |

9. Optionally provide an X.509 certificate to identify the application during mutual authentication exchanges. It is not required.



### Certificate(Optional)

Upload the X509 certificate for your application, in PEM format



Drag & Drop  
your file here, or

[Browse](#)

Cancel Save

10. Click Create.
11. Copy the credentials presented in the dialog box. You will need these for repeated API testing. Click OK.

### Credentials ×

Save the client secret (it will no longer be retrievable for security purposes)

**Client ID**

6e64dd4f725b2f86412c24d4802eb197
Copy


**Client Secret**

c672b03b82e9326c0f8be7e7e4e5d359
Copy

Cancel OK

## Subscribing to Products and APIs

You can subscribe the new client application to an existing published product if you have already published at least one product. Complete the following steps to subscribe the client application to a product.

1. Click the options icon , then Create subscription.

| TITLE              | APPLICATION TYPE | CONSUMER ORGANIZATION | STATE   |
|--------------------|------------------|-----------------------|---|
| > AppOne           | Production       | sandbox-test-org      | Enabled   |
| > Sandbox Test App | Production       | sandbox-test-org      | <div style="border: 1px solid #ccc; padding: 5px;">           Edit<br/>           Credentials<br/> <b>Create subscription</b><br/>           Disable<br/>           Delete         </div> |

2. Select the Product/Plan for your API.

**Product/Plan**  
Select the product and plan for the application subscription

---

FindBranch auto product:2.0.0/Default Plan

3. Click Create.

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a client application.
- Subscribed the client to a product.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Authoring policies

With the on-premise offering, you can control specific processing features in the Gateway server of IBM® API Connect by creating user-defined policies.

IBM API Connect enables organizations to easily promote business services as APIs to internal and external developer communities. API developers can rapidly create, proxy, assemble, and secure APIs through the API Manager user interface (UI). The API Manager assembly feature allows API developers to create complex REST or SOAP API operations that transform data, perform multiple service calls, aggregate data, and apply policies.

A policy is a piece of configuration that controls a specific aspect of processing in the IBM DataPower® Gateway server during the handling of an API invocation at run time. IBM API Connect provides a number of different types of policies, but you can also create user-defined policies to provide more processing control.

## When would you need a user-defined policy?

A user-defined policy enables you to control extra processing features in the Gateway server, such as security, or routing of requests. The user-defined policies feature is available only with the on-premise offering of IBM API Connect.

Create a user-defined policy in IBM API Connect when you need to augment the actions or activities that are performed by the API gateway. For example, you might want to perform the following operations:

- Implement your own proprietary logic for dynamic routing of requests.
- Enforce extra security constraints to your API.
- Make accessible an extra capability that is provided in DataPower that is not yet accessible in the IBM API Connect policy catalog.

**Note:** For best results, do not try to update your existing user-defined policies, because those are already in use by your APIs. Instead, create a new version of the policy that you want to update, and then modify the API assembly to use the newer version of the policy.

See the following topics for more information about user-defined policies, and how to create and import them into IBM API Connect:

- [Authoring policies for the DataPower Gateway \(v5 compatible\)](#)

A user-defined policy for the DataPower Gateway (v5 compatible) consists of a package that contains a YAML definition file that describes the policy, together with the policy implementation files. You import the package into one or more Catalogs in your IBM API Connect environment to make the policy available to APIs that are published to those Catalogs.

- [Authoring policies for the DataPower API Gateway](#)

A user-defined policy for the DataPower API Gateway consists of a package that contains a configuration file of DataPower API Gateway CLI commands that define the actions of the policy, together with the policy implementation files. You publish the package to the DataPower API Gateway to make it available to APIs that are deployed there.

## Related information

- [IBM API Connect Overview](#)
- [API policies](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Authoring policies for the DataPower Gateway (v5 compatible)

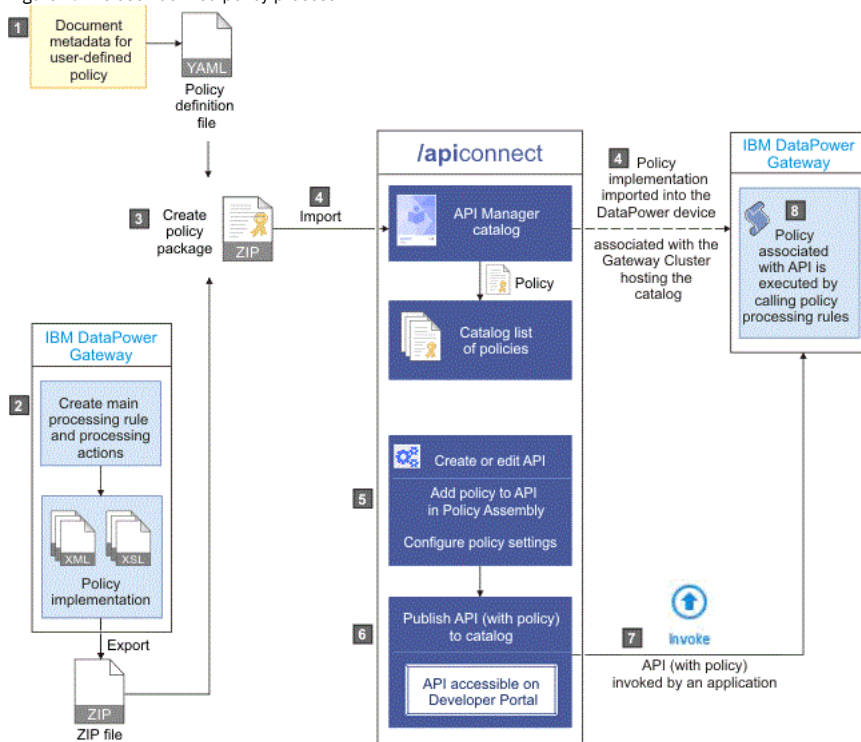
A user-defined policy for the DataPower® Gateway (v5 compatible) consists of a package that contains a YAML definition file that describes the policy, together with the policy implementation files. You import the package into one or more Catalogs in your IBM® API Connect environment to make the policy available to APIs that are published to those Catalogs.

## How do user-defined policies work?

A user-defined policy is implemented as a DataPower processing rule. After a policy is imported into an IBM API Connect Catalog, the policy is available to be placed into an assembly flow of an enforced API. When the API is published, and invoked by an application, the API Gateway executes all policies that are associated with this API. An API Gateway that is running in DataPower calls the IBM DataPower Gateway processing rules that those policies implement.

The following diagram provides an overview of how to create and execute a user-defined policy in IBM API Connect.

Figure 1. The user-defined policy process



For details on how to define and deploy user-defined policies for the DataPower Gateway (v5 compatible), see the following subtopics:

- [Describing your policy](#)  
Describe your user-defined policy by creating a definition file in YAML format to document metadata about the policy.
- [Creating a new user-defined policy](#)  
Create a user-defined policy by configuring a policy definition file and its implementation, and then packaging the policy and importing it into IBM API Connect.
- [Packaging and importing your policies into IBM API Connect](#)  
Make your user-defined policy available to API developers by packaging it and importing it into an IBM API Connect Catalog.
- [Reference](#)  
Reference information for authoring policies in IBM API Connect.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Describing your policy

Describe your user-defined policy by creating a definition file in YAML format to document metadata about the policy.

### About this task

A policy YAML file contains the following sections:

- A specification version.
- An information section.
- An attach section.
- A properties section.
- A gateways section.

You can create a policy YAML file by using any editor of your choice. Both .yaml and .yml file extensions are supported, but the use of the .yaml file extension is recommended by [yaml.org](http://yaml.org).

Note: All keys and enumeration values that are specified in this topic are case-sensitive.

### Procedure

The following steps describe how to construct a policy YAML file.

1. Set the specification version by adding the following line to the beginning of the file: `policy: 1.0.0`.
2. Complete the information section with details about the policy by using the following syntax:

```
info:
  title: Title of policy
  name: <policy-name>
  version: 1.0.0
  description: An example policy
  contact:
    name: name1
    url: url1
    email: email1
```

where:

- **title** is the title of the policy. Any string can be used, but the title should be kept short so that it can be displayed in the API Manager user interface.
- **name** is the short name for the policy and must contain only alphanumeric characters, the - (dash) character, and the \_ (underscore) character (blank spaces are not supported). The name is case-sensitive, and should be 20 characters or less so that it can be displayed in the API Manager user interface. In the example, the name is shown as `<policy-name>`.  
Important: The policy short name must be different from the OpenAPI names of the built-in policies, otherwise it will cause a policy conflict in the API assembly definition. For a list of the OpenAPI built-in policy names, see [execute](#). Also, consider including an appropriate unique prefix or suffix string in the short names of your user-defined policies to prevent possible name conflicts with future built-in policies.  
Note: The **name** property must be identical to the Node.js module name.
- **version** is the version number of the policy. This is a reserved property.  
Tip: The `version.release.modification` version numbering scheme is recommended, for example `1.0.0`.  
Note: The **info.version** in `policy.yaml` must match the **version** indicated in `package.json`.
- **description** (optional) is a short description of the policy.
- **contact** (optional) is the contact information for the policy, where:
  - **name** (string) is the identifying name of the contact person or organization.
  - **url** (string) is the URL that points to the contact information.
  - **email** (string) is the email address of the contact person or organization.

3. Complete the attach section by specifying which type of API flow the policy can be attached to.

```
attach:
- rest
- soap
```

where:

- **rest** indicates that this policy can be attached to a REST API flow.
- **soap** indicates that this policy can be attached to a SOAP API flow.

The attach section must contain at least one API flow type. If a policy can be attached to both API flow types, they must both be indicated in the attach section.

4. Complete the properties section by defining a JSON schema (based on JSON Schema Specification Draft 4, but rendered in YAML format) that contains the list of properties the policy will declare as required input. These properties are presented to the API author at development time, when the properties can be configured or mapped to values as required.

The JSON schema defines a root object that contains the following JSON properties:

- **type**
- **properties**
- **required**

The syntax and definition of these properties matches the JSON schema definition. The following code block shows an example of a simple schema that defines one required property (`a_property`):

```
properties:
  $schema: "http://json-schema.org/draft-04/schema#"
```

```

type: object
properties:
  a_property:
    label: a label
    description: a description
    type: a type
required:
  - a_property

```

The root *type* of the schema must be an **object**, as shown in the example schema. You can define zero or more properties in the schema. Required properties are declared by using the **required** parameter, as shown in the example schema. The policy will not be accessible in the API Manager assembly editor, unless all the required properties have values. Each property is an object with the following items:

- **label** is a short name for the property and is displayed in the Name column of the API Manager assembly editor.
  - **description** is a short description for the property and is displayed in the Description column of the API Manager assembly editor.
  - **type** must be one of the following primitives that are supported:
    - *integer*
    - *number*
    - *string*
    - *boolean* (this primitive is shown as a check box in the assembly editor)
    - *array*
  - **default** (optional) specifies a default value for the property.
  - **enum** (for properties with a **type** of *string*) is an array of valid values.
5. In your gateways section, specify that the policy is for the DataPower® Gateway. Enter the following code:

```

gateways:
  - datapower-gateway

```

6. Save the new policy as a YAML file using the value of the **name** property (in the **info** section) as the file's name. Required: To ensure a successful import, the YAML file must use the name specified in the policy's **name** property.

## Results

You have created a user-defined policy definition YAML file that documents metadata about your policy.

## Example

The following code block shows an example of a policy definition file that contains all the supported primitives. This policy must be saved as `sampleimpl.yaml` to ensure that it can be imported into API Connect.

```

policy: 1.0.0

info:
  title: Sample Policy
  name: sampleimpl
  version: 1.0.1
  description: This is a sample policy.
  contact:
    name: Steve Product Manager
    url: http://developer.acme.com/contacturl
    email: steve-product-manager@someemailservice.com

attach:
  - rest
  - soap

properties:
  $schema: "http://json-schema.org/draft-04/schema#"
  type: object
  properties:
    samplestring:
      label: String Property
      description: Sample string property
      type: string
    sampleboolean:
      label: Boolean Property
      description: Sample boolean property
      type: boolean
    sampleinteger:
      label: Integer Property
      description: Sample integer property
      type: integer
    samplenumber:
      label: Number Property
      description: Sample number property
      type: number
    samplestringwithenum:
      label: Enum Property
      description: Sample string enum property
      enum:
        - one
        - two
        - three
      default: one
      type: string
    samplearray:
      label: Array of properties
      description: Sample array of properties
      type: array

```

```

    items:
      type: object
      properties:
        string:
          label: String Property
          description: Sample string property in array
          type: string
        boolean:
          label: Boolean Property
          description: Sample boolean property in array
          type: boolean
      required:
        - string
  required:
    - samplestring
    - sampleboolean
    - sampleinteger
    - samplenumber
    - samplearray

```

```

gateways:
  - datapower-gateway

```

In the following code block, an example of a policy definition file for modifying message payload is shown.

```

policy: 1.0.0

info:
  title: Run GatewayScript
  name: gatewayscript-policy
  version: 1.0.0
  description: Execute GatewayScript
  contact:
    name: IBM DataPower Samples
    url: https://github.com/ibm-datapower/
    email: steve-product-manager@ibm.com

attach:
  - rest
  - soap

gateways:
  - datapower-gateway

properties:
  $schema: "http://json-schema.org/draft-04/schema#"
  type: object
  properties:
    source:
      label: "GatewayScript Source"
      description: "The location of the GatewayScript file to execute"
      type: string
      default: store:///identity.js
    value:
      label: "New Value"
      description: "The value to be added to the payload"
      type: string
      default: "Hello Policy"
  required:
    - source
    - value

```

## What to do next

Create an implementation for your user-defined policy by using DataPower processing rules and actions. For more information, see [Implementing your policy](#).

## Related concepts

- [Authoring policies](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Creating a new user-defined policy

Create a user-defined policy by configuring a policy definition file and its implementation, and then packaging the policy and importing it into IBM® API Connect.

Review the following topics to learn how to create a user-defined policy.

Tip: Some sample policies are available on GitHub, and these policies can be downloaded and adapted for use with IBM API Connect. For more information, see [API Connect Policies on GitHub](#).

- [Implementing your policy](#)  
Create an implementation for your user-defined policy by using DataPower processing rules and actions.

## Related information

---

- [IBM API Connect Overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

DataPower Gateway (v5 compatible) only

## Implementing your policy

Create an implementation for your user-defined policy by using DataPower® processing rules and actions.

### Before you begin

---

You must create a policy YAML file before you create an implementation. For more information, see [Describing your policy](#).

You can create an implementation for your policy by using the usual tools available to IBM® DataPower Gateway developers. However, the DataPower user interface is typically the tool that is used to create processing rules and actions.

**Note:** You must be skilled in DataPower tooling and concepts, before you can create a policy implementation.

In order for your processing actions to work well under the API Gateway configuration, and also to enable the use of contextual information that is relevant to the processing actions, a library of functions and templates is available that you can use to assist you in writing the XSLT or GatewayScript transformations. This library provides the following mechanisms:

- Access to input property values (input from the assembly editor).
- Access to context values (the API Gateway runtime context).
- Access to the runtime payload message and media-type.
- Ability to modify the payload message.
- Ability to end the policy execution with an error.

### About this task

---

You create an implementation for your user-defined policy in the IBM DataPower Gateway. The implementation must adhere to the following conventions:

1. You must create a main processing rule, and this rule will be the starting point for the policy. The rule name must start with the name of the policy (the value of the **name** property in the "info" section of the policy YAML file), followed by -main; for example *checkmembership-main*.
2. There are no restrictions on what actions the processing rule can execute, on condition that the rule adheres to the naming convention in point 1. and the instructions that are detailed in this topic.
3. The names of the processing actions and all other objects also must start with the name of the user-defined policy (the value of the **name** property in the "info" section of the policy YAML file).
4. If your processing rule runs transformation actions that use XSLT or GatewayScript files, these files must be stored in the following location: `local://policy/<policy-name>`.
5. If you have certificate and key files that must be stored in the `cert:` folder, the names of these files also must start with the name of the user-defined policy (as defined in the policy YAML file).
6. If your policy is using GatewayScript transformations, your policy code must require `apim.custom.js`, for example:

```
var apic = require('local://isp/policy/apim.custom.js');
```

The following steps describe how to create an implementation for your user-defined policy:

- [Create a main processing rule.](#)
- [Create one or more processing actions:](#)
  - [Configure access to input properties.](#)
  - [Configure access to the runtime context.](#)
  - [Configure access to the input payload.](#)
  - [Configure access to the HTTP headers.](#)
  - [Modify the payload in XSLT or gatewayscript.](#)
  - [Configure the implementation to produce error information.](#)
  - [Set variables.](#)
- [Export your policy implementation.](#)

## Procedure

---

1. Create a main processing rule that is called `<policy-name>-main` where `<policy-name>` is the name of your user-defined policy (the value of the **name** property in the "info" section of the policy YAML file).  
**Note:** You must specify the following processing rule settings:
  - Rule direction: Both Directions
  - Non-XML Processing: on
2. Create one or more processing actions.

Each processing action should have a unique name and must start with the name of the user-defined policy (as defined in the policy YAML file).

Note: Different processing actions in DataPower might require different releases of the DataPower firmware. When using the GatewayScript actions, the minimum DataPower firmware that should be used is version 7.2.0.

The following processing actions are available:

- Configure access to the input properties in XSLT (`policyProperties`), or in GatewayScript (`apic.getPolicyProperty`).  
If required by the properties that are defined in the policy YAML file, when the policy is attached to an API the API developer must enter the input settings or variables for a particular property. This function provides the mechanism to retrieve these input settings or variables at run time. For an example code snippet, see [Access to input properties code snippet](#).
- Configure access to the runtime context in XSLT (`getContext`), or in GatewayScript (`apic.getContext`).  
When an API is invoked, runtime information about the request can be accessed by using a user-defined policy, and this information is known as the runtime context. The function `getContext` provides the mechanism to access this runtime context. For an example code snippet, see [Access to runtime context code snippet](#). For a complete list of context variables, see [API Gateway context variables](#).
- Configure access to the input payload in XSLT (`payloadRead`), or in GatewayScript (`apic.readInput(callback)`).  
Some policies require access to the input message or payload that is provided by the application. Accessing the input message or payload during an assembly flow can be difficult, as this information might change as the policies and the assembly steps run. This function provides a mechanism to get the correct input message or payload at the time of policy execution. For an example code snippet, see [Access to input payload code snippet](#).

This function returns an XML node-set that contains the payload of the request. If the payload is in JSON format, a JSONx node-set is returned that can then be manipulated within an XSLT or GatewayScript stylesheet. If the payload is not in JSON or XML format, the node-set that is returned is empty.

Note: The `payloadType()` function can be called to determine what type of payload (XML or JSONx) will be returned by the `payloadRead()` or the `apic.readInput(callback)` function.

- Configure access to the HTTP headers in XSLT (`getContext`), or in GatewayScript (`apic.getContext`).  
The HTTP header information of a request can be retrieved from the request context. The HTTP header names are normalized to lowercase. The function `getContext()` provides the mechanism to access the HTTP header information. For an example code snippet, see [Access to HTTP headers code snippet](#).

Note: Access or modification of HTTP headers by using DataPower extensions, such as `dp:set-request-header`, is not advisable, as such actions might yield unexpected results when the policy is combined with other policies and assembly steps.

- Modify the payload in XSLT or GatewayScript.  
When a policy implementation is required to change a payload message, you must configure the DataPower Transform Processing Action to set the Output context field to `OUTPUT` (referred to as the `OUTPUT` named context).

The output must be an XML node-set, which represents an XML or SOAP message, or a JSON message by using JSONx. To assist the API Gateway policy framework to accept the new or transformed message, call the `apim-output` template. For an example code snippet, see [Modify the payload code snippet](#).

Note: The modify the payload processing action works only with a proxy task. An HTTP or web service task always overwrites the output of a policy implementation.

- Configure the implementation to produce error information.  
If you require your policy implementation to produce error information when there is a failure, you must configure the implementation by calling the error template. For an example code snippet, see [Configure error information code snippet](#).
- Set variables.  
Use the `setVariable` template to set a runtime variable to a specified string value. This value can then be retrieved by using the function `getVariable()`, or by mapping the value into a step. For an example code snippet, see [Set variables code snippet](#).

### 3. Export your policy implementation from DataPower.

From the DataPower user interface, export the objects and files that are referenced by your main processing rule as a compressed file. No other objects or files should be exported. The following file list is an example configuration of an export from DataPower:

```
Archive: checkmembership.zip
11819  11-19-2014 22:53 dp-aux/basetypes.xml
3028207 11-19-2014 22:53 dp-aux/drMgmt.xml
6456  11-19-2014 22:53 dp-aux/map-dmz.xsl
7299  11-19-2014 22:53 dp-aux/management.xsl
23130 11-19-2014 22:53 dp-aux/SchemaUtil.xsl
216003 11-19-2014 22:53 dp-aux/clixform.xsl
5284  11-19-2014 22:53 local/policy/checkmembership/check.xsl
5061  11-19-2014 22:53 export.xml
```

The example shows the exported configuration file (export.xml) that contains all the processing actions and DataPower objects, the referenced files, and the DataPower configuration schemas that are created by using the configuration export in the DataPower user interface.

## Results

---

You have created an implementation for your user-defined policy and exported this implementation from the DataPower user interface.

## What to do next

---

Create a user-defined policy package that can then be imported into IBM API Connect. For details, see [Packaging and importing your policies into IBM API Connect](#).

## Related concepts

---

- [Authoring policies](#)

## Related information

---

- [IBM DataPower Gateway](#)



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Packaging and importing your policies into IBM API Connect

Make your user-defined policy available to API developers by packaging it and importing it into an IBM® API Connect Catalog.

### Before you begin

Before you can package a user-defined policy and import it into IBM API Connect, you must complete the following tasks:

1. [Describe your policy](#), in a YAML file.
2. [Implement your policy](#), by using DataPower processing rules and actions.

### Procedure

To package and import your user-defined policy, complete the following steps:

1. Create a .zip file that contains the following folder structure:

```
<policy-name>.yaml
implementation/
  policy-name.zip
```

where

- `<policy-name>.yaml` is your policy definition file. The YAML file must use the name that is specified in its `name` property.
- `implementation/policy-name.zip` is your policy implementation for a policy that is deployed to the DataPower Gateway. The policy implementation file contains the DataPower processing rules and actions that were exported from DataPower in step 2.

The name of the .zip file must start with the name of the user-defined policy (as defined in the policy YAML file). If the implementation requires certificate and key files, these files must be added to the implementation directory.

Note: Your package .zip file must contain a .zip policy implementation file.

2. Use the API Manager user interface to import your policy into a Catalog. For more information, see [Importing a user-defined policy into a Catalog](#).

Note:

- You must import your user-defined policy into every Catalog in API Manager that you require your policy to run in.
- For the policy to be displayed on the palette in the [API assembly editor](#), you must import the user-defined policy into the Sandbox Catalog.

### What to do next

Make your user-defined policy package accessible to the API Designer UI. To display user-defined policies in the API Designer UI, edit the .apiconnect/config file to point to the directory containing the policy.yaml file. You can edit the global config file (~/.apiconnect/config) or a project-specific config file (~/ProjectA/.apiconnect/config). Follow these steps:

1. Create one or more directories to hold your policy.yaml files that are accessible to Node.js, and copy the user-defined policy package into this directory. For example, create directories named /ProjectA/Policies and /ProjectB/Policies.
2. Extract the policy package so that the folder structure is visible, for example `policy.yaml` and `implementation/policy-name.zip`. The file containing the custom policies must be named `policy.yaml`.
3. Add the `userPolicies`: [ ] array object to the config file in `~/ProjectA/.apiconnect/` or `~/.apiconnect/`, and edit the array object so it contains the absolute path to the director(ies) containing the policy.yaml file.

Following is an example of an `~/.apiconnect/config` file showing the `userPolicies` entry pointing to the directory containing the policy.yaml file:

```
apim_server: us.apiconnect.ibmcloud.com
us.apiconnect.ibmcloud.com.meta:
  formFactor: BLUEMIX_PUBLIC
  serverCapabilities:
    authTypes:
      - basic
      - token
  toolkit:
    version_recommended: 2.1.30
    version_minimum: 2.1.30
catalog: >-
  apic-catalog://us.apiconnect.ibmcloud.com/orgs/sampleorg-myspace2/catalogs/sb
app: >-
  apic-app://us.apiconnect.ibmcloud.com/orgs/sampleorg-myspace2/apps/acme-bank
userPolicies:
  - /ProjectA/Policies
```

Note:

- The `userPolicies`: [ ] array object contains a list of absolute paths to the policy director(ies). The values can point to a parent location, for example `userPolicies`: ["/~/mypolicies"]. Alternatively, each policy location can be specified in the array, for example `userPolicies`: ["/~/mypolicies/policya", "/~/mypolicies/policyb"]. However, do not use both of these approaches in the same directory. For example, do not create policies in a directory structure like this `~/foo`, `~/foo/policya`, and `~/foo/policyb`.

4. Open the API Designer UI. The user-defined policy is available on the policy assembly palette, and you can add the policy to your API definition.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Reference

Reference information for authoring policies in IBM® API Connect.

- [Implementation code examples](#)  
Example code snippets to help the creation of a user-defined policy implementation.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

DataPower Gateway (v5 compatible) only

## Implementation code examples

Example code snippets to help the creation of a user-defined policy implementation.

Note: If you are using GatewayScript, you must include the following command:

```
var apic = require('./apim.custom.js');
```

where *apic* is the common name used for the GatewayScript examples in this topic. However, *apic* could be any given name of your choice, for example you could use:

```
var apim = require('./apim.custom.js');
```

and then you would start your calls with **apim**.

- [Access to input properties code snippet](#)
- [Access to runtime context code snippet](#)
- [Access to input payload code snippet](#)
- [Access to HTTP headers code snippet](#)
- [Modify the payload code snippet](#)
- [Configure error information code snippet](#)
- [Set variables code snippet](#)

## Access to input properties code snippet

The following code block shows an example of how to access the input properties by using the XSLT `policyProperties()` function. The example defines a property that is named `a_property`, which is declared as an integer value, but is retrieved in XSLT as text.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimmanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="p" select="apim:policyProperties()" />
    <xsl:message>
      The value of my input property is
      <xsl:value-of select="$p/a_property" />
    </xsl:message>
  </xsl:template>
</xsl:stylesheet>
```

If you are using GatewayScript, you must call the function:

```
apic.getPolicyProperty(propertyName)
```

where *propertyName* is the name of the input property that you want to access. If the input property name is blank, the action will return all input properties.

## Access to runtime context code snippet

The following code block shows an example of how to access the runtime context by using the XSLT `getContext()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="client-id" select="apim:getContext('client.app.id') " />
    <xsl:message>
      The calling application is
    <xsl:value-of select="$client-id" />
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, you must call the function:

```
apic.getContext(varName)
```

where `varName` is the name of the context variable that you want to access.

For a complete list of context variables, see [API Gateway context variables](#).

## Access to input payload code snippet

The following code block shows an example of how to access the input payload by using the XSLT `payloadRead()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="input" select="apim:payloadRead() " />
    <xsl:message>
      The input payload is
    <xsl:copy-of select="$input" />
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, you must call the function:

```
apic.readInput(callback)
```

A callback is required because the actual payload read is asynchronous. The callback method is called when the payload is ready.

This function returns an XML node-set that contains the payload of the request. If the payload is in JSON format, a JSONx node-set is returned that can then be manipulated within an XSLT or GatewayScript stylesheet. If the payload is not in JSON or XML format, the node-set that is returned is empty.

The following example shows how to use the `payloadType()` function to determine what type of payload (XML or JSONx) will be returned by the XSLT `payloadRead()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:variable name="payloadType" select="apim:payloadType() " />
    <xsl:message>
      <xsl:text>Payload type is [</xsl:text>
    <xsl:value-of select="$payloadType" />
    <xsl:text>]</xsl:text>
    </xsl:message>
  </xsl:template>

</xsl:stylesheet>

```

## Access to HTTP headers code snippet

The following code block shows an example of how to access the HTTP headers in XSLT by using the `getContext()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

```

```

xmlns:dp="http://www.datapower.com/extensions"
xmlns:func="http://exslt.org/functions"
xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

<!-- Contains the APIM functions -->
<xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

<xsl:template match="/">
  <xsl:variable name="content-type" select="apim:getContext('request.headers.content-type')"/>
  <xsl:message>
    The request content type is
    <xsl:value-of select="$content-type" />
  </xsl:message>
</xsl:template>

</xsl:stylesheet>

```

If you are using GatewayScript, you must call the function:

```
apic.getContext(request.headers.headerName)
```

where `headerName` maps to the name of the header you want to access.

Note: Access or modification of HTTP headers by using DataPower® extensions, such as `dp:set-request-header`, is not advisable, as such actions might yield unexpected results when the policy is combined with other policies and assembly steps.

## Modify the payload code snippet

The output from a user-defined policy must be an XML node-set, which represents an XML or SOAP message, or a JSON message by using JSONx. The following code block shows an example of how to modify the payload in XSLT. To assist the API Gateway policy framework to accept the new or transformed message, call the `apim-output` template, as shown in the following example.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:apim="http://www.ibm.com/apimanagement"
  xmlns:jsonx="http://www.ibm.com/xmlns/prod/2009/jsonx">

  <!-- Contains the APIM functions -->
  <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <!-- Creates a JSON document (empty object is for simplicity) -->
    <jsonx:object>
    </jsonx:object>

    <!-- Indicates the media type of the output being produced -->

    <xsl:call-template name="apim:output">
      <xsl:with-param name="mediaType" select="'application/json'"/>
    </xsl:call-template>
  </xsl:template>

</xsl:stylesheet>

```

where `mediaType`:

- `'application/json'` is when the output is written in JSONx format.
- `'application/xml'` is when the output is written in XML format.

If you are using GatewayScript, you must call the function:

```
apic.output(mediaType)
```

where `mediaType` is:

- `application/json` is when the output is written in JSONx format.
- `application/xml` is when the output is written in XML format.

Specifying the media type allows the next steps in the assembly flow to understand how to process the new payload.

Tip: The output from a user-defined policy must be XML or JSONx. JSONx is an IBM standard format to represent JSON as XML. One way to convert output GatewayScript JSON data into JSONx, is to add a **Convert Query Params to XML** action to follow the GatewayScript action within the same policy rule. The **Convert Query Params to XML** action must have an **Input Conversion** with the **Default Encoding** set to **JSON**. The output from the GatewayScript action must be the input for the **Convert Query Params to XML** action for JSONx to be produced.

## Configure error information code snippet

The following XSLT code block shows an example of how to configure the policy implementation to produce error information by calling the `apim-error` template.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:apim="http://www.ibm.com/apimanagement">

  <!-- Contains the APIM functions -->
  <xsl:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

```

```

<!-- Indicates this policy has a failure and provides
      additional information for the client application -->
<xsl:template match="/">
  <xsl:call-template name="apim:error">
    <xsl:with-param name="httpCode" select="'401'" />
    <xsl:with-param name="httpReasonPhrase" select="'Unauthorized'" />
    <xsl:with-param name="errorMessage" select="'Please select a Plan'" />
  </xsl:call-template>
</xsl:template>

```

```
</xsl:stylesheet>
```

where:

- `httpCode` is the code of the required error message.
- `httpReasonPhrase` is the reason for the error.
- `errorMessage` is the suggested action for the user.

If you are using GatewayScript, you must call the template:

```
apim.error(name, httpCode, httpReasonPhrase, message)
```

where:

- `name` is the name of the error.
- `httpCode` is the code of the required error message.
- `httpReasonPhrase` is the reason for the error.
- `message` is the suggested action for the user.

## Set variables code snippet

The following XSLT code block shows an example of how to set a runtime variable to a specified string value by calling the `setVariable` template.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"
  xmlns:apim="http://www.ibm.com/apimmanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
    href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

  <xsl:template match="/">
    <xsl:call-template name="apim:setVariable">
      <xsl:with-param name="varName" select="'serviceEndpoint'" />
      <xsl:with-param name="value" select="'https://endpoint.host.com/data'" />
    </xsl:call-template>
    <xsl:message>
      <xsl:text>Variable [ </xsl:text>
      <xsl:value-of select="'serviceEndpoint'" />
      <xsl:text>] set to [ </xsl:text>
      <xsl:value-of select="'https://endpoint.host.com/data'" />
      <xsl:text>] </xsl:text>
    </xsl:message>
  </xsl:template>

```

```
</xsl:stylesheet>
```

where:

- `varName` is the name of the runtime variable that you want to set a value to.
- `value` is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, you would specify the `value` as `$(request.headers.content-type)`.

If you are using GatewayScript, you must call the template:

```
apim.setvariable(varName, varValue, action)
```

where:

- `varName` is the name of the runtime variable that you want to set a value to, or that you want to add or clear.
- `varValue` is the string value that you want to set the variable to. This can be a literal value, or another variable. For example, to set your named variable to the value of the Content-Type header in a request, you would specify the `varValue` as `request.headers.content-type`. This property is only required when `set` or `add` is specified as the action.
- `action` is the action that you want to apply to the variable. Valid options are:
  - `set`
  - `add`
  - `clear`

If no option is set, the default option of `set` is applied.

The following XSLT example shows how to retrieve the value of a runtime variable by using the `getVariable()` function.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  xmlns:func="http://exslt.org/functions"

```

```

xmlns:apim="http://www.ibm.com/apimanagement" extension-element-prefixes="dp func apim">

<!-- Contains the APIM functions -->
<xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.policy.doc_local:_isp_policy_apim.custom.xsl" />

<xsl:template match="/">
  <xsl:variable name="varValue" select="apim:getVariable('serviceEndpoint')" />
  <xsl:message>
    <xsl:text>Variable [</xsl:text>
    <xsl:value-of select="'serviceEndpoint'" />
    <xsl:text>] = [</xsl:text>
    <xsl:value-of select="$varValue" />
    <xsl:text>]</xsl:text>
  </xsl:message>
</xsl:template>

</xsl:stylesheet>

```

where

- **varValue** is the name of the runtime variable that you want to retrieve a value for.

If you are using GatewayScript, you must call the function:

```
apic.getvariable(varName)
```

where **varName** is the name of the runtime variable that you want to retrieve a value for.

## Related concepts

- [Authoring policies](#)

## Related tasks

- [Implementing your policy](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

API Gateway only

## Authoring policies for the DataPower API Gateway

A user-defined policy for the DataPower® API Gateway consists of a package that contains a configuration file of DataPower API Gateway CLI commands that define the actions of the policy, together with the policy implementation files. You publish the package to the DataPower API Gateway to make it available to APIs that are deployed there.

For details on how to define and publish user-defined policies for the DataPower API Gateway, see the following subtopics:

- [Defining and packaging your user-defined policy for the DataPower API Gateway](#)  
You define your policy by creating a .cfg configuration file. This configuration file consists of DataPower API Gateway CLI commands that specify the actions of your policy. You then package this file with any dependent files that are referenced from the CLI commands.
- [Publishing your user-defined policy to the DataPower API Gateway](#)  
You publish a user-defined policy by uploading the policy package .zip file, as a gateway extension, to the DataPower API Gateway.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

API Gateway only

## Defining and packaging your user-defined policy for the DataPower API Gateway

You define your policy by creating a .cfg configuration file. This configuration file consists of DataPower® API Gateway CLI commands that specify the actions of your policy. You then package this file with any dependent files that are referenced from the CLI commands.

### Configuration file example

The following code extracts describe the sections in a sample policy configuration file. You can obtain the complete sample by downloading the file [udp-basic.policy.txt](#); to re-use this file, rename the extension to .cfg.

Define a **set-variable** assembly policy

```

assembly-setvar udp-basic_1.0.0_set-variable_0
reset
title "set-variable"
correlation-path "$.x-ibm-configuration.assembly.execute[0]"
variable
  action set
  name "param1"
  type string
  value "$(local.parameter.credential)"
exit
exit

```

These commands define a `set-variable` assembly policy that sets a variable called `param1` to the value supplied by the API developer in the API assembly editor. The variable `$(local.parameter.credential)` refers to an input field labeled `credential` that is displayed in the properties window for the policy when it is added to an API assembly, and the value of the variable will be set to the value entered in that field.

Define a `gatewayscript` assembly policy

```

assembly-gatewayscript udp-basic_1.0.0_gatewayscript_1
reset
title "gatewayscript"
correlation-path "$.x-ibm-configuration.assembly.execute[1]"
gatewayscript-location temporary:///filestores/extensions/gateway-extension/udp-basic/udp-basic-gws.js
exit

```

This command defines a `gatewayscript` assembly policy that executes the GatewayScript in the file `udp-basic-gws.js` (see [The code for the GatewayScript policy](#)); this file must be packaged with the configuration file when the policy is deployed to the DataPower API Gateway. The `gatewayscript-location` command must specify the location of the file in the gateway extension folder on the gateway after the policy is deployed, the root of which will be `temporary:///filestores/extensions/gateway-extension`.

Build the DataPower API Gateway configuration for the policy

```

api-rule udp-basic_1.0.0_main
reset
action udp-basic_1.0.0_set-variable_0
action udp-basic_1.0.0_gatewayscript_1
exit

assembly udp-basic_1.0.0
reset
rule udp-basic_1.0.0_main
exit

```

These commands specify the policies, in the required sequence, that will be invoked by the user-defined policy.

Create the policy

```

assembly-function "udp-basic_1.0.0"
reset
summary "udp-basic-policy_1.0.0"
title "UDP Basic"
parameter
  name "credential"
  description "Parameter name"
  value-type string
exit
assembly udp-basic_1.0.0
exit

```

These commands create the user-defined policy and define the information that will be displayed in the properties window for the policy when it is added to an API assembly. The `parameter` section causes an input field labeled `credential` to be displayed; it is the value entered in this field, by the API developer, that is referenced by the `set-variable` policy definition described in [Define a set-variable assembly policy](#).

Important: The policy short name, `udp-basic` in this example, must be different from the OpenAPI names of the built-in policies, otherwise it will cause a policy conflict in the API assembly definition. For a list of the OpenAPI built-in policy names, see [execute](#). Also, consider including an appropriate unique prefix or suffix string in the short names of your user-defined policies to prevent possible name conflicts with future built-in policies.

Deploy the policy to the DataPower API Gateway

```

apic-gw-service
admin-state disabled
exit

apic-gw-service
admin-state enabled
exit

apic-gw-service
user-defined-policies udp-basic_1.0.0
exit

```

These commands restart the API Connect Gateway Service to trigger the API Connect management service to push the user-defined policy to the DataPower API Gateway, then add the user-defined policy to the API Connect Gateway Service. After the user-defined policy is published to the DataPower API Gateway, the policy will, as a result of these commands, be displayed in the palette on the assembly page in the API editor, from where it can be added to an API assembly flow.

The code for the GatewayScript policy

The code for the GatewayScript policy that is referenced by the `gatewayscript` policy definition described in [Define a gatewayscript assembly policy](#) is as follows:

```

var apim = require('apim');

var param1 = context.get('param1');
console.info ("param1 %s", param1);

```

```

if (param1 == null || param1 == '') {
    context.reject('Invalid Parameter', 'The parameter param1 you provided is invalid');
    context.message.statusCode = '401 Unauthorized';
}

var jsonBody = {
    "param1" : param1,
    "body" : apim.getvariable('message.body')
}

session.output.write(jsonBody);

```

The variable `param1` contains the value that is entered by the API developer in the credential field for the user-defined policy. If the policy is placed after an `invoke` policy in an API assembly, the `param1` value is prepended to the response body returned by the `invoke` policy.

## Packaging your user-defined policy

To package your user-defined policy, create a .zip file that contains the .cfg file together with all policy implementation files.

Note: When you publish the policy to the DataPower API Gateway, the files in the .zip file are uploaded to the temporary:///filestores/extensions/gateway-extension/ folder in the gateway file system, so any file references in your policy configuration must be defined accordingly.

## What to do next

Publish your user-defined policy; see [Publishing your user-defined policy to the DataPower API Gateway](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



## Publishing your user-defined policy to the DataPower API Gateway

You publish a user-defined policy by uploading the policy package .zip file, as a gateway extension, to the DataPower® API Gateway.

## Before you begin

Define and package your policy; see [Defining and packaging your user-defined policy for the DataPower API Gateway](#).

Note: This page includes instructions for restarting the gateway, which is essential for the user-defined policy to be published successfully.

## Procedure

To publish your user-defined policy, complete the following steps:

1. Log in to your API Connect Management server as an administrator, by using the following command:

```
apic login --server mgmt_endpoint_url --username admin_user_ID --password admin_password --realm admin/identity_provider
```

where:

- `mgmt_endpoint_url` is the platform API endpoint URL.
- `admin_user_ID` is the user ID of your administrator account, and is the same as the user ID that you use to log in to the Cloud Manager user interface.
- `admin_password` is the password for your administrator account.
- `identity_provider` is the identity provider that is used to authenticate administrative users.

For example:

```
apic login --server platform-api.myserver.com --username admin --password password --realm admin/myldap
```

For full details on how to log in to your management server from the CLI, see [Logging in to the management server](#).

2. Upload the user-defined policy .zip file, as a Gateway extension, by using the following command:

```
apic gateway-extensions:create udp_zip_file --scope org --org admin --gateway-service gateway_service --availability-zone availability-zone --server mgmt_endpoint_url
```

where:

- `udp_zip_file` is the user-defined policy .zip file that you want to upload.
- `gateway_service` is the name of the Gateway service that you want add the extension to.
- `availability-zone` is the name of the availability zone that contains the Gateway service.
- `mgmt_endpoint_url` is the platform API endpoint URL.

For example:

```
apic gateway-extensions:create myudp.zip --scope org --org admin --gateway-service mygateway-service --availability-zone myavailabilityzone --server platform-api.myserver.com
```

You can confirm that the user-defined policy has been added to the Gateway service by using the `gateway-extensions:get` command; for example:

```
apic gateway-extensions:get --scope org --org admin --gateway-service mygateway-service --availability-zone myavailabilityzone --server platform-api.myserver.com --output -
```



(the parameter setting `--output` - writes the details of the Gateway extension object to the command window. You can specify the name of an existing folder to have the details written to a `.yaml` file in that folder.)

For reference details of the `apic gateway-extensions` commands, see [apic gateway-extensions](#).

Note: You cannot upload more than one Gateway extension .zip file to the same Gateway service. If the Gateway service already has a Gateway extension, you must use the `apic gateway-extensions:delete` command to remove the extension from the Gateway service, add your user-defined policy files to the original .zip file, then upload it again.

## Results

---

The user-defined policy is deployed to the DataPower API Gateway, and is displayed in the palette on the assembly page in the API editor, from where it can be added to an API assembly flow.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Developer Portal: socialize your APIs

The Developer Portal is your one-stop-shop for sharing your APIs with application developers.

In addition to allowing application developers to find and subscribe to APIs, the Developer Portal provides additional features including forums, blogs, comments, and ratings, together with an administrative interface for customizing the Developer Portal to your brand and theme. See the following topics for more information.

### [Using the Developer Portal](#)

As well as enabling application developers to find and use your APIs, the Developer Portal also provides features such as API analytics, forums, blogs, and rating facilities.

### [Configuring the Developer Portal site](#)

Customize the look and feel of the Developer Portal site, and brand it for your organization.

### [Concepts in the Developer Portal](#)

Understand the various concepts and terminology that are referenced throughout the Developer Portal.

### [Getting started configuring the Developer Portal site](#)

A quick start guide to configuring your Developer Portal site.

### [Developer Portal tutorials](#)

Guided examples that take you through some of the most common user scenarios, as well as some of the more complex areas.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Using the Developer Portal

As well as enabling application developers to find and use your APIs, the Developer Portal also provides features such as API analytics, forums, blogs, and rating facilities.

The Developer Portal is the one-stop-shop for sharing APIs with application developers. After a Developer Portal has been enabled through the API Manager, and one or more API Products have been published, application developers can browse and use the APIs. Typically, a Developer Portal is customized to fit the corporate branding and design requirements of a particular organization. For more information about configuring a Developer Portal site, see [Configuring the Developer Portal site](#). However, when the Developer Portal is first enabled it can be used as is, for example for test and development purposes, or for internal use only.

The following topics provide information about the main API consumer concepts in the Developer Portal, and how an application developer can navigate the default site. However, bear in mind that the actual consumer experience might be very different depending on how the site is configured. For tutorials guiding you through enabling a Developer Portal site, and navigating and using APIs as an application developer, see [Tutorial: Creating the Developer Portal](#) and [Tutorial: Creating Accounts and Applications on the Developer Portal](#).

- [Exploring APIs and Products in the Developer Portal](#)

As an application developer, you can browse, test, and subscribe to APIs in the Developer Portal by exploring the available API Products.

- [Applications in the Developer Portal](#)

You can view, edit, configure, and create applications in the Developer Portal.

- [Analytics in the Developer Portal](#)

You can view analytics for APIs in the Developer Portal at the application and organization levels. This information is displayed in dashboard views that show the analytics metrics in the form of *visualizations* (represented as charts).

- [Consumer organizations in the Developer Portal](#)

Consumer organizations provide a way of grouping application developers together. You can add, remove, and configure Consumer organizations in the Developer Portal.

- [User accounts, passwords, and support in the Developer Portal](#)

You can change general elements such as user accounts and passwords in the Developer Portal.

## Related information

---

- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Exploring APIs and Products in the Developer Portal

As an application developer, you can browse, test, and subscribe to APIs in the Developer Portal by exploring the available API Products.

### Browsing available APIs

In API Connect API providers package their APIs into Products to offer one or more APIs to application developers. The providers then use Plans to control access to APIs and to manage API usage. Products are packages that contain both the APIs and the accompanying Plans.

From the Developer Portal home screen, click API Products to browse the available products. You can then select a Product to explore the available Plans and APIs further, including viewing API code snippets, and request and response examples.

Note: If you have an account on the Developer Portal, you can specify a default programming language for the code snippets to display in by editing the Code Snippet Language in your account settings.

### Subscribing to a Plan

A Plan is a collection of API operations or subsets of operations from one or more APIs. In order to subscribe to a Plan, you must have an account on the Developer Portal and have created an App. For more information, see [User accounts, passwords, and support in the Developer Portal](#) and [Applications in the Developer Portal](#). Rate limits specify how many requests an App is allowed to make during a specified time interval. A Plan can have a shared rate limit for all the operations, or each operation can have a different rate limit. In addition, Plans can have multiple rate limits set per Plan and per operation, at second, minute, hour, day, and week time intervals. Plans can also have burst limits to prevent usage spikes that might damage infrastructure. Multiple burst limits can be set per Plan, at second and minute time intervals.

After you have identified the Plan that you want to use, click Subscribe and then select the App that you want to use with this Plan. If the Plan is not restricted, you can use it immediately. If the Plan is restricted, the subscription is shown as Pending Approval, and you cannot use the requested Plan until the administrator approves your request.

### Testing an API

You can test an API in the Developer Portal by using the interactive API document test tool. If the API operation that you want to test doesn't require a client ID, then you can use the test tool without the need to log in. For more information, see [Testing an API by using the Developer Portal test tool](#).

### Calling an API in your application

When you have subscribed to a Plan and you begin coding your application, you must retrieve the operation URL in order to call the API. You can retrieve the operation URL from the API overview page. For more information, see [Calling an API](#).

- **Testing an API by using the Developer Portal test tool**

You can test the behavior of an API without the need to write any code by using the Developer Portal test tool. You provide the necessary API parameters within the test tool and click Invoke to see the response.

- **Calling an API**

When you have a selected a Plan and you begin coding your application, you must retrieve the operation URL to call the API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Testing an API by using the Developer Portal test tool

You can test the behavior of an API without the need to write any code by using the Developer Portal test tool. You provide the necessary API parameters within the test tool and click Invoke to see the response.

### Before you begin

To use the Developer Portal test tool with an API that requires an App client ID, you must first complete the following tasks:

- Create an App. For more information, see [Registering an application](#).
- Subscribe to a Plan that contains the API that you want to test. For more information, see [Exploring APIs and Products in the Developer Portal](#).

### About this task

The Developer Portal test tool is an interactive API document test tool. If the operation that you want to test does not require a client ID, then you can use Developer Portal test tool without the need to sign in. However, if the operation that you want to interact with requires a client ID, then you must sign in to the Developer Portal first.

Using the Developer Portal test tool is subject to the rate limit that is applied to an operation or Plan. For example, if an operation has a rate limit of 10 requests per minute and you invoke the operation, the number of requests that can be made is reduced to nine. The limitations are triggered every time that you click invoke within the minute interval. This caveat affects the quota of the App that is selected to use with the Developer Portal test tool, but it does not affect the quota of the other Apps that are using the same operation or Plan.

To test an API in the Developer Portal, the Allow this API to be tested check box must be selected in the API Manager. For more information, see [Creating API definitions](#).

#### Restriction:

- There are security mechanisms that prevent you using the Developer Portal test tool to test an Implicit or Authorization Code grant type in an OAuth provider API. Other grant types in the same OAuth provider API **can** be tested. If your OAuth provider allows those actions to function correctly from the test tool, you can disable

- this option. For more information, see [Disabling test tool restrictions](#).
- You cannot use the Developer Portal test tool with suspended applications.
  - You can only test an unenforced API if `testable=true`, and if the existing API implements CORS and is using HTTPS.

## Procedure

1. To use the Developer Portal test tool with an API that does not require the client ID of an App, complete the following steps:

- Click API Products.  
All of the APIs that can be used by application developers are displayed.
- Click the name of the API that you want to test.
- Select an operation and then select Try It.
- Click Try this operation.
- Supply the values for required headers or parameters.
- If the operation is secured with Basic Authentication, supply the credentials.
- Click Send Request.

The result is displayed in the Response Body field. You can continue to test different parameter values as necessary.

Note:

The first time that you click Try It, you might be presented with a security error. Copy the URL from the Request URL field, open it in a browser window, and accept the security certificate. You are not presented with the security error again.

Also, for performance reasons, if the payload is large, above 1500 DOM highlighted elements, you get the response, but it is not pretty printed.

2. To use the Developer Portal test tool with an API that does require the client ID of an App, complete the following steps:

- Sign in to the Developer Portal.
- If you have not already done so, you must create an App so that you can test an API that requires a client ID. For more information, see [Registering an application](#).
- Ensure that your App is subscribed to the Plan that contains the API that you want to test. For more information, see [Exploring APIs and Products in the Developer Portal](#).
- Click API Products.  
All of the APIs that can be used by application developers are displayed.
- Click the API that you want to test.
- Select the **Provide credentials for Client ID (API Key)** operation and then select Try It.  
The API Key Identification window displays.
- Use the Register an application in order to select a client ID drop-down list to select an App to test the API with.
- Click Save credentials.
- If a client secret is required, enter the value in the Client Secret field.
- Find the operation that you want to test, then click Try It.
- Supply the required parameters and values.
- If the operation is secured with Basic Authentication, supply the credentials.
- Click Send.

The result is displayed in the Response section. You can continue to test different field values as necessary.

Note: The first time that you click Invoke, you might be presented with a security error. Click the link that is provided to accept the security certificate. You are not presented with the security error again.

## Related tasks

- [Calling an API by using CORS](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Calling an API

When you have a selected a Plan and you begin coding your application, you must retrieve the operation URL to call the API.

## About this task

Be aware of the following points when you call APIs in IBM® API Connect:

- In the HTTP response message for status code 200 the reason phrase is replaced with OK.
- API error messages are displayed in English only.

## Procedure

To retrieve the operation URL, complete the following steps:

- Click API Products, then select a product.
- Click the API that you want to work with.  
The API overview page opens.
- Select the operation you need and then copy the endpoint.  
This is the URL that the application calls, and its structure is defined:

| Type of API | URL |
|-------------|-----|
|-------------|-----|

| Type of API | URL   |
|-------------|---|
| REST APIs   | <code>https://host/org/catalog/api/operation</code>                       |
| SOAP APIs   | <code>https://host/org/catalog/api</code> for all operations in the WSDL. |

where:

- *host* is the fully qualified host name of your gateway cluster.
- *org* is the URL path of your organization.
- *catalog* is the name of your Catalog.
- *api* is the name of your API.
- *operation* is the URL path of your operation.

4. Take note of any parameters, the request body, and the response body. Code your application to create the expected requests and handle the expected responses.

Depending on the Identify your application using setting for the API, you might have to provide a client ID, or client ID and client secret. To do this, complete the following steps:

- To find the client ID, complete the following steps:
  - Click Apps, then click the application name that you want to work with.
  - Select the Show check box for Client ID.  
The client ID is displayed.
  - Provide the client ID with the header parameter `&client_id=`  
For example, the URL used in the API might be:

```
https://host/org/catalog/api/quote?loanAmount=20000
```

but when you call it with a client ID of 1234, change the URL to:

```
https://host/org/catalog/api/quote?loanAmount=20000&client_id=1234
```

Note: API Gateway only A client ID is generated automatically by API Connect when you register an application. However, if you specify a customized client ID, by using the CLI or REST API, then the length must not exceed 512 bytes, otherwise the gateway will reject the API request and return a **401** error.

- The client secret is produced when you register an application. Provide the client secret with the query parameter `&client_secret=`. If you did not note the client secret when you registered the application, you must reset it; for information, see [Managing applications](#).

The client ID, or client ID and secret can be logged along with the URL. Web servers usually log the URL in their access logs, which would give away the client secret. If you do not want to expose your client ID or secret in the URL, complete the following steps.

- For the client ID, set the header, **X-IBM-Client-Id**, as part of the HTTP message that the application sends when it calls the API.  
An example URL statement might be:

```
curl --header "X-IBM-Client-Id: 1234" https://host/org/catalog/api/quote?loanAmount=20000
```

Note: API Gateway only A client ID is generated automatically by API Connect when you register an application. However, if you specify a customized client ID, by using the CLI or REST API, then the length must not exceed 512 bytes, otherwise the gateway will reject the API request and return a **401** error.

- For the client secret, set the header, **X-IBM-Client-Secret**, as part of the HTTP message that the application sends when it calls the API.  
For example, the URL would be:

```
https://host/org/catalog/api/quote?loanAmount=20000
```

and set the following HTTP headers:

```
X-IBM-Client-Id=1234
X-IBM-Client-Secret=ABCD
```

## What to do next

Monitor the API and application usage. For more information, see [Managing applications](#).

- [Calling an API by using CORS](#)  
CORS (cross origin resource sharing) is a technique that allows calls to be made from code that is running in a browser to a third-party server (such as APIs running on an API Connect Gateway). These calls are, by default, not allowed as per the same origin security policy that is applied to the browser sandbox. Without CORS support, web developers are required to use more complex techniques such as server-side proxies.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Calling an API by using CORS

CORS (cross origin resource sharing) is a technique that allows calls to be made from code that is running in a browser to a third-party server (such as APIs running on an API Connect Gateway). These calls are, by default, not allowed as per the same origin security policy that is applied to the browser sandbox. Without CORS support, web developers are required to use more complex techniques such as server-side proxies.

## About this task

API Connect Gateway servers support CORS to make it as easy as possible for web developers to use APIs within their web applications.  
CORS is supported in the following browsers:

- Chrome 3+

- Firefox 3.5+
- Internet Explorer V11, or later
- Opera 12+
- Safari 4+

A CORS enabled browser automatically sends either a simple CORS request, consisting of the original request with the addition of the **Origin** header, or a preflight request followed by a simple CORS request.

An example CORS preflight request is as follows:

```
OPTIONS /org/env/api/resource HTTP/1.1
User-Agent: useragent details
Access-Control-Request-Method: GET
Access-Control-Request-Headers: header names
Host: x.xx.xxx.xx
Origin: https://example.com
Accept: */*
```

You do not need to create CORS requests yourself, other than for testing or troubleshooting purposes.

A CORS response is received from the gateway; for example:

```
HTTP/1.1 200 OK
X-Backside-Transport: FAIL FAIL
Connection: Keep-Alive
Transfer-Encoding: chunked
Access-Control-Allow-Origin: https://example.com
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: accept, accept-language, content-type, x-ibm-client-id
Access-Control-Allow-Methods: methods allowed on the resource
Vary: Origin
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Applications in the Developer Portal

You can view, edit, configure, and create applications in the Developer Portal.

Before you can subscribe to a Plan that contains the API or APIs that you want to use, you must first register an application by creating an App. You can create more than one App, and these Apps can be edited and deleted as required.

To learn about applications in the Developer Portal, use the following topic links.

- [Registering an application](#)  
Before you can use an API, you must register your application to the Developer Portal.
- [Managing applications](#)  
You can show or reset a client ID for an application, verify or reset a client secret, and view the details of the APIs in the application. You can also unsubscribe from a Plan that the application is subscribed to.
- [Editing an application](#)  
You can edit applications in the Developer Portal. If a user has the correct permissions, they can edit the content of any application in their organization.
- [Deleting an application](#)  
You can delete applications in the Developer Portal.
- [Changing an application image](#)  
You can change the image for an application in the Developer Portal.
- [Verifying an application client secret](#)  
You can verify an application client secret in the Developer Portal

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Registering an application

Before you can use an API, you must register your application to the Developer Portal.

### About this task

---

When you register an application, you are provided with a client ID and client secret for the application. You must supply the client ID when you call an API that requires you to identify your application by using a client ID, or a client ID and client secret.

You can, optionally, add further client ID/client secret pairs to an application, any of which can be used to identify the application when calling an API.

---

## Procedure

To register an application, complete the following steps:

1. Click Apps.  
The Apps page opens.
2. Click Create new App.
3. Optional: Specify a URL in the Application OAuth Redirect URL(s) field.  
You can specify multiple OAuth redirect URLs by clicking Add another item. If only one redirect URL is specified, and the application does not provide the `redirect_uri` in the OAuth request, then API Connect automatically uses the one redirect URL specified. However, if more than one redirect URL is specified, then the application must provide the `redirect_uri` in the OAuth request, or the OAuth request is rejected.
4. Click Submit.  
Your application is displayed.
5. The client secret is hidden; click the Show check boxes, make a note of your client secret because it is displayed only once.  
You must supply the client secret when you call an API that requires you to identify your application by using a client ID and client secret.  
Note: The client secret cannot be retrieved. If you forget it, you must reset it.
6. Optional: In the Certificate field, paste the public X509 certificate for your application, in PEM format.  
The Certificate field is available only if the APIs that have been published to the Developer Portal include at least one API that has been secured with TLS mutual authentication. You must complete this field if you want to call an API that has been secured with TLS mutual authentication. For more information, see [Composing a REST API definition](#).  
  
These certificates are sent to the management server and to the gateway, and are used for certificate verification when the application calls an API that has been secured with TLS mutual authentication.  
  
Important: When you paste the certificate, ensure that it contains no line breaks, and that the (-----BEGIN CERTIFICATE-----) prefix and (-----END CERTIFICATE-----) suffix are removed.
7. Optional: The client ID is hidden, to display the client ID for your application, click Subscriptions, then select the Show check box for Client ID.  
The client ID is displayed and can be hidden again by clearing the check box.
8. Optional: To verify your client secret, click Subscriptions, then click Verify, enter your client secret in the Secret field, then click Submit.  
You have confirmed whether your client secret is correct or incorrect.
9. To add an additional client ID and client secret to the application, complete the following steps:
  - a. Click Subscriptions.
  - b. Click Add alongside Credentials.  
The "Add new application credentials" window opens.
  - c. Enter a name and an optional summary, and click Submit.
  - d. Select the Show Client ID or Show Client Secret check box to display the client ID or client secret for the new credentials.
  - e. To add a description to a set of client credentials, or to change the current description, click Update alongside the required credentials.
  - f. To remove a set of client credentials from the application, click Delete alongside the required credentials.If you add additional credentials to an application, any of the associated client ID/client secret pairs can be used to identify the application when calling an API.  
Note: If you add two or more sets of client credentials to an application, OAuth tokens are not shared between them; each client credential set uses a different OAuth token.
10. To add an image, click Upload image from the application overflow menu.  
A new window opens; click Browse, select an image from your directory, and click Submit.
11. Optional: To change the URL that authenticated OAuth flows for this application should be redirected to, click Edit from the application overflow menu, then enter a URL in the Application OAuth Redirect URL(s) field.
12. Optional: To change the application name or description, click the Edit from the application overflow menu.

---

## Results

Your application is registered.

---

## What to do next

Select a Plan to use with your application, see [Exploring APIs and Products in the Developer Portal](#).

---

## Related information

- [IBM API Connect overview](#)
- [Creating an API key security definition](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing applications

You can show or reset a client ID for an application, verify or reset a client secret, and view the details of the APIs in the application. You can also unsubscribe from a Plan that the application is subscribed to.

---

## About this task

On the application details page, you can display and reset a client ID and client secret. You might want to reset a client secret if it is compromised or you forget it. A client secret is only displayed once, so after you reveal it for the first time at a time of your choosing, you can use this window to verify that what you have is correct or reset it if necessary.

You can, optionally, add further client ID/client secret pairs to an application, any of which can be used to identify the application when calling an API.

## Procedure

---

To manage an application, complete the following steps:

1. Click Apps.  
The Apps page opens.
2. Select the application that you want to work with from the displayed list.
3. Click Subscriptions.
4. The client ID is hidden; to display the client ID, select the Show check box. The client ID is displayed, and can be hidden again by clearing the Show check box.
5. To reset your client ID, click Reset.  
Warning: API calls made by the application with the existing client ID will fail.
6. If the client secret is compromised or you forget it, click Reset to generate a new value for the client secret.
7. If you are unsure whether the value that you have for the client secret is correct, you can click Verify, enter the value, and click Submit to confirm whether or not it is correct.
8. To add an additional client ID and client secret to the application, complete the following steps:
  - a. Click Subscriptions.
  - b. Click Add alongside Credentials.  
The "Add new application credentials" window opens.
  - c. Enter a name and an optional summary, and click Submit.
  - d. Select the Show Client ID or Show Client Secret check box to display the client ID or client secret for the new credentials.
  - e. To add a description to a set of client credentials, or to change the current description, click Update alongside the required credentials.
  - f. To remove a set of client credentials from the application, click Delete alongside the required credentials.If you add additional credentials to an application, any of the associated client ID/client secret pairs can be used to identify the application when calling an API.  
Note: If you add two or more sets of client credentials to an application, OAuth tokens are not shared between them; each client credential set uses a different OAuth token.
9. To view more details about an individual API, click the API name.  
The operations are listed.
10. To unsubscribe from a Plan, click Unsubscribe alongside the Plan name.

## Related information

---

- [IBM API Connect overview](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing an application

You can edit applications in the Developer Portal. If a user has the correct permissions, they can edit the content of any application in their organization.

## Procedure

---

1. From the Developer Portal home page, click Apps.
2. Select the target application.
3. Click Edit from the application overflow menu.
4. Make any required changes, then click Submit.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting an application

You can delete applications in the Developer Portal.

## Procedure

---

1. From the Developer Portal home page, click Apps.
2. Select the required application.
3. Click Delete from the application overflow menu.
4. Click Delete again to confirm.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing an application image

You can change the image for an application in the Developer Portal.

### Procedure

---

1. From the Developer Portal home page, click Apps.
2. Select the required application.
3. Click Upload image from the application overflow menu.
4. Browse for the target image, then click Submit.  
You can remove the image by clicking Remove.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Verifying an application client secret

You can verify an application client secret in the Developer Portal

### Procedure

---

1. From the Developer Portal home page, click Apps.
2. Select the target application.
3. Click Subscriptions.
4. Click Verify for the client secret.
5. Enter your secret string and click Submit.  
You receive a notification saying whether or not the secret that you entered is correct.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Analytics in the Developer Portal

You can view analytics for APIs in the Developer Portal at the application and organization levels. This information is displayed in dashboard views that show the analytics metrics in the form of *visualizations* (represented as charts).

Restriction:

- If the Catalog that is associated with this Developer Portal has two or more gateway services enabled then, for analytics data to be available, the gateway services must all be associated with the **same** analytics service. If the set of associated analytics services for the gateway services includes two or more different analytics services then the Developer Portal does **not** display analytics data. An analytics service is associated with a gateway service in the Cloud Manager user interface; see [Associating an analytics service with a gateway service](#).
- If the visualizations in your dashboard views display no data or show data only for a time period in the past, a possible reason might be that access to analytics data within API Connect has been switched off. Contact your API provider for confirmation.

Use the following topic links to learn how to access the various analytics available to you in the Developer Portal:

- [Application analytics in the Developer Portal](#)  
From the Developer Portal, you can view interactive analytic information about all of the APIs used by an application.
- [Organization analytics in the Developer Portal](#)  
From the Developer Portal, you can view interactive analytic information about all of the APIs within an organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Application analytics in the Developer Portal

From the Developer Portal, you can view interactive analytic information about all of the APIs used by an application.

### About this task

---



All members of the consumer organization can view analytics information relating to APIs used by an application.

## Procedure

---

1. Click Apps from the Developer Portal dashboard.
2. Click the name of the application for which you want to view analytic information.  
From the Dashboard tab, you can see your application analytics including, API Stats, Total Calls, and Total Errors.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Organization analytics in the Developer Portal

From the Developer Portal, you can view interactive analytic information about all of the APIs within an organization.

### About this task

---

All members of the consumer organization can view analytics information relating to the APIs that are used within an organization.

## Procedure

---

1. Click your userName from the Developer Portal dashboard.
2. From the drop-down menu, select My organization.
3. Click the Analytics tab.  
From the Analytics tab, you can see your organization analytics including, API Stats, Total Calls, and Total Errors.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Consumer organizations in the Developer Portal

Consumer organizations provide a way of grouping application developers together. You can add, remove, and configure Consumer organizations in the Developer Portal.

Developer Portal users must be a member of at least one Consumer organization. A Consumer organization can have multiple members and multiple applications. Members must register an application before they can subscribe to an API. A Consumer organization owns all the applications that are created within it. The members do not own any applications. As long as there is one member of a Consumer organization, all applications continue unchanged.

Note: Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in: **A user already exists with this email address.**

To learn more about Consumer organizations in the Developer Portal, see the following topics.

- [Adding a Consumer organization from within the Developer Portal](#)  
You can create a new Consumer organization from within your user account in the Developer Portal. The new Consumer organization appears also in the API Manager UI after it is created.
- [Editing the name of a Consumer organization](#)  
You can edit the name of your Consumer organization from within the Developer Portal.
- [Switching Consumer organizations](#)  
You can switch from one Consumer organization to another in the Developer Portal.
- [Adding users to a Consumer organization](#)  
If the API Provider organization administrator has granted you permission to collaborate, you can invite users to join your Consumer organization. Those organization members can then access the Developer Portal and use the APIs that have been made available to the Consumer organization.
- [Removing a user from a Consumer organization](#)  
You can remove a developer from a Consumer organization in the Developer Portal.
- [Changing the ownership of a Consumer organization](#)  
You can change the ownership of a Consumer organization in the Developer Portal.
- [Deleting a Consumer organization](#)  
You can delete a Consumer organization in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding a Consumer organization from within the Developer Portal

You can create a new Consumer organization from within your user account in the Developer Portal. The new Consumer organization appears also in the API Manager UI after it is created.

### Procedure

---

1. Sign in to your user account in the Developer Portal.
2. Click the arrow next to your Consumer organization name on the upper-right corner of the Developer Portal home page.
3. Select Create organization from the drop-down menu.
4. In the Organization Title field, type the name of the new Consumer organization, then click Submit.

### Results

---

Your new Consumer organization is created, and you can switch to it by clicking on its name in the Organization drop-down menu on the upper right of the home page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing the name of a Consumer organization

You can edit the name of your Consumer organization from within the Developer Portal.

### Procedure

---

1. Sign in to your user account in the Developer Portal.
2. Click the arrow next to your Consumer organization name on the upper-right corner of the Developer Portal home page, and click My organization.
3. In the Manage tab, click the vertical ellipsis Manage organization icon, and click Edit organization.
4. Enter the new name for your Consumer organization in the Organization Title field, and then click Submit.  
You changed the name of your Consumer organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Switching Consumer organizations

You can switch from one Consumer organization to another in the Developer Portal.

### Procedure

---

1. If you are a member of more than one Consumer organization, click the menu of Consumer organizations in the upper-right corner of the Developer Portal home page.
2. Select the Consumer organization that you want to switch to, and ensure that any applications you register are registered to the correct catalog.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding users to a Consumer organization

If the API Provider organization administrator has granted you permission to collaborate, you can invite users to join your Consumer organization. Those organization members can then access the Developer Portal and use the APIs that have been made available to the Consumer organization.

### About this task

---

Depending on the permissions enabled by the Provider organization administrator, members can view applications and application activity, create and edit applications, manage client keys and subscribe to API Plans, or perform all activities.

Note: Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning

correctly, and returning the following error message when trying to log in: **A user already exists with this email address.**

## Procedure

---

To invite users to join your Consumer organization, complete the following steps.

Note: The ability to invite users is only visible if the API Provider organization administrator has granted you permission to collaborate.

1. Sign in to your user account in the Developer Portal.
2. Click the arrow next to your Consumer organization name on the upper-right corner of the Developer Portal home page, and click My organization. The organization page is displayed.
3. On the Manage tab, select Invite.
4. Enter the email address of the new user in the Email field.
5. Under the Assign Roles heading, use the radio buttons to select a role for the new user, the roles are:
  - **Administrator**-- Can administer the Consumer organization, including invite new members, and can create, and edit applications, manage client keys, and subscribe to API Plans.
  - **Developer**-- Can create and edit applications, manage client keys, and subscribe to API Plans.
  - **Viewer**-- Can only view applications and application activity.

For full details of the permissions that are assigned to the Developer Portal roles, see [API Connect user roles](#).

6. Click Submit.  
An invitation is sent to the new user.

## Results

---

The user is added to the Consumer organization list of members, and they are sent an email inviting them to join the organization. The user must click the link that is provided in the email to activate their account, and complete the setup.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing a user from a Consumer organization

You can remove a developer from a Consumer organization in the Developer Portal.

### Before you begin

---

You must be the owner of your Consumer organization and have previously added the developer you wish to remove from the organization. Note that removing a user from a Consumer organization does not delete their user account, it only removes them from that organization.

## Procedure

---

1. Click your user name in the Developer Portal home page.
2. Click My organization in the menu.
3. Click Remove user for the user you want to remove.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the ownership of a Consumer organization

You can change the ownership of a Consumer organization in the Developer Portal.

### Before you begin

---

You can change the ownership of a Consumer organization if you are the owner of that organization, or if you have the administrator role in that organization. The user that you transfer ownership to must already be a member of the organization. To have an account on a Developer Portal, you must be a member of at least one Consumer organization.

### About this task

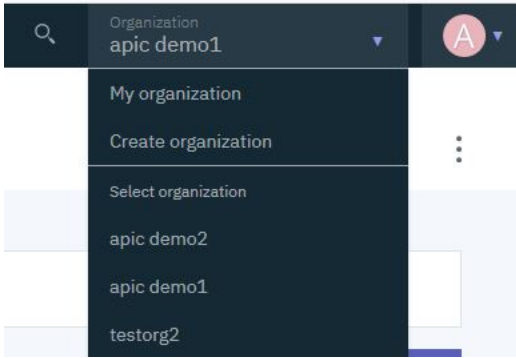
---

Transferring the ownership of a Consumer organization to another organization member, changes your role in that organization to App Developer.


## Procedure

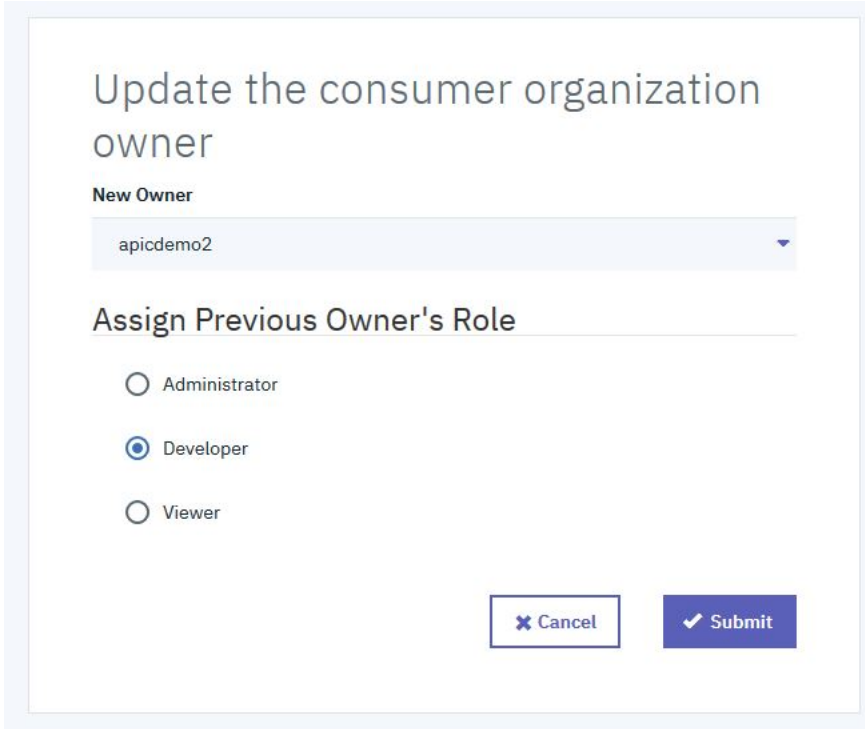
---

1. Click your Organization name in the Developer Portal home page, and select the organization whose owner you want to change by using the drop-down arrow.



2. Click your Organization name in the Developer Portal home page, and select My organization.

3. Click the options icon, , and select Change ownership.



4. Select the **New Owner** by using the drop-down arrow.

5. Use the radio buttons to **Assign Previous Owner's Role**, then click Submit.

## Results

You successfully transferred ownership of a Consumer organization in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting a Consumer organization

You can delete a Consumer organization in the Developer Portal.

### Before you begin

You can delete a Consumer organization only if you are the owner of that organization. To have an account on a Developer Portal, you must be a member of at least one Consumer organization. Therefore, if you are a member of only one Consumer organization, to delete that organization you must first either join or create another Consumer organization, or you can delete your whole developer account. For more information, see [Deleting your developer account](#).

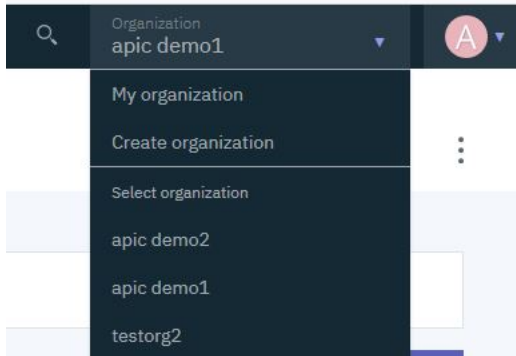
### About this task


Deleting a Consumer organization in the Developer Portal permanently removes access to the organization, and all of its applications and subscriptions, for all members of the organization. Consider carefully the impact on any members of your Consumer organizations before you delete the organization. Any users that were members of only the deleted organization, must create a new Consumer organization when they next logon to the Developer Portal.

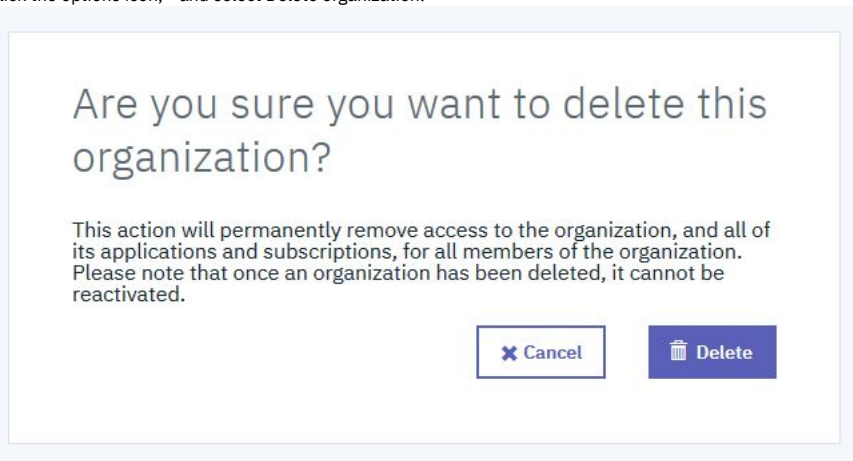
Important: When an organization is deleted, it cannot be reactivated. You might want to consider changing the ownership of a Consumer organization, rather than deleting it. For more information, see [Changing the ownership of a Consumer organization](#).

## Procedure

1. Click your Organization name in the Developer Portal home page, and select the organization that you want to delete by using the drop-down arrow. You must be in more than one Consumer organization.



2. Click your Organization name in the Developer Portal home page, and select My organization.
3. Click the options icon, , and select Delete organization.



4. Click Delete.

## Results

You permanently deleted a Consumer organization in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## User accounts, passwords, and support in the Developer Portal

You can change general elements such as user accounts and passwords in the Developer Portal.

Note: Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in: **A user already exists with this email address.**

To learn about general user features of the Developer Portal, see the following topics:

- [Creating a new developer account](#)  
You can create new developer accounts, and their associated Consumer organizations, in the Developer Portal.
- [Changing your account settings](#)  
You can edit your user account settings in the Developer Portal, such as changing your name, password, time zone, and language settings.
- [Deleting your developer account](#)  
You can delete your account in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a new developer account

You can create new developer accounts, and their associated Consumer organizations, in the Developer Portal.

### Before you begin

---

You need to provide an email address to create a new account. That email address must be a valid email address, and the domain that is used is required to have a valid MX record that is determined by DNS. You can disable the security module that checks that the email domain has a DNS record.

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Extend > Disable** module.
4. In the search bar enter **check\_dns**, then select the **check\_dns** module and click **Disable**.
5. Click **Disable** to confirm that you want to disable the module.

### About this task

---

In order to use the Developer Portal, you must have a developer account, and be a member of at least one Consumer organization. When you create a developer account, a Consumer organization is created at the same time. You can be an owner, and a member of, multiple Consumer organizations. For more information about Consumer organizations, see [Consumer organizations in the Developer Portal](#).

**Note:** Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in: **A user already exists with this email address**.

### Procedure

---

1. From the Developer Portal home page, click **Create account**.
2. Populate the available fields with the details of the developer account and Consumer organization that you want to create.
3. Click **Sign up**.

The email address that you provided for the new developer account is sent an email that contains an activation link for the new developer account. Use this link to activate your new account.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing your account settings

You can edit your user account settings in the Developer Portal, such as changing your name, password, time zone, and language settings.

### About this task

---

You can edit various details about your user account, for example, change your password, update your code snippet language, change your time zone, select your preferred site language, and select an avatar.

### Procedure

---

1. Click your user name in the Developer Portal home page, and click **My account**.
2. To update your password, click **Change Password**, and update the fields as required.
3. To update other elements of your account, such as your name, time zone, and language, click **Edit** and update the fields as required.
4. Click **Save** to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting your developer account

You can delete your account in the Developer Portal.

## Before you begin

---

You must be the owner of the developer account that you want to delete, and you must not own more than one Consumer organization. If you do own more than one Consumer organization, you must either change the ownership of those organizations, or delete them, before you can delete your account. If you own only one Consumer organization, when you delete your account the Consumer organization is deleted at the same time as your account.

## About this task

---

When you delete your account, any Consumer organizations that you own must also be deleted or transferred to new owners. If there are other members in a Consumer organization that you delete, these members will no longer have access to that organization, or any of its applications and subscriptions. Consider carefully the impact on any members of your Consumer organizations before you delete your account. For more information, see [Deleting a Consumer organization](#).

Important: When an organization has been deleted, it cannot be reactivated. You might want to consider changing the ownership of your Consumer organizations before you delete your account. For more information, see [Changing the ownership of a Consumer organization](#).

## Procedure

---

1. Click your user name in the Developer Portal home page, and select My account.
2. Click the Edit tab.
3. Click Delete account at the bottom of the page.
4. Click Delete to confirm deletion of the account.

This action deletes your developer account and, if you own a Consumer organization, this organization is also deleted.

## Results

---

You successfully deleted your developer account in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the Developer Portal site

Customize the look and feel of your Developer Portal site, and brand it for your organization.

The Developer Portal is based on the open source Drupal content management system, and consequently it is almost infinitely customizable. What an individual user is able to configure, depends on the role the user has been assigned, and the associated permissions of that role. A user can be assigned one of four Developer Portal roles:

### Authenticated user

An authenticated user is, typically, an application developer, and can perform all of the actions that are described in the [Using the Developer Portal](#) section.

### Forum Moderator, Content Author, or Administrator

In addition to performing all of the tasks that an Authenticated user role can, a Forum Moderator, Content Author, or Administrator role have additional permissions related to configuring the Developer Portal site. For more information, see [Working with roles in the Developer Portal](#).

Note: By default, the Developer Portal administration operations are available only if you have Administrator access.

Before you start to configure your Developer Portal site, you should familiarize yourself with the various concepts and terminology that is referenced throughout the Developer Portal. See [Concepts in the Developer Portal](#).

When you are ready to start configuring your Developer Portal site, see [Getting started configuring the Developer Portal site](#) for a quick start guide on customization. There is also a tutorial that you can follow that takes you through an example customization scenario; see [Tutorial: Creating a custom theme for the Developer Portal](#).

In addition to the aforementioned options, Developer Portal configuration operations are described in the following sections:

### [Appearance](#)

Control and configure the general appearance of your Developer Portal site by setting and configuring a default theme.

Important: Editing API Connect themes, modules, or Drupal core on the filesystem is not permitted or supported. For more information, see [Creating a sub-theme](#) and [Extend](#).

### [Content](#)

Control multiple content entities in the Developer Portal, including the customization and restriction of specific content.

### [Structure](#)

Control the layout of the Developer Portal, such as configuring the front page, adding and changing blocks, and displaying Products and APIs in categories.

### [Configuration](#)

As an administrator, you can control multiple aspects of the Developer Portal configuration, including managing security and general configuration tasks.

### [People](#)

You can manage and customize the user experience in the Developer Portal, by altering permissions and assigning roles to specific users.

### [Forums](#)

Create and control forums in the Developer Portal.

### [Reports](#)

You can view reports about the general configuration and status of your Developer Portal site.

### [Extend](#)

You can extend the Developer Portal by creating and installing custom modules, or installing additional Drupal 9 contributed modules.

---

## Related information

- [🌐 Drupal website](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Concepts in the Developer Portal

The Developer Portal is built on Drupal 9, an open source content management technology. A good understanding of Drupal 8 concepts and terminology enhances your ability to work with the Developer Portal.

In Drupal, most content on the website is treated as variations on the concept of a **node**. Node types, such as static pages, blog posts, and news items, are all stored in the same way. The navigation structure of the site is designed separately by editing **menus**, **views**, and **blocks**. Finally, the **theme** of the system controls the overall appearance of the site to visitors. Access to these depends on the permissions that your role possesses. These are explained more fully in the following sections.

## Nodes

A node is a set of individual related content such as a page, a poll, an article, a forum topic, or a blog post. For example, when you create a blog post, and define the body text, you also define its title, content, author link, creation date, and taxonomy. Some of these elements are shown by the theme layer when the node is displayed, and some elements are metadata that control where and when the node is displayed at all. Each set of content in a Developer Portal site is a node. You can apply new features or changes to all content of a single node type.

## Fields

A field is somewhere that you can add extra metadata to a node. The following types of data are examples of fields:

- Title
- Body
- Comment body
- Tags
- Image

You can create different types of fields, and they can be defined at the content-type level for content items and comments, at the vocabulary level for taxonomy terms, and at the site level for user accounts. Field types are defined by modules, and managed by the Field module. In addition, other modules might enable fields to be defined for their data.

## Content types

A content type is a predefined collection of fields. Content types define the default fields that content editors use to add content in a Developer Portal site, and help structure the authoring and developing of content. Content types can be displayed in the Developer Portal. You can control which content types, and the order and format that they are displayed, in the Developer Portal.

## Themes

You can use themes to control the overall appearance of your Developer Portal site. You can modify the appearance of the theme in following ways:

1. Extend the code of an existing theme.
2. Identify and use a theme that is provided by the Drupal community or a third-party website, and modify the theme settings.
3. Create a complete custom theme from scratch.

However, you cannot directly edit the API Connect theme as editing is not supported. Any edited versions of these files are overwritten when a fix pack or iFix is installed. So, the way to create a custom theme is to create a custom subtheme of the standard API Connect theme that the Developer Portal uses by default. A subtheme inherits all of the settings of its parent theme, apart from the settings that are overridden. For more information, see [Creating a sub-theme](#).

Most themes use the Twig template engine for PHP.

## Regions

The specific areas of a Developer Portal site in which content can be placed. Regions are customized and styled in the theme.

## Blocks

Blocks are boxes of content that can be displayed in regions on your Developer Portal page. Blocks can be made available to your Developer Portal site by enabling specific modules. After a block is created, its appearance, shape, size, and position can be modified. You can also define which Developer Portal page or pages blocks appear on. Some modules can provide multiple blocks when they are enabled, while other modules might not define new blocks. For more information, see [Adding and changing the blocks displayed on Developer Portal pages](#).

## Modules

Modules are similar to the concept of plug-ins, in that they extend the core functions of the Developer Portal site. A set of modules is implemented by default with the Drupal core, and there are extra modules that can be enabled to extend the default functions. You can find and add more modules to your Developer Portal site from the internet. For more information, see [Extend](#).



## Views

---

You can use views to manipulate the content that is displayed on a page, block, and other visual elements in the Developer Portal site. Views can be used along with content types to format the appearance of your site to your specification. For more information, see [Using the Views module in the Developer Portal](#).

## Pages

---

Pages are used by the features that can modify the appearance of the Developer Portal to customize the appearance of your Developer Portal site. Pages can be customized to be as specific as you require, and can be configured to satisfy the context of the situation that they are used in.

## Users

---

After you log in to the Developer Portal site, you have a user record in the Developer Portal database. User ID 1 is always reserved for the administrator account. Regardless of whether other user accounts are using remote authentication such as LDAP, the administrator account is always a local account to enable administration of the Developer Portal site. For more information, see [People](#).

## Roles

---

A role is a collection of permissions that define the actions that a user can do in the Developer Portal site. Users can be given one or more roles. By default, a user that is logged in to the Developer Portal is in the Authenticated role. Other roles that can be assigned to a user include:

- Administrator
- Forum Moderator
- Content Author

You can also create new custom roles. For more information, see [Working with roles in the Developer Portal](#).

## Permissions

---

Permissions define the actions that a user can or cannot do in a Developer Portal site. Permissions are additive. If a permission that enables a user to do an action is not assigned, the user cannot do the action. If a user has multiple roles, and any of them contain a specific permission, the user is able to do that action. There are also permissions that have security implications, and you are recommended to assign those permissions to trusted roles. For more information, see [Controlling access to Developer Portal content](#).

## Templates

---

Template files define how the output of a particular component can look. They are formatted as Twig template files. The following are examples of where you can use templates:

- `html.html.twig` - the template for Developer Portal HTML pages
- `page.html.twig` - the body of Developer Portal HTML page
- `node.html.twig` - the template for all of the content nodes
- `comment.html.twig` - the template for all of the comment nodes
- `search-result.html.twig` - the template for search results
- `node--product.html.twig` - the template for nodes that use the Product content type
- `node--product--teaser.html.twig` - the template for the previews for the Product content type

For more information about overriding templates in the Developer Portal, see [Applying a modified content type template](#).

## Related information

---

- [Understanding Drupal](#)
- [Theming Drupal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Getting started configuring the Developer Portal site

A quick start guide to the main tasks involved in configuring the Developer Portal to your brand or theme.

The Developer Portal is based on the open source Drupal 9 content management software, and consequently is almost completely customizable. The following sections take you through the most common tasks to get your Developer Portal branded to your organizational requirements.

Important:

All of the customizations that are detailed in this topic are preserved when you update to new versions of API Connect. Drupal updates, as well as new functionality, defect fixes, and security updates, are made available through API Connect update packages as determined by IBM. These update packages can be applied by upgrading your deployment; see the upgrade instructions for your platform in [Installing and maintaining your IBM API Connect cloud](#). To find out what update packages are available, select your product on [IBM Fix Central](#).

Directly editing any API Connect themes, modules, or Drupal core on the file system is not permitted or supported, as edited versions of these files are overwritten when an update package is installed. Customizations must be done by using the proper mechanisms as described in this topic, such as custom modules and themes, and not by

directly editing the source.

For more information about the API Connect support policy, see [IBM® API Connect Support Lifecycle Policy](#).

- [Create a custom theme](#)
- [Changing the page layout](#)
- [Changing the front page](#)
- [Advanced theme development](#)
- [Custom modules](#)
- [What to do next](#)

---

## Create a custom theme

If you want your Developer Portal to have your corporate branding and style, you need to create a custom theme. Themes are composed primarily of Cascading Style Sheets (CSS) files, although they can include much more; see [Advanced theme development](#).

Directly editing the API Connect theme is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed. The way to create a custom theme is to create a custom sub-theme of the standard API Connect theme that the Developer Portal uses by default. A sub-theme inherits the parent theme's resources, and this means that your custom sub-theme CSS file needs to contain only the changes or overrides that you want to make from the default theme. The CSS file can contain as little or as many updates as you like.

For more information, see [Creating a sub-theme](#). You can also follow a tutorial that takes you through how to use the theme generator, customize a theme, and install a custom theme; see [Tutorial: Creating a custom theme for the Developer Portal](#).

While experience with Drupal is beneficial, the only skill you really need to create a sub-theme is CSS.

---

## Changing the page layout

While the theme defines *regions* of the page, such as a footer, or sidebar first, the page layout is defined by configuring which blocks appear in which regions. Blocks are controlled by using the Structure > Block layout options on the Developer Portal administrator dashboard. You can then further limit those blocks to show only on specific pages, or show on all pages except certain ones, to show only for certain roles, or for certain languages, and more.

For information about how to configure the page layout, see [Adding and changing the blocks displayed on Developer Portal pages](#).

Changing the page layout is done entirely in the Developer Portal UI as an Administrator.

---

## Changing the front page

The front page is a special case and does not use the blocks system that is mentioned previously. Instead, you configure it from the Structure > Pages options on the Developer Portal administrator dashboard.

For information about how to configure the front page, see [Configuring the front page](#).

Further options that you might want to consider when designing your front page include:

- Change the banner block content or image: [Changing the front page banner block](#).
- Display featured API Products by creating a featured content block: [Changing the front page Featured Content block](#).
- Create custom blocks with your own HTML content: [Adding and changing the blocks displayed on Developer Portal pages](#).
- Set up the social block to include your organization Twitter feed: [Integrating Twitter data into the social block](#).
- Add a carousel to the front page to display a series of images: [Implementing an image carousel](#).

Again, changing the front page is done entirely in the Developer Portal UI as an Administrator.

---

## Advanced theme development

The DOM structure for specific blocks and parts of the page is controlled by templates. These templates are Twig files that define the actual HTML output for a particular piece of content. You can override a Twig template by copying the original template into the templates folder of your custom sub-theme, and then editing the template to your specification. The Developer Portal then uses your template in preference to the original one. For more information, see [Applying a modified content type template](#).

Modifying Twig templates allows very fine-grained control over the structure of pages. However, it also means you might miss new features and defect fixes that are made to the original templates, as your Developer Portal is using your overrides instead of the originals. So, if you override a template, it is your responsibility to check the templates in the latest API Connect releases, and to ensure that any equivalent changes that are needed to your overrides to maintain functional equivalence are made.

If you're developing custom templates, knowledge of Twig templating language, Drupal, HTML, and CSS is recommended.

---

## Custom modules

If you need to modify the functional behavior of the Developer Portal, then you can create a custom module to do so. Drupal has an extensible programmatic API that is built around a system of *hooks*. By using these hooks in your custom modules you can change what happens when, for example, forms are displayed, forms are submitted, or make extra variables available to custom templates, and other options.

For more information on writing custom modules, see [Extend](#). You can also follow a tutorial that includes using a custom module, see [Tutorial: Adding validation to a field on the sign-up form](#).

Custom modules are written in PHP, and so Drupal experience and PHP knowledge are very beneficial.

---

## What to do next

There are an almost infinite number of configuration options available to you on the Developer Portal. Explore the following sections to find out more:

- [Appearance](#)
- [Content](#)
- [Structure](#)
- [Configuration](#)
- [People](#)
- [Forums](#)
- [Reports](#)
- [Extend](#)
- [Developer Portal tutorials](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Appearance

Control and configure the general appearance of your Developer Portal site by setting and configuring a default theme.

To learn how to control the appearance of your Developer Portal site, use the following topic links.

- [Controlling general appearance elements in the Developer Portal](#)  
You can control general appearance aspects of the Developer Portal.
- [Adding custom JavaScript to a custom theme](#)  
To increase customization of the Developer Portal, you can add custom JavaScript to custom themes that you install.
- [Applying a modified content type template](#)  
You can apply a modified content type template to override a default content type template in the Developer Portal. For example, you can override the product content type template to change its layout.
- [Creating a sub-theme](#)  
Create a custom theme for your Developer Portal site by generating and configuring a sub-theme. A sub-theme is a theme that inherits all the resources of a specified parent theme. You can then override specific resources to configure your required customizations.
- [Installing additional themes](#)  
You can install additional themes in the Developer Portal.
- [Deleting themes](#)  
You can delete a theme from your Developer Portal site.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Controlling general appearance elements in the Developer Portal

You can control general appearance aspects of the Developer Portal.

To learn how to control general aspects of the Developer Portal appearance, use the following topic links:

- [Changing the shortcut icon](#)  
You can change the shortcut icon, or favicon, in the Developer Portal.
- [Changing the site email address](#)  
You can change the site email address in the Developer Portal.
- [Changing the site logo](#)  
You can change the site logo for your Developer Portal.
- [Changing the site name or site slogan](#)  
You can change the site name, or the site slogan, for your Developer Portal.
- [Changing the visibility of site branding](#)  
You can toggle the visibility of your site branding elements for your Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the shortcut icon

You can change the shortcut icon, or favicon, in the Developer Portal.

---

### Before you begin

You must have administrator access to complete this task.

## About this task

---

The shortcut icon, or favicon, is displayed in the address bar or bookmarks of most browsers. Drupal provides a default favicon, the water drop logo, but if you want to make your Developer Portal stand out, you should provide your own. The default Drupal favicon is 32 pixels high by 32 pixels wide. As a favicon is displayed only in the address bar and favorites (bookmarks) list, any favicon that you create should be similarly small.

## Procedure

---

To change the shortcut icon, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
  2. Click Appearance in the administrator dashboard.
  3. Click Settings for the default theme.
  4. Click Favicon in the Override Global Settings section.
  5. Deselect the Use the favicon supplied by the theme check box.
  6. Provide a path to a custom icon on the server by entering it in the Path to custom icon field. Alternatively, you can browse for, and upload, a favicon image under the Upload favicon image subheading.
  7. Click Save configuration.
- Note: If the changes don't appear immediately in your browser, clear your browser's cache and reload the page. If you've bookmarked the site, you may need to delete the bookmark and then create it again, so the new favicon is used.

## Results

---

You have changed the favicon for the site.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the site email address

You can change the site email address in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To change the site email address, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. Under the System heading, click Basic site settings.
4. In the SITE DETAILS section, enter the new site email address in the Email address field.
5. Click Save configuration.

## Results

---

You changed the email address for the site.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the site logo

You can change the site logo for your Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To change the site logo, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
  2. Click Appearance in the administrator dashboard.
  3. Click Settings for the default theme.
  4. Click Logo image in the Override Global Settings section.
  5. Deselect the Use the logo supplied by the theme check box.
  6. Provide a path to your custom logo by completing the Path to custom logo field. Alternatively, you can browse and upload a logo image under the Upload logo image subheading.
  7. Click Save configuration.
- Note: If the changes don't appear immediately in your browser, clear your browser's cache and reload the page.

---

## Results

You changed the site logo for your Developer Portal. How the site logo is used depends on the settings for your theme.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the site name or site slogan

You can change the site name, or the site slogan, for your Developer Portal.

---

### Before you begin

You must have administrator access to complete this task.

---

### About this task

The site name is used to identify the site and can be displayed in browsers. You can also set a site slogan for your theme.

---

### Procedure

To change the site name or site slogan, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration > System > Basic site settings in the administrator dashboard.
3. In the SITE DETAILS section, enter the new site name in the Site name field, or enter a new slogan in the Slogan field.
4. Click Save configuration.

---

## Results

The site name or site slogan is updated. How the site name and the site slogan are used depends on the settings for your site theme. For more information, see [Changing the visibility of site branding](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Changing the visibility of site branding

You can toggle the visibility of your site branding elements for your Developer Portal.

---

### Before you begin

You must have administrator access to complete this task.

---

### About this task

The site branding block includes the site logo, name, and slogan. You can choose which of these branding elements you want to be displayed in this block.

---

### Procedure

To change the visibility of your site branding elements, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure > Block layout in the administrator dashboard.
3. Find the Site branding block in the Navigation section, and click Configure.
4. In the TOGGLE BRANDING ELEMENTS section, use the check boxes to select which branding elements you want to be visible.

5. Click Save block to save your changes.

## Results

---

You changed the visibility of your site branding elements. How these elements are used depends on the settings for your theme.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding custom JavaScript to a custom theme

To increase customization of the Developer Portal, you can add custom JavaScript to custom themes that you install.

### Procedure

---

1. Add your JavaScript file to your theme. For example, in a sub-directory named `js` and the file `myfilename.js`.
2. Add the JavaScript file to your theme, by editing your `{custom_theme}.libraries.yml` file and adding in the JavaScript file.

An example of the JavaScript file:

```
myfile:
  js:
    js/myfilename.js: {}
```

3. Add the following code to your theme's `.info.yml` file:

```
libraries:
- {custom_theme}/myfile
```

Note: You must replace `{custom_theme}` with your theme name. Also, both files must reference `myfile`.

4. Zip up the directory and upload it to your portal site. For more information, see [Installing additional themes](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Applying a modified content type template

You can apply a modified content type template to override a default content type template in the Developer Portal. For example, you can override the product content type template to change its layout.

### Before you begin

---

You must have administrator access to complete this task.

You must also have a custom sub-theme, and a copy of an existing template that you want to update. For information about how to create a sub-theme, see [Creating a sub-theme](#).

To obtain a copy of an existing template, you can download a copy from the following locations on GitHub:

- [https://github.com/ibm-apiconnect/devportal/tree/APIC\\_v2018/ibm\\_apim/templates](https://github.com/ibm-apiconnect/devportal/tree/APIC_v2018/ibm_apim/templates)
- [https://github.com/ibm-apiconnect/devportal/tree/APIC\\_v2018/apic\\_api/templates](https://github.com/ibm-apiconnect/devportal/tree/APIC_v2018/apic_api/templates)
- [https://github.com/ibm-apiconnect/devportal/tree/APIC\\_v2018/apic\\_app/templates](https://github.com/ibm-apiconnect/devportal/tree/APIC_v2018/apic_app/templates)
- [https://github.com/ibm-apiconnect/devportal/tree/APIC\\_v2018/product/templates](https://github.com/ibm-apiconnect/devportal/tree/APIC_v2018/product/templates)
- [https://github.com/ibm-apiconnect/devportal/tree/APIC\\_v2018/consumerorg/templates](https://github.com/ibm-apiconnect/devportal/tree/APIC_v2018/consumerorg/templates)

Note that you must copy the template to your sub-theme, and then modify the content. Do not modify the content that is stored on GitHub.

### About this task

---

You can override the Developer Portal core templates by creating a custom sub-theme, and then adding your modified templates to the sub-theme folder. A sub-theme inherits all of the settings of its parent theme, apart from the settings that are overridden, such as by applying modified templates. After you have applied a modified content types template, these changes take precedence over the default content types template. The Developer Portal templates are formatted as Twig template files. The following levels are examples of where you can use templates:

- `html.html.twig` - the template for Developer Portal HTML pages
- `page.html.twig` - the body of Developer Portal HTML page
- `node.html.twig` - the template for all of the content nodes
- `comment.html.twig` - the template for all of the comment nodes
- `search-result.html.twig` - the template for search results
- `node--product.html.twig` - the template for nodes that use the Product content type
- `node--product--teaser.html.twig` - the template for the previews for the Product content type

For more information about Twig templates, see [Working with Twig templates](#) and [Debugging Twig templates](#) on the Drupal documentation website. Modifying Twig templates allows very fine-grained control over the structure of pages. However, it also means you might miss new features and defect fixes that are made to the original templates, as your Developer Portal is using your overrides instead of the originals. So, if you override a template, it is your responsibility to check the templates in the latest API Connect releases, and to ensure that any equivalent changes that are needed to your overrides to maintain functional equivalence are made.

Important:

Directly editing any API Connect theme files is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.

## Procedure

To override a default content type template:

1. Apply the modifications that you require to the default template that you obtained.

Warning: Although the ability to override templates is supported, if you override a template they are your responsibility. Templates can be from multiple sources and compatibility with an earlier version is not ensured. You must check the templates in the latest API Connect release to check whether you need to make equivalent changes to your overrides to maintain functional equivalence.

2. Create a directory in your custom sub-theme, and name it templates.

3. Add your modified template to the templates directory.

Note: The modified template file name must be identical to the original name of the template file that it is overriding.

4. Log in to the Developer Portal UI as an administrator.

5. If the administrator dashboard is not displayed, click Manage to display it.

6. Install your custom sub-theme onto the Developer Portal by clicking Appearance > Install new theme on the administrator dashboard.

7. Click Choose file, select your sub-theme, and then click Install.

8. Click Install newly added theme, find your new theme in the list of Uninstalled themes, and click Install and set as default to set your new custom sub-theme as the default theme for your site.

Your new theme, which contains the modified template file, is now set as the default theme, and is listed in the Installed themes section.

## Results

You successfully modified and applied a template to override a default content type template in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Creating a sub-theme

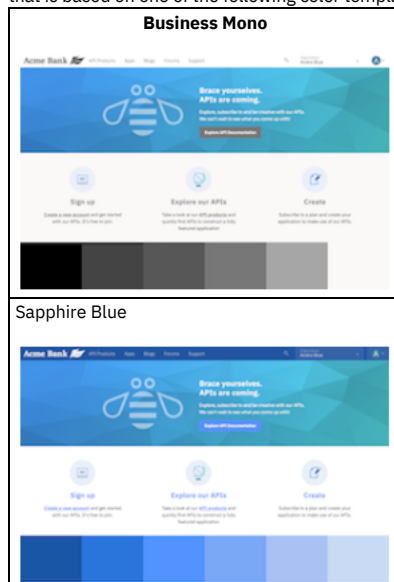
Create a custom theme for your Developer Portal site by generating and configuring a sub-theme. A sub-theme is a theme that inherits all the resources of a specified parent theme. You can then override specific resources to configure your required customizations.

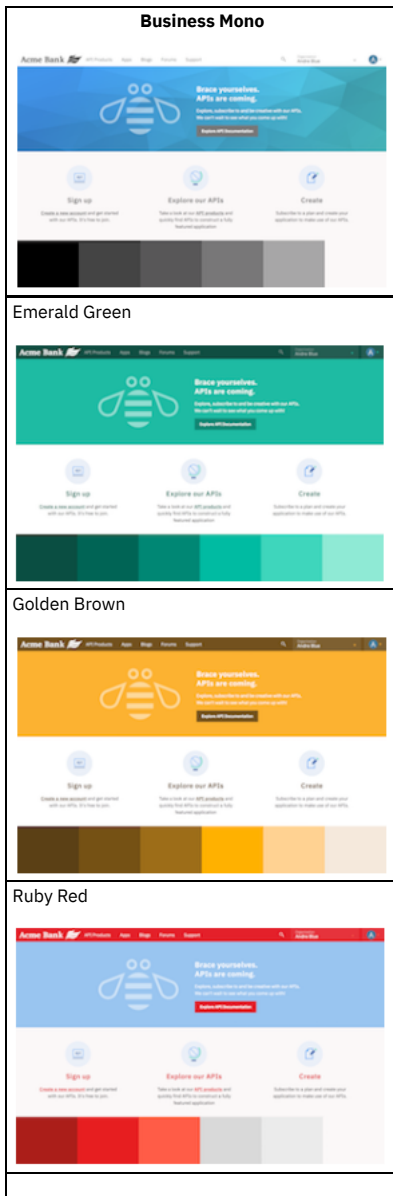
## Before you begin

You must have administrator access to complete this task.

## About this task

Directly editing the API Connect theme isn't permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed. So, the way to create a custom theme is to create a custom sub-theme of the standard API Connect theme that the Developer Portal uses by default. You can also create a sub-theme that is based on one of the following color templates:





Selecting a color template provides you with a base theme that you can build on more easily to produce your required site branding and styling. Create a sub-theme by using the theme generator in the Developer Portal UI, customize the appearance as required, and then upload the compressed file back onto the site.

**Important:**

- You're not permitted to include any IBM® API Connect code within any custom themes that you create. Also, directly editing any API Connect themes on the file system is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.
- All custom development is your responsibility. Although the use of custom themes is supported, IBM API Connect do not provide support in their development or modification.

For a guided example on creating a sub-theme, see [Tutorial: Creating a custom theme for the Developer Portal](#). In this example, you create a sub-theme, and then customize the styling by editing the overrides.css file.

## Procedure

1. If the administrator dashboard is not displayed, click **Manage** to display it.
2. On the administrator dashboard, click **Appearance**, **Generate sub-theme**.  
The **Generate sub-theme** page is displayed.
3. Enter a Sub-theme name, and select the Sub-theme type that you would like to use. The name can contain only lowercase characters a - z, and numerals 1 - 9.
  - CSS - Cascading Style Sheets; the language that is used to describe how to display the HTML elements in the theme.
  - SCSS - Sass CSS; a superset of CSS. SCSS contains all the features of CSS, but is extended to include the features of Sass as well. However, SCSS must be compiled into CSS before it can be installed and used in the Developer Portal.
4. Select the template to base your sub-theme on:
  - Default - the standard connect\_theme that the Developer Portal is based on.
  - Business Mono - a black and white color theme.
  - Sapphire Blue - a blue color theme.
  - Emerald Green - a green color theme.
  - Golden Brown - a brown color theme.
  - Ruby Red - a red color theme.



5. Click Generate.  
Your sub-theme is available to download from the Generate sub-theme page.
6. Click your new sub-theme and save it to your preferred location.
7. Extract the resources from the sub-theme file, and then edit the specific resources to configure the customizations that you require.  
Define your theme by creating metadata in the *your\_theme\_name*.info.yml file. You can then customize the appearance of your theme by configuring the overrides.css file. For detailed information about how to customize themes, see [Theming Drupal](#).
8. After you have completed your customizations, compress the sub-theme resources back into the .zip file.  
Note that the following file extensions are also supported: tar, tgz, gz, and bz2.
9. Install your new sub-theme onto the Developer Portal by clicking Appearance > Install new theme.
10. Click Choose file, select your sub-theme, and then click Install.
11. Click Install newly added theme, find your new theme in the list of Uninstalled themes, and click Install and set as default to set your new custom sub-theme as the default theme for your site.  
Your new theme is now set as the default theme, and is listed in the Installed themes section.

---

## Results

You created and installed a new sub-theme for your Developer Portal site.

---

## What to do next

You can edit the display settings for your new theme, by clicking Settings next to your theme.

---

## Related information

- [Tutorial: Creating a custom theme for the Developer Portal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Installing additional themes

You can install additional themes in the Developer Portal.

---

## Before you begin

You must have administrator access to complete this task.

---

## Procedure

To install additional themes, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click + Install new theme.
3. You can enter a path to the theme in the Install from a URL field. Alternatively, you can upload a theme under the Upload a module or theme archive to install heading.
4. Click Install.
5. Click Enable newly added themes.
6. Click Enable and set default for the theme that you want to be displayed in the Developer Portal.  
Your theme is displayed in the Developer Portal when you navigate to the home page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting themes

You can delete a theme from your Developer Portal site.

---

## Before you begin

You must have administrator access to complete this task.

---

## About this task

If you have a custom theme which is disabled, you can remove it from the Developer Portal. Deleting a theme is useful if you have identified a theme that you do not want to use again.

## Procedure

---

1. Click Appearance from the administrator dashboard.
2. Identify the custom theme that you want to delete, and ensure that it is disabled.
3. Click Delete against the theme.

The theme is now deleted from your Developer Portal site.

## Results

---

You deleted a disabled theme in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Content

---

Control multiple content entities in the Developer Portal, including the customization and restriction of specific content.

A *content entity* is an item of content data, which can consist of text, HTML markup, images, attached files, and other data that is intended to be displayed to site visitors. Content entities can be defined by the core software or by modules. Content entities are grouped into *entity types*, which have different purposes and are displayed in different ways on the site. Most entity types are also divided into *entity sub-types*, which are divisions within an entity type to allow for smaller variations in how the entities are used and displayed.

The Developer Portal site contains the following content entity types:

- API
- Application
- Article
- Basic page
- Blog post
- Book page
- Consumer Organization
- FAQ
- Forum Topic
- Product

Within content entity items the data is stored in individual *fields*, each of which holds one type of data, such as formatted or plain text, images or other files, or dates. Field types can be defined by the core software or by modules.

Fields can be added by an administrator on content entity sub-types, so that all the entity items of a given entity sub-type have the same collection of fields available. When you create or edit entity items, you are specifying the values for the fields on the entity item.

To learn how to configure the content entities of your Developer Portal site, use the following topic links.

- [Adding content elements in the Developer Portal](#)  
You can add content elements in the Developer Portal.
- [Configuring and restricting content in the Developer Portal](#)  
You can configure and restrict certain content elements in the Developer Portal.
- [Turning content on or off in the Developer Portal](#)  
You can turn certain content elements on or off in the Developer Portal

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding content elements in the Developer Portal

You can add content elements in the Developer Portal.

To learn how to add various content elements in the Developer Portal, use the following topic links:

- [Adding and configuring meta tags](#)  
You can customize the Developer Portal by adding and configuring meta tags.
- [Adding custom pages](#)  
You can create custom pages within the Developer Portal. You can add a basic page for static content, or an article for news content.
- [Integrating Twitter data into the social block](#)  
You can retrieve and integrate data from a Twitter account of interest and display it into a social block in a Developer Portal site.
- [Adding content in multiple languages](#)  
You can create content in multiple languages in the Developer Portal.
- [Adding custom pages to APIs and Products](#)  
After you create a custom page in the Developer Portal, you can add a link to the new page to any APIs and Products that exist in the Developer Portal.

- [Adding new frequently asked questions](#)  
You can add new FAQs to the Developer Portal.
- [Applying an image to an API or Product](#)  
You can apply an image to an API or a Product in the Developer Portal.
- [Attaching a documentation file to APIs](#)  
You can attach a file to APIs in the Developer Portal, and display them as documentation attachments. You can attach multiple files to APIs.
- [Configuring blogs](#)  
You can post new blog entries in the Developer Portal, and configure them to suit your requirements. You can also turn blogs off.
- [Configuring the taxonomy menu block](#)  
To display your tag hierarchy in the Developer Portal, you must configure the taxonomy menu block.
- [Customizing the URL alias for a specific API or Product page](#)  
You can customize the URL of an API or Product page from the default alias that they are assigned.
- [Embedding multimedia in site content](#)  
You can embed multimedia elements in site content in the Developer Portal.
- [Linking from one piece of site content to another](#)  
You can link from one piece of site content to another in the Developer Portal.
- [Linking to social media sites](#)  
You can link to social media sites from the Developer Portal. You can configure where and how these links are displayed and which sites you want to link to.
- [Managing tags in the Developer Portal](#)  
These instructions show you how to add new tags to your taxonomy, and how to manage the tag hierarchy in the Developer Portal. You can use tags to classify your Developer Portal content.
- [Posting a poll](#)  
You can post a poll in the Developer Portal.
- [Adding images to site content](#)  
You can add .jpg, .png and .gif images to your site content in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Adding and configuring meta tags

You can customize the Developer Portal by adding and configuring meta tags.

### Before you begin

You must have administrator access to complete this task.

### About this task

Meta tags provide preview information that is displayed for a link to a website. The preview information can include images, descriptions, tags, and links. Meta tags are used when you link to external website, for example Facebook, Twitter, and Slack.

You can see which entities have the meta tag form available on their respective edit pages by default, and can be configured, Configuration > Search and metadata > Metatags > Settings:

- CONTENT
  - API
  - APPLICATION
  - ARTICLE
  - BLOG POST
  - BOOK PAGE
  - CONSUMER ORGANIZATION
  - FAQ
  - FORUM TOPIC
  - BASIC PAGE
  - PRODUCT
- TAXONOMY TERM
  - FORUMS
  - TAGS
- USER
  - USER

### Procedure

1. In the administrator dashboard, click Extend > Update, then click List.
2. Enter `meta tag` in the Filter list field.
3. Select the check boxes for the Meta tags you want to enable, click Enable.

- [Disabling external search engine indexing](#)  
If you do not want your Developer Portal site content to be indexed by external search engines, then you can disable the indexing.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Disabling external search engine indexing

If you do not want your Developer Portal site content to be indexed by external search engines, then you can disable the indexing.

### Before you begin

---

You must have administrator or content author access to complete this task.

### About this task

---

By disabling the external search indexing, you can increase the confidentiality of your content and restrict its exposure.

### Procedure

---

1. If the administrator dashboard is not displayed, click **Manage** to display it.
2. On the administrator dashboard, click **Configuration**.
3. In the **SEARCH AND METADATA** section, click **Metatag**.
4. Click **Edit** for the **GLOBAL** type.
5. Expand the **ADVANCED** section and, in the **Robots** section, select **Prevents search engines from indexing this page**.
6. Click **Save**.

### Results

---

You have disabled external search engine indexing.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding custom pages

You can create custom pages within the Developer Portal. You can add a basic page for static content, or an article for news content.

### Before you begin

---

You must have administrator or content author access to complete this task.

### About this task

---

Use basic pages for your static content, such as an "About us" page. Typically, a basic page is used for static content that can be linked into the main navigation bar, although this is not a requirement. Use articles for time-sensitive content, like news, press releases, or blog posts.

### Procedure

---

To add custom pages, complete the following steps.

1. Log in to the Developer Portal as an administrator or content author.
2. If the administrator dashboard is not displayed, click **Manage** to display it.
3. In the administrator dashboard, click **Content**.
4. Click **+Add content**, then click either **Basic page** or **Article**, depending on the type of content you want to display.
5. Enter a page title, select the language, and enter the content.
6. Specify the remaining page options as required.
  - If you want your new page to be linked to from a menu, you must complete the **Menu settings** section.
7. Optional: Click **Edit summary** and specify a summary page description.
8. Click **Save** to publish the new page immediately. If you don't want to publish the new page immediately, ensure that you **deselect Published** before saving.

### Results

---

You added a new custom page to the Developer Portal. You can edit the page after it is created by clicking **Edit** for the custom page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Integrating Twitter data into the social block

You can retrieve and integrate data from a Twitter account of interest and display it into a social block in a Developer Portal site.

### Before you begin

---

You must have administrator access to complete this task.

You must have a Twitter App for the Twitter account of interest. The Twitter App contains information that is used to configure the social block. For information on creating Twitter Apps, see <https://apps.twitter.com/>.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. On the administrator dashboard, click Configuration.
3. In the WEB SERVICES section, click IBM Social Block settings.
4. Using the information from your Twitter App, enter the Consumer Key, Consumer Secret, Access Token, and Token Secret in the corresponding fields.
5. Click Save configuration.

Twitter data from a Twitter account of interest has been integrated with the Developer Portal, and is displayed on the social block of the site.

- **Editing the social block**

After you have integrated Twitter data into the Developer Portal social block, you can edit the content, appearance, and functionality of the social block.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing the social block

After you have integrated Twitter data into the Developer Portal social block, you can edit the content, appearance, and functionality of the social block.

### Before you begin


---

You must have administrator access to complete this task.

You must have Twitter data integrated into the Developer Portal social block. For more information, see [Integrating Twitter data into the social block](#).

### Procedure

---

1. On your social block, click the Settings icon , then click Edit Block.  
The Edit Social Block window is displayed, with the Edit Block tab open.
2. Configure your social block by configuring any of the following options:

Number of tiles to display

Configure the number of tiles that you want to display on the social block by entering a number in the Number of tiles to display field. The default number of tiles that are displayed on the social block is 8, and there is a limit of 100 tiles.

Display topics from all forums

If you want to display the topics from all forums, ensure that the Display topics from all forums check box is selected. If you want to specify which forum topics are displayed, clear the check box and select the appropriate forums.

Get tweets from

From the Get tweets from drop-down list, select whether you want to display Twitter data from users or a search term. If you select User, enter the account name for the user, in the adjacent field. If you select Search term, enter the specific term that will return the corresponding Twitter data, in the adjacent field.

Twitter cache duration

Enter the length of time, in minutes, that you want the Twitter data to remain in the cache of the Developer Portal. The default length of time is 60 minutes.

Types of tweets to display

Select the types of tweets that you want to display from the drop-down list. You can select to display tweets, or tweets and replies.

3. After you have configured the social block, click Save.  
Your social block is configured, and the configurations are implemented when you return to the home page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding content in multiple languages

You can create content in multiple languages in the Developer Portal.

## Before you begin

---

You must have administrator or content author access to complete this task.

## About this task

---

You can provide custom pages in multiple languages to your APIs and Products to improve your customer experience and understanding. By default, multiple languages are enabled in the Developer Portal.

Note: In API Manager, translations for APIs and products must be done within a document. For more information, see [Using x-ibm-languages to create multilingual API and Product documentation](#).

## Procedure

---

1. Log in to the Developer Portal as an administrator or content author.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Content.
4. If you are creating a new page to be translated, click +Add content.
  - a. Click Basic page, then select type, for example **Basic page**.
  - b. Enter a title for your custom page in the Title field.
  - c. Enter your information in the Body field.
  - d. Click Save.
  - e. Click Translate.
  - f. Click Add for the language you want to translate.
  - g. Provide the language content.
  - h. Click Save (this translation).
5. If you are translating an existing page, find the page that you want to update in the list.
  - a. Click Translate from the drop-down list for the page.
  - b. Click Add for the language you want to translate.
  - c. Provide the language content.
  - d. Click Save (this translation).

## Results

---

You successfully created a translated version of your content.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding custom pages to APIs and Products

After you create a custom page in the Developer Portal, you can add a link to the new page to any APIs and Products that exist in the Developer Portal.

## Before you begin

---

You must have administrator or content author access to complete this task.

You must have an API or Product to add the custom page to.

## About this task

---

You can add custom pages to your APIs and Products to provide further information on their use and value.

For example, you can create a custom documentation page on the implementation of an API in a specific circumstance, and add this custom documentation page to the API.

## Procedure

---

1. Log in to the Developer Portal as an administrator or content author.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Content, then click +Add content.
4. Click Basic page.

The Create Basic page window is displayed.
5. Enter a title for your custom page in the Title field.
6. Enter your information of interest in the Body field.
7. If you want to add the custom page to a specific Product, enter the Product name in the Link to one or more specific Products field. As you type into this field, a list of available Products is displayed for you to select. You can add more Products by clicking Add another item. If you want to add the custom page to all Products, select the Link to all Products check box.
8. If you want to add the custom page to a specific API, select the APIs tab, and then enter the API name in the Link to one or more specific APIs field. As you type into this field, a list of available APIs is displayed for you to select. You can add more APIs by clicking Add another item. If you want to add the custom page to all APIs,

- select the Link to all APIs check box.
- 9. Optional: Configure the additional settings of the custom page to your specification.
- 10. Click Save.

---

## Results

Your custom page is created and linked to the APIs or Products that you have specified. You can edit the page after it is created, and update the APIs and Products that it is linked to, by clicking Edit for the custom page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding new frequently asked questions

You can add new FAQs to the Developer Portal.

---

### Before you begin

You must have administrator or content author access to complete this task.

---

### About this task

To add new FAQs, you create a new FAQ node and add questions and answers to that node. Each FAQ node represents an FAQ topic, and each topic can contain multiple questions and answers.

---

### Procedure

To add new FAQs, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Content in the administrator dashboard.
4. Click +Add content.
5. Click FAQ.
  - The Create FAQ page displays.
6. Provide a Title for the new FAQ topic.
7. Enter the question in the Question field, and enter the answer in the Answer field.
8. Optional: If you have more content for this FAQ topic, click Add another item and complete the Question and Answer fields. Repeat this step as required.
9. Click Save to save your new FAQ topic.
  - Your new FAQ topic is displayed in the Support section of the Developer Portal.

---

## Results

You successfully added new FAQs to the Developer Portal.

---

### What to do next

You can edit the current content of FAQs by clicking Content in the administrator dashboard, finding the FAQ content that you want to edit, and clicking Edit against that FAQ.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Applying an image to an API or Product

You can apply an image to an API or a Product in the Developer Portal.

---

### Before you begin

You must have administrator or content author access to complete this task.

---

### Procedure

To add an image to an API or Product, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.

2. Click Content in the administrator dashboard.
3. Click Edit next to the API or Product that you want to edit.
4. Under the Image heading, click Choose File to select an image.
5. Click Save and keep published, to publish the updated content, or select Save and unpublish if you want to publish the updated content later.  
You have added a new picture to the selected API or Product.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Attaching a documentation file to APIs

You can attach a file to APIs in the Developer Portal, and display them as documentation attachments. You can attach multiple files to APIs.

### Before you begin

---

You must have administrator or content author access to complete this task.

### About this task

---

You can attach only one file at a time. The default file upload size limit applies to the files that you want to attach, and you can attach up to 10 files to an API by default.

**Note:** You can change the default limit for the number of files that you want to attach.

An administrator can configure the types of file that are uploaded, and the number of files that a user can upload.

The following file types can be uploaded:

- txt doc pdf
- xls ppt pptx
- docx xlsx rtf
- odt ods odp md json
- yaml tgz tar zip

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Click Edit next to the API you want to attach the file to.
4. In the Documentation section, click Browse and identify the file that you want to upload, then click Open.
5. Click Upload, then click Save.
6. Optional: Clear the Display checkbox for the file if you want to upload and attach the file to the API, but not show it in the Developer Portal.
7. Optional: Enter a description for your file in the Description field. This description can be used as the label of the link to the file.
8. Optional: If you uploaded multiple files, you can use the drag handle next to the file name to rearrange them.

### Results

---

You successfully attached a documentation file to your API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring blogs

You can post new blog entries in the Developer Portal, and configure them to suit your requirements. You can also turn blogs off.

### Before you begin

---

You must have administrator or content author access to complete this task.

### Procedure

---

To post a new blog entry, complete the following steps:

- If the administrator dashboard is not displayed, click Manage to display it.
- Click Content in the administrator dashboard.
- Click Add content.
- Click Blog post.
- Fill in the Title and Body fields.
- Click Save and keep published, to publish the updated content, or select Save and unpublish if you want to publish the updated content later.



You have posted a new blog entry.

To configure a blog post, complete the following steps:

- If the administrator dashboard is not displayed, click Manage to display it.
- Click Content in the administrator dashboard.
- Click Edit next to the blog post that you want to configure.
- You can configure the selected blog from this view.
- Click Save and keep published, to publish the updated content, or select Save and unpublish if you want to publish the updated content later. You have configured the selected blog post.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the taxonomy menu block

To display your tag hierarchy in the Developer Portal, you must configure the taxonomy menu block.

### Before you begin

---

You must have administrator or content author access to complete this task.

### About this task

---

Along with controlling how your tag hierarchy is displayed through configuring your taxonomy menu block, you can configure options for the taxonomy tree. The following options can be configured for your taxonomy menu block:

- Vocabulary
- Parent option
- Depth
- Node options and content
- Region settings
- Visibility settings

Important: The taxonomy block will appear after configuration only if there are tags for it to display.

### Procedure

---

1. In the Administrator dashboard, click Structure > Blocks.
2. Click configure for the Taxonomy Menu Block (Tags) block.
3. Configure the options that are available for your taxonomy menu block.
4. When you have configured your taxonomy menu block, click Save block.  
You have configured your taxonomy menu block, and your tag hierarchy is displayed based on your configurations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Customizing the URL alias for a specific API or Product page

You can customize the URL of an API or Product page from the default alias that they are assigned.

### Before you begin

---

You must have administrator or content author access to complete this task.

### About this task

---

Creating a custom URL alias is useful if you want to make the URL relevant to the API or Product page.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. On the administrator dashboard, click Content.
3. Identify the API or Product page that you want to customize the URL alias for, and click Edit.
4. Expand URL path settings.
5. In the URL alias text field, enter the custom URL alias that you want to assign your API or Product page.

- Note: URL aliases must all be unique.
6. Click Save and keep published, to publish the updated content, or select Save and unpublish if you want to publish the updated content later.

---

## Results

You have applied a custom URL alias for your specified API or Product page, and it is displayed in the URL of your browser when the API or Product page is selected.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Embedding multimedia in site content

You can embed multimedia elements in site content in the Developer Portal.

---


### Before you begin

You must have administrator or content author access to complete this task.

---

### Procedure

To embed multimedia in site content, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Under the TITLE heading, click the title of the required content.
4. Click the Edit tab.
5. Use the Insert Media Embed icon  in the WYSIWYG editor to embed multimedia in the content.  
Note: The Insert Media Embed icon is available only if you have selected Full HTML mode.
6. Click Save and keep published to publish the new page immediately, or select Save and unpublish if you want to publish the page later.  
You have embedded multimedia in the selected site content.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Linking from one piece of site content to another

You can link from one piece of site content to another in the Developer Portal.

---


### Before you begin

You must have administrator or content author access to complete this task.

---

### Procedure

To link from one piece of site content to another, complete the following steps:

1. Click Content in the administrator dashboard.
2. Under the TITLE heading, click the title of the required content.
3. Select the Edit tab.
4. Use the Link to content icon  in the WYSIWYG editor.
5. Click Save.  
You have linked the selected site content to another site content element.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Linking to social media sites

You can link to social media sites from the Developer Portal. You can configure where and how these links are displayed and which sites you want to link to.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

1. Click Structure in the administrator dashboard.
2. Click Blocks.
3. Under the Disabled heading, locate the Follow site block.
4. Select the location for your social media block from the drop-down list for Follow Site in the REGION column.  
The Follow Site block will appear under the heading that corresponds to the region you selected in the REGION drop-down list.
5. Required: Click Save blocks before you begin configuring your new block.
6. Click configure for the Follow Site block.
7. In the Block title field, type the name of your block.
8. Under the Default block title heading, select one of the following leading text options for your block using the radio buttons for each option:
  - "Follow *your.portal.site* on"
  - "Follow me on"
  - "Follow us on"
9. Use the User pages check box to decide whether you want your block to display on your user's profile pages.
10. Use the Alignment and Icon Style drop-down lists to adjust the appearance of your block.
11. Click Save block.
12. Click Configuration in the administrator dashboard.
13. Under the WEB SERVICES heading, click Follow.
14. Type the URL under the URL heading for each of the social media sites you want your users to access. The following is an example of a link to the @ibmapiconnect twitter handle:  

```
https://twitter.com/ibmapiconnect
```
15. Click Submit.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing tags in the Developer Portal

These instructions show you how to add new tags to your taxonomy, and how to manage the tag hierarchy in the Developer Portal. You can use tags to classify your Developer Portal content.

## Before you begin

---

You must have administrator or content author access to complete this task.

## Procedure

---

To manage your tags, complete the following steps

1. If the administrator dashboard is not displayed, click Manage to display it.
  2. Click Structure > Taxonomy > Tags in the administrator dashboard.  
The List window for Tags is displayed.
  3. You can add new terms, delete, and reorganize terms from this view.
    - a. To add a new term, click Add term, define the new term, then click Save.
    - b. To delete a term, select Delete from the OPERATIONS menu next to the term that you want to delete. Then click Delete to confirm.
    - c. To reorganize terms, ensure that the row weights are hidden by clicking Hide row weights if necessary, then use the drag-and-drop handles to manage the tag hierarchy. Terms can be dragged up and down in the list, and they can also be dragged to the right or left to organize a parent grouping hierarchy. Then click Save.
- [Tagging APIs based on their lifecycle phase](#)  
In the Developer Portal, you can have APIs automatically tagged with their lifecycle phase; realized, identified, or specified.

## Related tasks

---

- [Editing tags for a specific item](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tagging APIs based on their lifecycle phase

In the Developer Portal, you can have APIs automatically tagged with their lifecycle phase; realized, identified, or specified.

---

## Before you begin

You must have administrator or content author access to complete this task.

---

## About this task

APIs are tagged in the API Manager with a lifecycle phase as follows:

- Realized (default) - The API is in the implementation phase.
- Identified - The API is in the early conceptual phase and is neither fully designed nor implemented.
- Specified - The API has been fully designed and passed an internal milestone, but has not yet been implemented.

You can configure the Developer Portal to tag APIs with their lifecycle phase automatically.

---

## Procedure

Tagging APIs with their phase is disabled by default. To enable the function:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. On the administrator dashboard, click Configuration.
3. In the SYSTEM section, click IBM API Connect.
4. Select Automatically tag APIs with their phase, and then click Save configuration.

---

## What to do next

You can manage the tag hierarchy of your APIs. For more information, see [Managing tags in the Developer Portal](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Posting a poll

You can post a poll in the Developer Portal.

---

## Before you begin

You must have administrator or content author access to complete this task.

---

## Procedure

To post a poll, complete the following steps:

1. Click Content in the administrator dashboard.
2. Click Add content.
3. Click Poll.
4. Enter the poll question in the Poll question field.
5. Under the Choice heading, add answers to the available fields as choices.
6. From the Poll duration drop-down menu, select a time limit for the poll.
7. Click Save.

You have posted a poll.

---

## What to do next

See [Editing sample content](#) to learn how to add a links to your poll to sample content.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding images to site content

You can add .jpg, .png and .gif images to your site content in the Developer Portal.

---


## Before you begin

You must have administrator or content author access to complete this task.

## Procedure

---

To upload images for use in site content, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Under the TITLE heading, click the title of the required content.
4. Click the EDIT tab.
5. Use the Image icon  in the WYSIWYG editor to add an image.
6. Click Save and keep published to publish the new page immediately, or select Save and unpublish if you want to publish the page later.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring and restricting content in the Developer Portal

You can configure and restrict certain content elements in the Developer Portal.

To learn how to configure and restrict content in the Developer Portal, use the following topic links:

- [Creating content types](#)  
Content types define default fields for editors to add content on your Developer Portal site and are the building blocks for authoring content. You can control which content types are available
- [Configuring documentation upload limitations](#)  
You can attach documentation to an API by uploading it in the Developer Portal. You can configure the API content type to place limitations on the uploaded documentation.
- [Configuring image upload limitations](#)  
You can attach an image file to an API by uploading it in the Developer Portal. You can configure the API content type to place limitations on the uploaded image files.
- [Configuring the appearance of ratings in the Developer Portal](#)  
You can configure the appearance of ratings in the Developer Portal.
- [Customizing the privacy policy statement](#)  
You can define and customize a privacy policy statement that sets out the privacy conditions of your Developer Portal site. The privacy policy can be linked to from the cookie compliance banner.
- [Customizing the terms of use statement](#)  
You can define and customize a terms of use statement that sets out the terms and conditions that your users must accept to use your Developer Portal site.
- [Editing a site comment](#)  
You can edit site comments in the Developer Portal.
- [Editing sample content](#)  
You can edit sample content, such as basic pages, in the Developer Portal.
- [Editing tags for a specific item](#)  
You can edit tags for a specific item in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating content types

Content types define default fields for editors to add content on your Developer Portal site and are the building blocks for authoring content. You can control which content types are available

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

After you create your content type, it is added to the following list of default content types that are present in the Developer Portal:

- API
- Application
- Article
- Basic page
- Blog post
- Book page
- Consumer organization
- Forum topic
- Product

The content type that you create, and the default content types, can be edited and configured further to your specifications.

## Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. Click +Add content type.
5. Enter a name for your content type in the Name field.
6. Optional: Enter the text that you want to appear on the Add new content page, in the Description field.
7. Enter a title field label for your content type in the Title field label field.
8. Optional: Configure any of the following options for your content type:
  - Submission form settings
  - Publishing options
  - Language settings
  - Display settings
  - Menu settings
9. After you have configured your content type to your specifications, click Save and manage fields.  
You have created a new content type.

## What to do next

---

You can add fields to your new content type; see [Adding fields to content types](#).

- [Adding fields to content types](#)  
Fields can be added to content types in the Developer Portal for customization, and to add functionality.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Adding fields to content types

Fields can be added to content types in the Developer Portal for customization, and to add functionality.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

You can specify the type of data that a field can store, and the form element to edit the data with. The type of form element that is available is dependent on the type of data that your field stores.

**Note:** You can specify the number of values that a field can store, including an unlimited amount. You can configure the value after the field is created. Reducing the number of values that can be stored after the field is created might lead to data loss.

## Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types, and identify the content type that you want to add a field to.
4. Click Manage fields for the content type, then click Add field.
5. Select a field type or select an existing field to reuse, then enter a label for your field in the Label field.  
The label that you specify for your field is displayed in the UI.
6. Optional: If you want to edit the machine name that is automatically created based on your label, click Edit and enter the new name.  
You cannot change the machine name after the field is created.
7. Click Save and continue.
8. Modify the field type specific settings as required, then click Save field settings.  
You have added a field to your content type.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring documentation upload limitations

You can attach documentation to an API by uploading it in the Developer Portal. You can configure the API content type to place limitations on the uploaded documentation.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

The set of configurable limitations that you can place on uploading documentation is different to the set of limitations that you can place on uploading images.

The file size limit that is applied when you upload documentation is also different to the limit that is applied when you upload images.

## Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. For the API content type, click Manage fields.
5. For the Documentation field, click Edit.
6. Configure the type of files that are allowed to be uploaded in the Allowed file extensions field.
7. Enter the file size upload limit in the Maximum upload size field.  
The default size is 10MB, while the maximum upload size possible is 64MB.
8. Click Save settings.  
You have configured the documentation upload limitations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring image upload limitations

You can attach an image file to an API by uploading it in the Developer Portal. You can configure the API content type to place limitations on the uploaded image files.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. For the API content type, click Manage fields.
5. For the Image field, click Edit.
6. Configure the type of image files that are allowed to be uploaded in the Allowed file extensions field.
7. Specify the maximum and minimum allowed image sizes in the Maximum image resolution and Minimum image resolution fields.
8. Enter the file size upload limit in the Maximum upload size field.  
The default size is 2MB, while the maximum upload size possible is 64MB.
9. Click Save settings.  
You have configured the image upload limitations.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the appearance of ratings in the Developer Portal

You can configure the appearance of ratings in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

You can configure the appearance of ratings for only the following content types in the Developer Portal:


- APIs
- Applications
- Products

Note: The Applications and Products content types rating functionality is set to hidden by default.

## Procedure

---

These following steps describe how to customize the rating on a selected API, Application, or Product. To ensure that the configured rating is also displayed in the list view of APIs, Applications, or Products, then you must repeat these steps on the Teaser tab in the Manage Display window:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. Select Manage display from the list in OPERATIONS column, next to the required content type.
5. Select the style of the Rating field from the list in the FORMAT column.
6. Optional: Click the Settings icon  this icon to refine the display configuration further, then click Update.
7. Click Save. You have configured the appearance of the ratings feature for the required content type.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Customizing the privacy policy statement

You can define and customize a privacy policy statement that sets out the privacy conditions of your Developer Portal site. The privacy policy can be linked to from the cookie compliance banner.

### Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To define or customize a privacy policy, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Content, then enter `privacy` into the title search filter, and click Filter.  
The Privacy Policy content item is displayed in the list of content.
4. Click Edit next to the Privacy Policy content item.
5. Update the content as required. For example, you can complete the following tasks:
  - a. Edit the text of the privacy policy statement.
  - b. Set the display settings.
  - c. Link the content to Products and APIs.
  - d. Enter an explanation of the changes that have been made to the privacy policy since the last version.
6. Click Save to save your changes.

## Results

---

The privacy policy statement is successfully updated.

### What to do next

---

You can set your cookie compliance banner to link to your updated privacy policy. For more information, see [Enabling a cookie compliance banner](#).

### Related tasks

---

- [Enabling a cookie compliance banner](#)
- [Customizing the terms of use statement](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Customizing the terms of use statement



You can define and customize a terms of use statement that sets out the terms and conditions that your users must accept to use your Developer Portal site.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To define or customize a terms of use statement, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Content, then enter `terms` into the title search filter, and click Filter.  
The Terms of use content item is displayed in the list of content.
4. Click Edit next to the Terms of use content item.
5. Update the content as required. For example, you can complete the following tasks:
  - a. Edit the text of the terms of use statement.
  - b. Set the display settings.
  - c. Link the content to Products and APIs.
  - d. Enter an explanation of the changes that have been made to the terms of use since the last version.
6. Click Save to save your changes.

## Results

---

The terms of use statement is successfully updated.

## What to do next

---

You can force new users to accept the terms and conditions on the Developer Portal. For more information, see [Forcing new users to accept terms and conditions](#).

## Related tasks

---

- [Enabling a cookie compliance banner](#)
- [Customizing the privacy policy statement](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing a site comment

You can edit site comments in the Developer Portal.

## Before you begin

---

You must have administrator or content author access to complete this task.

## Procedure

---

To edit a site comment, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Click the Comments tab.
4. In the OPERATIONS column, click Edit alongside the comment that you want to edit.
5. Edit the content of the comment as required.
6. Click Save.  
You have edited the selected comment.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing sample content

You can edit sample content, such as basic pages, in the Developer Portal.

## Before you begin

---

You must have administrator or content author access to complete this task.

## Procedure

---

To edit sample content, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. In the OPERATIONS column, click Edit alongside the content item that you want to edit.
4. Edit the sample content of the selected element as required.
5. Click Save and keep published, to publish the updated content, or select Save and unpublish if you want to publish the updated content later.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing tags for a specific item

You can edit tags for a specific item in the Developer Portal.

## Before you begin

---

You must have administrator or content author access to complete this task.

You must have some tags that are available to apply to your content; see [Managing tags in the Developer Portal](#).

## Procedure

---

To edit tags for a specific item, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. In the list of content items, click Edit alongside the content item that you want to update.
4. In the Tags section, select the required tag.
5. To add another tag, click Add another item, then select the required tag.
6. If you want to publish the updated content, ensure the Published check box is selected. Deselect the check box if you want to publish the updated content later.
7. Click Save.

## Related tasks

---

- [Managing tags in the Developer Portal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Turning content on or off in the Developer Portal

You can turn certain content elements on or off in the Developer Portal

To learn how to turn content on or off in the Developer Portal, use the following topic links:

- [Deleting a site comment](#)  
You can delete site comments in the Developer Portal.
- [Deleting blocks](#)  
You can delete blocks in the Developer Portal.
- [Disabling blogs](#)  
You can disable blogs on your Developer Portal.
- [Deleting forums from the front page](#)  
You can delete the help information on forums from the front page of your Developer Portal.
- [Turning comments for specific content types on or off in the Developer Portal](#)  
You can turn the comments for specific content types on or off in the Developer Portal.
- [Turning comments off for an individual item](#)  
You can turn off comments for individual items within a content type.
- [Turning off ratings for specific content types in the Developer Portal](#)  
You can turn off ratings for specific content types in the Developer Portal by hiding the ratings field.

- [Turning the ability to tag specific content types off in the Developer Portal](#)

You can turn the ability to tag specific content types on or off in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting a site comment

You can delete site comments in the Developer Portal.

### Before you begin

---

You must have administrator or content author access to complete this task.

### Procedure

---

To delete a site comment, complete the following steps

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Click the Comments tab.
4. From the UPDATE OPTIONS list, select Delete the selected comments.
5. Select the comments that you want to delete, then click Update.  
The confirmation screen displays.
6. Click Delete comments.  
You have deleted the selected comments.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Deleting blocks

You can delete blocks in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Pages.
4. In the OPERATIONS column, find the welcome page row and select Edit from the drop-down list.
5. Expand Panels, then click Content.
6. In the OPERATIONS column, find the block that you want to delete and select Delete for the drop-down list.
7. Click Delete to confirm, then click Update and save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Disabling blogs

You can disable blogs on your Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

Before any changes are made, **Blogs** show as an option on the front page like this:

## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure > Menus from the administrator dashboard.
3. In the OPERATIONS column, click Edit menu for **Main navigation**.
4. Clear Enabled for the **Blogs Menu link**, then click Save.

**Title \***

Main navigation Machine name: main

**Administrative summary**

Site section links

**Menu language**

English ▼

| MENU LINK      | ENABLED                             | OPERATIONS |
|----------------|-------------------------------------|------------|
| ⊕ Home         | <input checked="" type="checkbox"/> | Edit ▼     |
| ⊕ API Products | <input checked="" type="checkbox"/> | Edit ▼     |
| ⊕ Apps         | <input checked="" type="checkbox"/> | Edit ▼     |
| ⊕ Blogs        | <input type="checkbox"/>            | Edit       |
| ⊕ Forums       | <input checked="" type="checkbox"/> | Edit ▼     |
| ⊕ Support      | <input checked="" type="checkbox"/> | Edit       |

Save

## Results

You removed the **Blogs** option from the front page. The options on the front page of your Developer Portal now look like this:



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting forums from the front page

You can delete the help information on forums from the front page of your Developer Portal.

## Before you begin

You must have administrator access to complete this task.

Before any changes are made, the front page help bar looks like this:



## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Block layout > Custom block library.
4. In Block description, enter `Get help`, then click Apply.

|                                       |                                      |
|---------------------------------------|--------------------------------------|
| Block description                     | Block type                           |
| <input type="text" value="Get help"/> | <input type="text" value="- Any -"/> |

Apply

| BLOCK DESCRIPTION | BLOCK TYPE  | UPDATED            | OPERATIONS |
|-------------------|-------------|--------------------|------------|
| Get Help [en]     | Basic block | 05/07/2020 - 19:52 | Edit       |

5. In the OPERATIONS column, click Edit for `Get Help`.
6. In the **Body** section, delete the forum information, then click Save.

## Results

You removed the forum information from the front page of your Developer Portal and the front page now looks like this:

The screenshot shows a dark-themed navigation bar with three main sections:

- Getting Help:** "Be sure to check out these extra help resources." Below it is a link "Create an Account" with a right-pointing arrow.
- Get Started:** "Get started with our APIs by creating an account and exploring the documentation to find what's right for you." Below it is a link "Email us" with a right-pointing arrow.
- Contact Us:** "Can't find the answer to your question? Need more help? Have some feedback? Let us know!" Below it is a link "Email us" with a right-pointing arrow.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Turning comments for specific content types on or off in the Developer Portal

You can turn the comments for specific content types on or off in the Developer Portal.

### Before you begin

You must have administrator access to complete this task.

## Procedure

To turn comments for specific content types on or off in the Developer Portal, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. In the OPERATIONS column, click Manage fields for the required content type.
5. In the OPERATIONS column, click Edit for the Comments field.

Note:

- If there is no Comments field, you can enable commenting for the selected content type by adding one as follows:
    - Click Add field.
    - From the Add a new field list, select Comments.
    - Enter a field label, then click Save and continue.
    - Select the required comment type, then click Save field settings.
  - You can completely disable commenting for the selected content type, and permanently remove all current comments, by selecting Delete.
6. Select the required option in the DEFAULT VALUE section.
    - a. To allow commenting, select Open.
    - b. To keep existing comments but disallow further commenting, select Closed.
    - c. To hide all comments, select Hidden.
  7. When done, click Save settings.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Turning comments off for an individual item

You can turn off comments for individual items within a content type.

## Before you begin

---

Users with edit permission for the selected content type can perform this task.

## Procedure

---

To turn off comments for a single item, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. In the OPERATIONS column, click Edit for the required content item.
4. Click Comment settings in the side pane.
5. Select Closed to turn comments off for the selected item.
6. Click Save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Turning off ratings for specific content types in the Developer Portal

You can turn off ratings for specific content types in the Developer Portal by hiding the ratings field.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To turn off ratings for specific content types in the Developer Portal, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Content types.
4. In the OPERATIONS column for the required content type, select Manage fields from the drop-down list.
5. Click on the Manage form display tab. In the FIELD list, click on Rating and drag and drop it into the Disabled, then click Save.
6. Click on the Manage display tab. In the FIELD list, click on Rating and drag and drop it into the Disabled, then click Save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Turning the ability to tag specific content types off in the Developer Portal

You can turn the ability to tag specific content types on or off in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To turn the ability to tag specific content types on or off in the Developer Portal, complete the following steps:

1. Click Structure in the administrator dashboard.
2. Click Content types.
3. Click Manage fields next to the required content type.
4. From the Tags field, click Edit.
5. Click Delete, then confirm the Delete as the action cannot be undone.  
You have turned the ability to tag specific content types off.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Structure

Control the layout of the Developer Portal, such as configuring the front page, adding and changing blocks, and displaying Products and APIs in categories.

To learn how to control the layout of the content on your Developer Portal site, use the following topic links.

- [Configuring the front page](#)  
You can configure the Developer Portal front page.
- [Adding a menu](#)  
You can add a menu to the Developer Portal, and define the items that are included in the menu.
- [Adding and changing the blocks displayed on Developer Portal pages](#)  
You can add new blocks, and modify the existing blocks that are displayed on Developer Portal pages.
- [Changing the front page banner block](#)  
You can change the front page banner content and image that is displayed on the home page when a user first logs in to the Developer Portal.
- [Changing the front page Featured Content block](#)  
You can display featured Products or APIs on the front page of your Developer Portal by configuring the Featured Content block.
- [Changing the items in the main menu](#)  
You can change the items in the main menu that is displayed on all pages in the Developer Portal.
- [Changing the ratings display](#)  
You can change the appearance of the content ratings in the Developer Portal.
- [Displaying APIs and Products in categories](#)  
You can create taxonomies in the Developer Portal, which dictate the categorization of APIs and Products, and how they are displayed. The taxonomies that you define can be used by API developers to categorize APIs and Products in YAML files.
- [Implementing an image carousel](#)  
You can customize your Developer Portal home page to display images in a carousel. The image carousel provides your Developer Portal home page with a continuous slide show that is customized to your specifications.
- [Making your application's image public](#)  
You might want to display your application's image in an oauth authentication page to reassure customers that they are authenticating the correct application in the Developer Portal. That authentication uses a URL to the image that is stored on the Developer Portal. However, by default, those images are private unless you are logged in, which doesn't work if you want to display it in an oauth page. The solution is for the admin to change the `application_image` field to use public instead of private.
- [Providing navigation by tag hierarchy](#)  
You can provide the ability for users in the Developer Portal to navigate by using tag hierarchy.
- [Using the Views module in the Developer Portal](#)  
By using the Views module, you can fetch content from your Developer Portal site, and present it to users in different formats such as lists, graphs, and tables.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the front page

You can configure the Developer Portal front page.

---

## Before you begin

You must have administrator access to complete this task.

---

## About this task

Create a customized welcome page for your users. Note that blocks that are placed on the front page of the Developer Portal are visible to all users, regardless of whether the blocks have access restrictions placed on them. The visibility of blocks to all users also extends to the Featured Content block.

---

## Procedure

To configure the front page, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure, then click Pages.
3. Alongside the welcome entry, click Edit.
4. Use the options provided to configure the front page.  
Here you can edit the page information, access, and configuration of the welcome page.
5. Click Update and save to save your changes.

---

## What to do next

You can create a custom page. For more information, see [Creating custom pages](#).

- [Disabling blogs](#)  
You can disable blogs on your Developer Portal.
- [Deleting forums from the front page](#)  
You can delete the help information on forums from the front page of your Developer Portal.

## Related tasks

- [Changing the front page banner block](#)
- [Changing the front page Featured Content block](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Disabling blogs

You can disable blogs on your Developer Portal.

## Before you begin

You must have administrator access to complete this task.

Before any changes are made, **Blogs** show as an option on the front page like this:



## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure > Menus from the administrator dashboard.
3. In the OPERATIONS column, click Edit menu for **Main navigation**.
4. Clear Enabled for the **Blogs Menu link**, then click Save.

**Title \***

Machine name: main

### Administrative summary

### Menu language

| MENU LINK    | ENABLED                             | OPERATIONS                          |
|--------------|-------------------------------------|-------------------------------------|
| Home         | <input checked="" type="checkbox"/> | <input type="button" value="Edit"/> |
| API Products | <input checked="" type="checkbox"/> | <input type="button" value="Edit"/> |
| Apps         | <input checked="" type="checkbox"/> | <input type="button" value="Edit"/> |
| Blogs        | <input type="checkbox"/>            | <input type="button" value="Edit"/> |
| Forums       | <input checked="" type="checkbox"/> | <input type="button" value="Edit"/> |
| Support      | <input checked="" type="checkbox"/> | <input type="button" value="Edit"/> |

## Results

You removed the **Blogs** option from the front page. The options on the front page of your Developer Portal now look like this:





**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Deleting forums from the front page

You can delete the help information on forums from the front page of your Developer Portal.

### Before you begin

You must have administrator access to complete this task.

Before any changes are made, the front page help bar looks like this:



### Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Block layout, Custom block library.
4. In Block description, enter Get help, then click Apply.

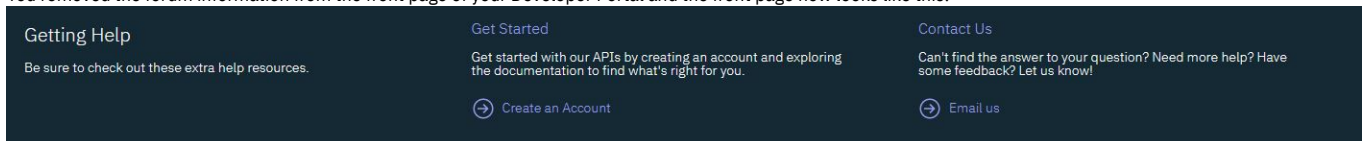
Block description:  Block type:

| BLOCK DESCRIPTION             | BLOCK TYPE  | UPDATED            | OPERATIONS                          |
|-------------------------------|-------------|--------------------|-------------------------------------|
| <a href="#">Get Help [en]</a> | Basic block | 05/07/2020 – 19:52 | <input type="button" value="Edit"/> |

5. In the OPERATIONS column, click Edit for **Get Help**.
6. In the **Body** section, delete the forum information, then click Save.

### Results

You removed the forum information from the front page of your Developer Portal and the front page now looks like this:



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Adding a menu

You can add a menu to the Developer Portal, and define the items that are included in the menu.

### Before you begin

You must have administrator access to complete this task.

### Procedure

To add a menu, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure, Menus.
3. Click Add menu, provide a title, and an optional summary and menu language, and click Save.
4. To add items to your menu, click Add link, and configure the menu link details that you require.
5. Click Save to save your changes.  
A new block for your menu is created automatically, and your menu is now in the **Menus** list.

## What to do next

You can use block configuration options to control which pages your menu is displayed on, and where it is positioned. You can use your menu with, or instead of, other menus available within the site. For more information, see [Changing the blocks displayed on Developer Portal pages](#).

You can also update an existing menu, or change your menu. For more information, see [Changing the items in the main menu](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Adding and changing the blocks displayed on Developer Portal pages

You can add new blocks, and modify the existing blocks that are displayed on Developer Portal pages.

### Before you begin

You must have administrator access to complete this task.

### About this task


Blocks are boxes of content that are rendered into an area, or region, of a page. The content can be custom HTML, Github Markdown, or plain text, and you can specify the appearance, shape, size, position, and also configure which pages a block appears on by editing its visibility settings. You can also modify an existing block to change its appearance, shape, size, position, and visibility.

- [Add a new block](#)
- [Edit the content of an existing block](#)
- [Modify the layout or visibility of an existing block](#)

### Procedure

To add a new basic block, complete the following steps.

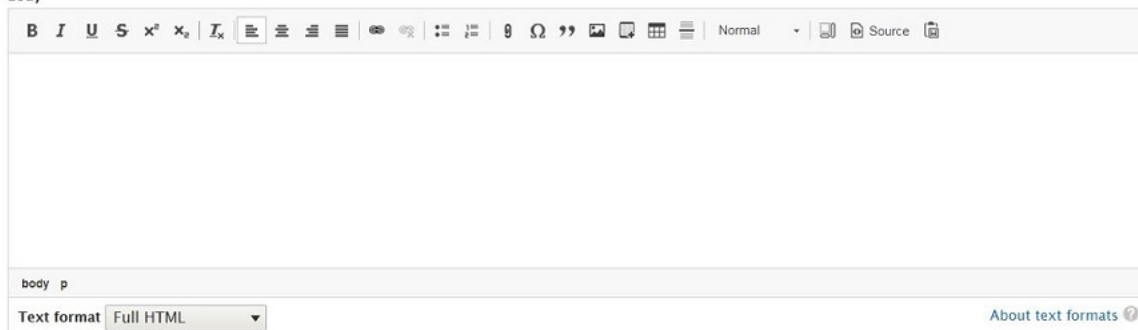
1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure, then click Block layout.  
All of the blocks that are in the system are displayed by theme. From the Block layout page you can assign blocks to a region, and control the order of the blocks within the region. Blocks are positioned on a per-theme basis, so if you enable more than one theme on your site, you can place blocks differently for each theme.
3. Click Place block, Add custom block.  
The Add custom block page is displayed.
4. Complete the Block description field.
5. Enter the content that you require for your block into the Body section.

You can set the text format to Full HTML, Basic HTML, Restricted HTML, or Github Markdown. You can use the edit icons at the top of the Body section to configure your content. Click the Source icon  to enter code source, such as HTML.

Block description \*

A brief description of your block.

Body



The screenshot shows the 'Body' section of a block configuration page. It features a rich text editor toolbar with various icons for bold, italic, underline, strikethrough, bulleted list, numbered list, link, unlink, and other text formatting options. The toolbar also includes a 'Normal' dropdown menu and a 'Source' icon. Below the toolbar is a large text area for entering content. At the bottom of the section, there is a 'Text format' dropdown menu currently set to 'Full HTML' and a link for 'About text formats'.

6. Click Save.  
The new custom block is saved and the Configure block page is displayed.
7. Complete the block Title field.
8. In the Visibility section, specify how you want the block to be displayed.  
You can specify the content types, configure the language settings, restrict the block to show only on pages that display specific content types, set which pages the block is displayed on, and restrict which user roles the block is visible to.

#### Visibility

|  |   |
|--|---|
| <b>Content type</b>                    | <b>When the user has the following roles</b><br><input type="checkbox"/> Anonymous user<br><input type="checkbox"/> Superuser<br><input type="checkbox"/> Authenticated user<br><input type="checkbox"/> Forum Moderator<br><input type="checkbox"/> Content Author<br><input type="checkbox"/> Administrator |
| <b>Language</b><br>Not restricted      |   |
| <b>Content types</b><br>Not restricted |   |
| <b>Pages</b><br>Not restricted         |   |
| <b>Roles</b><br>Not restricted         |   |

9. In the Region section, select the region where the block should be displayed.
10. Click Save block.

The new custom block is added to your Developer Portal site.

To edit the content of an existing block, from the Block layout page complete the following steps.

11. Click the Custom block library tab (or select Structure > Block layout > Custom block library from the administrator dashboard).
12. Click Edit alongside the required block.
13. Change the block description and body content as required.
14. Click Save to save your changes.

To modify the layout or visibility settings of an existing block, from the Block layout page complete the following steps.

15. To change the region in which a block is positioned, select the required Region from the drop-down list next to the required block.
16. To change the vertical sort-order of a block within a region, drag the block to the required position.
17. To edit the visibility settings of an existing block, click Configure from the Operations drop-down menu next to the required block.
18. To enable or disable an existing block, select Enable or Disable from the Operations drop-down menu next to the required block.
19. Click Save blocks to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing the front page banner block

You can change the front page banner content and image that is displayed on the home page when a user first logs in to the Developer Portal.

### Before you begin


You must have administrator or content author access to complete this task.

### About this task

You can change the front page banner by editing the Welcome Banner block in the custom block library.

### Procedure

To change the front page banner block, complete the following steps.

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure > Block layout.
3. Select the Custom block library tab.
4. Click edit against the Welcome Banner block entry.  
The Edit Basic block Welcome Banner page is displayed.
5. Edit the Body section with the content and image that you require.  
You can set the text format to Full HTML, Basic HTML, Restricted HTML, or Github Markdown. You can use the edit icons at the top of the Body section to configure your content. Click the Source icon  to view and edit the code source, such as HTML.
6. Configure the language and translation settings as required.
7. Click Save to save your changes.

### Results

You successfully updated the front page banner block.

### Related tasks

- [Configuring the front page](#)

- [Adding and changing the blocks displayed on Developer Portal pages](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing the front page Featured Content block

You can display featured Products or APIs on the front page of your Developer Portal by configuring the Featured Content block.

### Before you begin

You must have administrator access to complete this task.

### About this task

You can configure the Featured Content block on the front page of your Developer Portal. By default this block displays a list of the most recently created Products, but you can configure the block to display Products or APIs, and select how the featured content is chosen.

### Procedure

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Structure > Pages.
4. Click Edit against the welcome page.  
The edit menu for the Welcome page is displayed.
5. Select Panels > Content from the left navigation column, and then click Edit against the Featured Content block.  
The Edit block page for the Featured Content block is displayed.

The screenshot shows the 'Edit block' configuration page for the Featured Content block. The page has a dark header with the title 'Edit block' and a close button. Below the header, there is a paragraph of instructions: 'Configure the node selection settings for this Featured Content block. For each node the block will display its image, title and summary. If no summary is present then a truncated form of the description field will be used instead. To modify the content shown for a node either Edit that node in the portal or edit the YAML document for that node in the API Manager. Note that the node selection is per user, and so depending on the visibility settings, different users may see different nodes.'

The configuration form includes the following sections:

- Block description:** 'Featured Content block'
- Title \*:** A text input field containing 'Featured Content'. Below it is a checkbox labeled 'Display title' which is unchecked.
- Node type \*:** A dropdown menu set to 'Product'. Below it is the text 'Feature APIs or Products?'.
- Number of tiles to display \*:** A spinner control set to '3'. Below it is the text 'How many tiles should be shown?'.
- Node selection algorithm \*:** A dropdown menu set to 'Most recently created'. Below it is the text 'Select how the featured content should be chosen. For example, based on creation or modification time, name, or random.'
- Custom nodes:** An empty text input field. Below it is the text 'Manually specify the nodes to feature. ',' separated. (This field is only used if using 'Custom' node selection.)'
- Region \*:** A dropdown menu set to 'Content'.
- Update block:** A blue button at the bottom of the form.

6. Configure the Feature Content block as required.  
You can choose to display the title or not, select whether to use Products or APIs, for your node type. You can choose how many tiles to display, and select which algorithm is used to select the node. If you start typing in the **Custom nodes** field, you can choose from the options you are given.
7. Click Update block, then Update and save, to save your changes.

### Results

You successfully configured the Featured Content block on the front page of your Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Changing the items in the main menu

You can change the items in the main menu that is displayed on all pages in the Developer Portal.

## Before you begin

You must have administrator access to complete this task.

## About this task

The following screen capture shows an example of a main menu:



However, you can add new menu items, remove items, and change the item order.

## Procedure

To change the items in the main menu, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Structure > Menus > Main navigation.

[Home](#) > [Administration](#) > [Structure](#) > [Menus](#)

[+ Add link](#)

**Title \***  
 Machine name: main

**Administrative summary**

**Menu language**

| MENU LINK                      | ENABLED                             | OPERATIONS             |
|--------------------------------|-------------------------------------|------------------------|
| <a href="#">+ Home</a>         | <input checked="" type="checkbox"/> | <a href="#">Edit</a> ▾ |
| <a href="#">+ API Products</a> | <input checked="" type="checkbox"/> | <a href="#">Edit</a> ▾ |
| <a href="#">+ Apps</a>         | <input checked="" type="checkbox"/> | <a href="#">Edit</a> ▾ |
| <a href="#">+ Blogs</a>        | <input checked="" type="checkbox"/> | <a href="#">Edit</a>   |
| <a href="#">+ Forums</a>       | <input checked="" type="checkbox"/> | <a href="#">Edit</a> ▾ |
| <a href="#">+ Support</a>      | <input checked="" type="checkbox"/> | <a href="#">Edit</a>   |

[Save](#)

4. Use the options provided to change the menu items.
5. Click Save to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Changing the ratings display

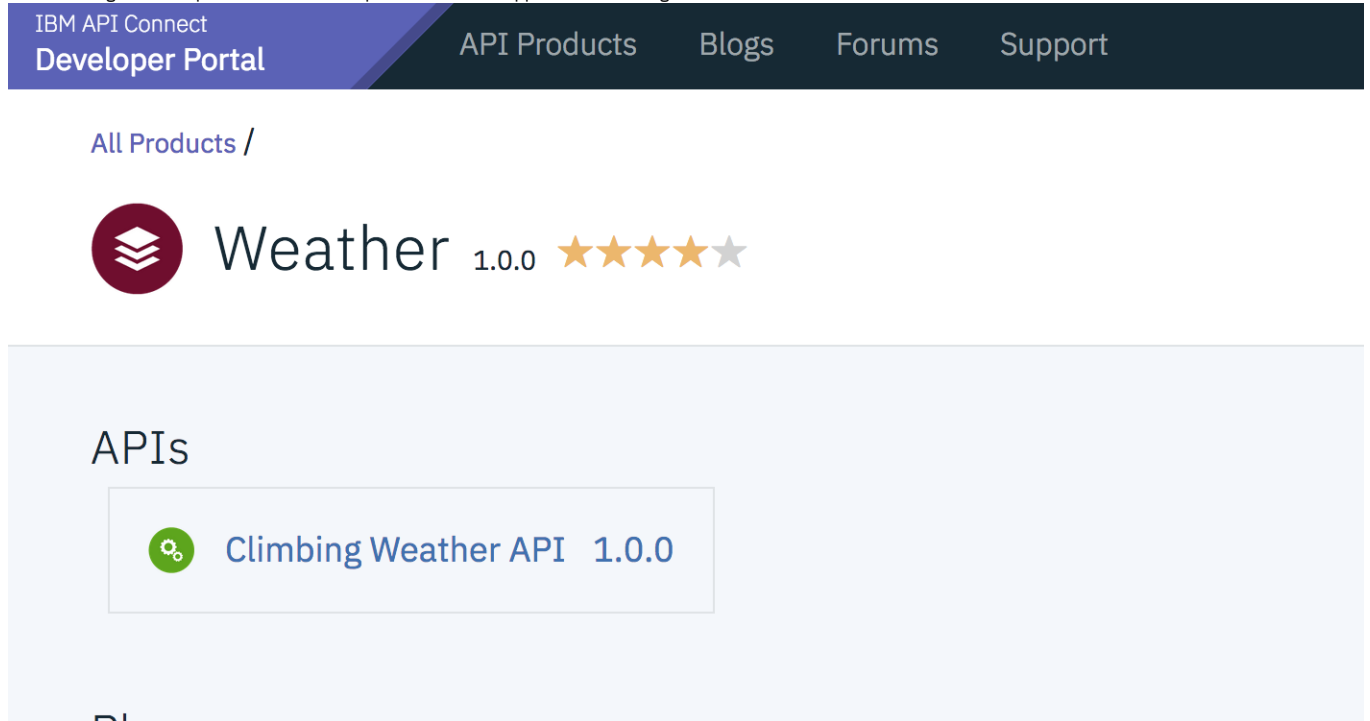
You can change the appearance of the content ratings in the Developer Portal.

## Before you begin

You must have administrator access to complete this task.

## About this task

The following screen capture shows an example of the default appearance of ratings:



You can change this display if you want to.

## Procedure

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Structure > Content types.
4. Click the Manage fields drop-down against the content type you want to change, for example **Product**.
5. Click Edit on the **Rating** row. Then, select the Field settings tab.

These settings apply to the *Rating* field everywhere it is used.

**Vote type \***

Normal vote ▾

**Vote plugin \***

Fivestar rating ▾

**Allowed number of values**

Limited ▾ 5

**Save field settings**

6. Change the style settings as required and click Save field settings.  
Here is an example of an alternative Ratings style:

[All Products /](#)

Weather 1.0.0

good

## APIs



Climbing Weather API 1.0.0

## Results

You successfully changed the ratings display for a content type in your Developer Portal.

Note: There are different view modes, and the settings would need to be changed for each view mode for each content type.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Displaying APIs and Products in categories

You can create taxonomies in the Developer Portal, which dictate the categorization of APIs and Products, and how they are displayed. The taxonomies that you define can be used by API developers to categorize APIs and Products in YAML files.

You might want to create taxonomies to control which set of APIs and products can and cannot be used in the Developer Portal.

By default, all taxonomies must be created in the Developer Portal. For more information, see [Managing tags in the Developer Portal](#).

After you create your taxonomies for your APIs and Products, API developers can assign content to those taxonomies manually in the Developer Portal, or by using categories in YAML files.

However, if you create categories for your APIs and Products in the API Manager or API Manager UI, you can have your APIs and Products that are displayed in those categories in the Developer Portal. For more information on how to create categories for your APIs and Products in the API Manager or API Manager UI, see [Organizing your APIs and Products into categories](#). For information on how to have your APIs and Products that are displayed in those categories in the Developer Portal, see [Enabling dynamic category creation](#).

You can add more Developer Portal taxonomies to the categories that are defined in the API Manager or API Manager UI.

- [Enabling dynamic category creation](#)

You can display APIs and Products in pre-defined categories in the Developer Portal. You can define the categories that the APIs and Products are displayed in, in the API Manager or API Manager UI.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Enabling dynamic category creation

You can display APIs and Products in pre-defined categories in the Developer Portal. You can define the categories that the APIs and Products are displayed in, in the API Manager or API Manager UI.

## Before you begin

You must have administrator access to complete this task.

Your APIs and Products must be categorized in the API Manager or API Manager UI, and they must be published. For more information, see [Organizing your APIs and Products into categories](#).

## About this task

---

By organizing your APIs and Products into categories, you can provide a hierarchical display for your APIs and Products in the Developer Portal.

Note: Organizing APIs and Products into a hierarchical view in the API Manager UI is different from tagging in the Developer Portal.

The categories that are defined in the API Manager or API Manager UI can be overridden by creating taxonomies in the Developer Portal. For more information, see [Displaying APIs and Products in categories](#).

## Procedure

---

1. To enable the Developer Portal to display the categories for your APIs and Products:
  - a. On the administrator dashboard, click Configuration > System > IBM API Connect.
  - b. In the Categories section, select the Create taxonomies for categories if they do not already exist check box.
  - c. Click Save configuration.
2. To enable the Developer Portal to display the change to the categories you made in step 1, you must republish your APIs and products.

## Results

---

You enabled the Developer Portal to display the categories that you defined for your APIs and Products. You can see the categories that you defined, including the number of APIs and Products that are in each category, by selecting your APIs and Products in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Implementing an image carousel

You can customize your Developer Portal home page to display images in a carousel. The image carousel provides your Developer Portal home page with a continuous slide show that is customized to your specifications.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

By implementing an image carousel, you can replace the default welcome banner or any previously set images.

## Procedure

---

1. Create a content type of Image for your carousel.
  - a. Click Structure > Content types > Add content type.
  - b. Enter the name of your content type in the Name field.  
For example, *Slide show picture*.
  - c. Optional: Specify the description of your content type in the Description field.  
For example, *A picture to display in the slideshow*.
  - d. Click Save and manage fields for your newly created content type.
  - e. Click Add field.
  - f. Select Image for Add a new field, and enter a label for your field.  
For example, *slide image*.
  - g. Click Save and continue.  
The Field settings page for your new field is displayed.
  - h. In the DEFAULT IMAGE section, click Choose file to assign the image that you want as your default image for your slide show if no other images are found.
  - i. Optional: Complete the Alternative text and Title fields.
  - j. Click Save field settings.
  - k. In the Manage fields tab for your newly created content type of *Slide show picture*, delete the `Body` field by clicking Delete on the OPERATIONS drop-down menu. Click Delete again to confirm.
2. Upload the images that you want to appear in your carousel.
  - a. Click Content > Add content, then click the content type that you added in step 1.
  - b. Enter a title for your image in the Title field.  
For example, if you had an image of a lighthouse, you can label the content as *Lighthouse*.
  - c. Click Choose file under Slide image and select an image for your slide show. Complete the Alternative text field with some text to use when the image cannot be loaded.
  - d. Click Save.
  - e. Repeat sub-steps 2.a to 2.d until you are satisfied with the content of the slides for your carousel.
3. Create a view for your carousel.
  - a. Click Structure > Views > Add new view.
  - b. Complete View name.



- For example, Slide show.
- c. Select the Create a block check box in the BLOCK SETTINGS section.
  - d. Select Slick Carousel for the display format.
  - e. Enter the number of slides you have for your carousel in the Items per block field, then click Save and edit.  
The Edit page for your slide show view is displayed.
  - f. Click the title of the view and enter <none> into the text field. Click Apply.
  - g. In the FIELDS section click Add, select the check box for slide image, and click Add and configure fields.
  - h. Ensure the check box for Create a label is not selected, and click Apply.
  - i. In the FIELDS section, click Content: Title, Remove.
  - j. In the FORMAT section, click Settings for Slick Carousel, and then set Skin Main to Default.
  - k. Scroll down to the CAPTION FIELDS section, and select the Content: slide image and Override main optionset check boxes.
  - l. In the OVERRIDABLE OPTIONS section, select the Autoplay, Dots, and Draggable check boxes.
  - m. Click Apply.
  - n. In the FILTER CRITERIA section, click Content: Published, Remove.
  - o. Click Save to save the Slide show view.
4. Configure your home page to host the carousel.
    - a. Click Structure, then click Pages.
    - b. Click edit for the welcome page, then click content in the Panels section.
    - c. To remove the current welcome banner, click the down-arrow under the OPERATIONS column for the Welcome Banner, and select Delete. Click Delete again to confirm.
    - d. To add your carousel, click Add new block and select the slide show view you created in step 3. Deselect Display title and click Add block.
    - e. Click Update and save.

---

## Results

Your Developer Portal front page now has a slide show carousel.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Making your application's image public

You might want to display your application's image in an oauth authentication page to reassure customers that they are authenticating the correct application in the Developer Portal. That authentication uses a URL to the image that is stored on the Developer Portal. However, by default, those images are private unless you are logged in, which doesn't work if you want to display it in an oauth page. The solution is for the admin to change the `application_image` field to use public instead of private.

---

## Before you begin

You must have administrator access to complete this task.

---

## About this task

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Structure, Content types, Application, Manage fields
4. Navigate to the Image label, then in the Edit drop down, select **storage settings**.
5. Click **Public files** for **Upload destination**.
6. Click **Save field settings**.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Providing navigation by tag hierarchy

You can provide the ability for users in the Developer Portal to navigate by using tag hierarchy.

---

## Before you begin

You must have administrator access to complete this task.

---

## About this task

The Categories block provides the ability to navigate by tag hierarchy. The Categories block is disabled by default, so you need to enable the block and then configure it to your requirements.

---

## Procedure

To provide navigation by tag hierarchy, complete the following steps:

1. Click Extend, in the filter box enter `Hierarchical Taxonomy Menu`.
2. Select **Hierarchical Taxonomy Menu**, then click Enable.
3. Click Structure\_>Block layout.

All of the blocks in the connect\_theme are displayed.

4. Scroll to the region that you want to add the block to, for example Collapse left, click Place block.
5. Find the **Hierarchical Taxonomy Menu** block in the list and click Place block.
6. Complete any configuration tasks that you require for the **Collapse left** block, including selecting the correct taxonomy to use.
7. Click Save block to save your changes.

You can add new blocks, and modify the existing blocks that are displayed on Developer Portal pages. For more information, see [Adding and changing the blocks displayed on Developer Portal pages](#).

---

## Related tasks

- [Editing tags for a specific item](#)
- [Managing tags in the Developer Portal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using the Views module in the Developer Portal

By using the Views module, you can fetch content from your Developer Portal site, and present it to users in different formats such as lists, graphs, and tables.

The Views module is a powerful SQL query builder that can access almost all the information in your Developer Portal site database and display it in any format.

There are many uses cases for views, but the following list shows some of the common ones:

- You want to display some of the content differently, such as Products, APIs, and applications.
- You want to display a block with the five most recent posts of some kind.
- You want to change the way that articles are displayed.
- You want to provide an unread forum posts list.
- You want to provide a monthly archive of posts.

For information about how to create a view in the Developer Portal, see the following topic.

- [Creating views in the Developer Portal](#)  
You can create new views in the Developer Portal, such as content lists of Products, APIs, and applications, by using the Views module.
- [Configuring the search results view in the Developer Portal](#)  
You can configure the way that the search results are displayed in the Developer Portal.
- [Configuring the default number of items in a view list in the Developer Portal](#)  
You can configure the default number of items that are displayed in a view list in the Developer Portal.

---

## Related information

- [Views module in Drupal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating views in the Developer Portal

You can create new views in the Developer Portal, such as content lists of Products, APIs, and applications, by using the Views module.

---

## Before you begin

You must have administrator access to complete this task.

---

## About this task

Creating views in the Developer Portal enables you to control the presentation of specific content for users. For more information about views, see [Using the Views module in the Developer Portal](#).

---

## Procedure

To create a view, complete the following steps.

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click **Manage** to display it.
3. On the administrator dashboard, click **Structure > Views > Add new view**.  
The **Add view** page is displayed.
4. Complete the **View name** and, optionally, the **Description**.
5. Complete the **VIEW SETTINGS** section with details of what you want your view to show, and how you want it to be sorted.  
For example, to create a view of APIs that are sorted by title, select **Show Content**, of type **API**, and sorted by **Title**.
6. Select whether to **Create a page**, **Create a block**, or both, for your view.  
A page is a full-screen page in the Developer Portal. Whereas a block would be a site block that you can then place on whatever pages you want to show a smaller view of the content, for example the 10 newest APIs.
7. Complete the options that you require for your view. For example, the display format, the number of items per page or block, whether to include a pager, whether to create a menu link, and whether to include an RSS feed.
8. Click **Save and edit**.  
Your new view opens in the **Edit** tab.
9. Refine your view further.  
Here you can further modify the display of your view or add new displays. More options include formatting of the view, what fields to include in a table or grid format, filtering and sorting criteria, add header or footer details, as well as advanced options around contextual filters, relationships, and behavior when there are no results to display. The **Edit** tab also includes a preview area, so you can check your view before publishing.
10. Click **Save** to make your changes permanent.  
When saved, your new view is visible in the **Structure > Views** pane.

---

## Results

You created a new view in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the search results view in the Developer Portal

You can configure the way that the search results are displayed in the Developer Portal.

---

### Before you begin

You must have administrator access to complete this task.

---

### About this task

In the Developer Portal, the search results are displayed as a view, and so you can control the presentation of this search content. For more information about views, see [Using the Views module in the Developer Portal](#).

---

### Procedure

To configure the search results view, complete the following steps.

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click **Manage** to display it.
3. On the administrator dashboard, click **Structure > Views**.  
The list of views is displayed.
4. Find the **Search content** view in the list, and click **Edit**.
5. Edit the **Search content** view settings as required.  
Here you can modify the display of the view or add new displays. For example, you can configure the formatting of the view, the filtering and sorting criteria, add header or footer details, as well as more advanced options around contextual filters, relationships, and behavior when there are no results to display. There is also a preview area so you can check the view before publishing.
6. Click **Save** to make your changes permanent.

---

## Results

You updated the view of the search results in the Developer Portal.

---

### What to do next

You can configure the search index and server that are used to generate the search results. For more information, see [Configuring search indexes and servers](#).

---

### Related information

- [Drupal Views module](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the default number of items in a view list in the Developer Portal

You can configure the default number of items that are displayed in a view list in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

In the Developer Portal, you can manage how many items in a list are shown for each page on a view. For more information about views, see [Using the Views module in the Developer Portal](#).

### Procedure

---

To configure the search results view, complete the following steps.

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Structure > Views.  
The list of views is displayed.
4. In the list, find the view name that you want to update, for example, Products, then click Edit.
5. Click Pager.
6. Select Display a specified number of items, click Apply.
7. Set the Items per page as required, click Apply.
8. Click Save to make your changes permanent.

### Results

---

You updated the default number of items in a view of the search results in the Developer Portal.

### What to do next

---

The Views module is a powerful SQL query builder that can access almost all the information in your Developer Portal site database and display it in any format. For more information, see [Using the Views module in the Developer Portal](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuration

As an administrator, you can control multiple aspects of the Developer Portal configuration, including managing security and general configuration tasks.

To learn how to configure specific aspects of your Developer Portal site, use the following section links.

- [Configuring content moderation](#)  
By configuring content moderation, you can configure a review and approval process for content types in the Developer Portal. You can specify which roles can perform which actions by assigning them the appropriate permissions.
- [General configuration tasks](#)  
You can perform general configuration tasks in the Developer Portal as an administrator.
- [Managing Developer Portal security](#)  
You can manage multiple security elements in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring content moderation

By configuring content moderation, you can configure a review and approval process for content types in the Developer Portal. You can specify which roles can perform which actions by assigning them the appropriate permissions.

## Before you begin

An example use case for configuring content moderation would be for a professional documentation writer to review an APIs documentation before it is published on the Developer Portal. They might, for example, notify the API Developer of typographical errors, suggest improvements to operation descriptions, or attach an image file. Edits might then be made before the documentation was republished to the API and then published to the Developer Portal.

You must have a Developer Portal enabled, and you must have administrator access to complete this task.

## About this task

By using content moderation, you can configure the workflow of your Developer Portal site so that you can moderate content types, such as APIs and Products. However, you cannot moderate Consumer organizations or applications.

## Procedure

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Enable the required modules:
  - a. On the administrator dashboard, click Extend.
  - b. In the search bar enter **workflows**, then select the Workflows module and click Enable.
  - c. In the search bar enter **content moderation**, then select the Content Moderation module and click Enable.
4. Navigate to Configuration > Workflow > Workflows
5. Check that **Editorial** is listed as a workflow. If it isn't listed:
  - a. Click + **Add workflow**.

**Label \***

Machine name: editorial [\[Edit\]](#)

**Workflow type \***

Content moderation ▼

Save

- b. Enter **Editorial** as the label.
  - c. Select the workflow type **Content moderation**.
  - d. Click **Save**.
6. Set up the workflow:
    - a. Click **Edit** on the **Editorial** workflow.
    - b. Click the **Add a new state** link.

**Label \***

Machine name: editorial

▼ STATES

| STATE                              | OPERATIONS                            |
|------------------------------------|---------------------------------------|
| <input type="checkbox"/> Draft     | <input type="button" value="Edit"/>   |
| <input type="checkbox"/> Published | <input type="button" value="Edit"/>   |
| <input type="checkbox"/> Review    | <input type="button" value="Edit"/> ▼ |

[Add a new state](#)

- c. Enter the value **Review** in the **State label** field, leave the **Published** and **Default revision** check boxes cleared then click Save.

**State label \***

Review

Machine name: review [Edit]

 Published

When content reaches this state it should be published.

 Default revision

When content reaches this state it should be made the default revision; this is implied for published states.

**Save**

- d. Drag the **review** state to between the **Draft** and **Published** states then click 'Save'. If the drag handles are not visible, then click the **Hide row weights** link in the **STATES** section to make them appear.

**Label \***

Editorial

Machine name: editorial

▼ STATES

⚠ You have unsaved changes.

| STATE                              | OPERATIONS                          |
|------------------------------------|-------------------------------------|
| <input type="checkbox"/> Draft     | <input type="button" value="Edit"/> |
| <input type="checkbox"/> Review*   | <input type="button" value="Edit"/> |
| <input type="checkbox"/> Published | <input type="button" value="Edit"/> |

[Add a new state](#)

- e. Click the **Add a new transition** link.
- f. Enter the value **Review** in the **Transition Label** field, tick the **Draft** check box in the **From** section and click **Review** in the **To** section then click Save.
- g. Drag the **Review** transition to between the **Create New Draft** and **Publish** transitions, then click Save.
- h. Click the **Edit** operation on the **Create New Draft** transition.
- i. In the **From** check box list, untick **Published** and tick **Review** then click Save.
- j. Click the **Edit** operation on the **Publish** transition.
- k. In the **From** check box list, untick **Draft** and tick **Review** then click Save.
- l. Click **Select** for the **Content types** item in the **THIS WORKFLOW APPLIES TO** section.
- m. Select the **Product** content type then click Save.
7. Set up the permission schema for the workflow:
- Click **People**, **Roles**, **Add a new role**.
  - In the **Role Name** field, enter the value **Editor** then click Save.
  - Drag the **Editor** role to between the **Content Author** and **Administrator** roles then click Save.
  - Click the **Permissions** tab and scroll down to the **Content Moderation** permission section.
  - For the **EDITOR** role, select all the permissions in the **Content Moderation** section.
  - For the **CONTENT AUTHOR** role, select the permissions **Editorial workflow: Use Create New Draft transition**, **View any unpublished content**, and **View the latest version** then click Save permissions.
  - Ensure that the **EDITOR** and **CONTENT AUTHOR** roles have the permission **Use the administration toolbar**.
8. Assign the permissions to users in the portal. If you do not already have some registered users in your portal, create a new consumer organization and invite two new users to it.
- Click **People**.
  - Tick the check box next to one of the users in the list, but not the admin user, then from the **Action** drop-down select **Add the Content Author role to the selected user(s)** then click **Apply to selected items**.
  - Tick the check box next to another one of the users in the list, but not the admin user, then from the **Action** drop-down select **Add the Editor role to the selected user(s)** then click **Apply to selected items**.

## Results

You successfully configured content moderation on your Developer Portal. You can now use the API manager user interface to publish a product to the portal, and transition the product from draft to review state.

- [Editing, reviewing, and publishing moderated content](#)

If you have published content from API Manager UI to your Developer Portal or created content within the Developer Portal, you can edit and review the content through the Moderated content page before you publish it in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Editing, reviewing, and publishing moderated content

If you have published content from API Manager UI to your Developer Portal or created content within the Developer Portal, you can edit and review the content through the Moderated content page before you publish it in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

You must have content that is published to the Developer Portal from the API Manager UI.

Your role must have the necessary permissions that enable you to perform the tasks. For more information on assigning the necessary tasks, see [Configuring content moderation](#).

### About this task

---

In addition to editing content through the content moderation page, you can specify whether the content needs further review, or even publish the content if you have the permission.

### Procedure

---

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. On the administrator dashboard, click Content.
4. Click the Moderated content tab.
5. If you have the necessary permissions to edit your content type, for example, you are a **Content author**:
  - a. Click the Title of the content type that you want to edit, then click the Edit.
  - b. Edit your content to your specification, then click Save.  
You are redirected to the **View** tab.
  - c. When you have finished editing your content type on the **edit** tab, ensure that the **Review** state is selected from the **Change to** drop down, then click Save.
6. If you have the necessary permissions to review and publish your content type, for example, you are a **Reviewer**:
  - a. On the administrator dashboard, click Content.  
The content page is displayed listing all content.
  - b. Click the Moderated content tab.
  - c. Click the Title of the content type that you want to review.
  - d. Optional: If the content that you are reviewing requires editing, click Edit, apply the changes to your content type, then click Save.  
You are redirected to the View tab.

### Results

---

You edited, reviewed, and published your draft content type.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## General configuration tasks

You can perform general configuration tasks in the Developer Portal as an administrator.

To learn how to configure specific aspects of your Developer Portal site, use the following topic links.

- [Adding a page load progress indicator](#)  
You can extend your Developer Portal site by adding a page load progress indicator. This option might be useful on servers that have a slow connection to the API Manager, or for customers that are using OIDC and have slow response times from their OIDC provider.
- [Adding custom fields to user records](#)  
You can add custom fields to user records in the Developer Portal.
- [Checking the site status report](#)  
You can check the site status report in the Developer Portal.

- [Clearing the server caches](#)  
You can clear the server caches from within the Developer Portal.
- [Configuring a proxy](#)  
You can configure the proxy settings in your Developer Portal.
- [Configuring cron to run scheduled tasks](#)  
You can configure cron to run scheduled tasks in the Developer Portal.
- [Configuring search indexes and servers](#)  
You can configure search indexes and servers in the Developer Portal by using the Search API module.
- [Configuring the date and time](#)  
You can configure the date and time in the Developer Portal.
- [Configuring the settings of a Consumer organization](#)  
You can configure the settings of your Consumer organization from within your Developer Portal.
- [Configuring the site default timezone](#)  
You can configure the site default timezone in the Developer Portal.
- [Configuring the site error handling](#)  
You can configure the site error handling in the Developer Portal.
- [Configuring which buttons are displayed in the WYSIWYG rich text editor](#)  
You can configure which buttons are displayed in the WYSIWYG rich text editor in the Developer Portal.
- [Configuring which languages are available](#)  
You can configure which languages are available in the Developer Portal.
- [Customizing user account settings](#)  
You can customize the user account options that are displayed to users when they register and use the Developer Portal.
- [Disabling languages](#)  
You can extend your Developer Portal site by managing which languages are available for use on the site.
- [Disabling test tool restrictions](#)  
The default OAuth provider contains security mechanisms that prevent the ability to obtain authorization tokens in the Developer Portal test tool when you use Implicit or Authorization Code grant types. If your OAuth provider allows those actions to function correctly from the test tool, you can disable this option.
- [Enabling a cookie compliance banner](#)  
You can add a cookie compliance banner to your Developer Portal site by configuring the EU Cookie Compliance module.
- [Enabling code languages for code snippets](#)  
You can specify the languages that code snippets for APIs can be displayed in.
- [Enabling code snippets for SOAP APIs](#)  
By default, code snippets are only shown for REST APIs. You can enable code snippets for SOAP APIs.
- [Forcing new users to accept terms and conditions](#)  
You can extend your Developer Portal site by forcing new users to accept the terms and conditions before they are able to create accounts.
- [Hiding the admin registry on the login form](#)  
You can hide the admin registry as an option on the login form. The use case for this option might be a public facing Developer Portal where admin access shouldn't be shown to the customer.
- [Hiding the certificate in the header for APIs secured with mutual TLS](#)  
You can hide the certificate in the header for APIs secured with mutual TLS. By default, the `x-client-certificate` header is shown in an API when mutual TLS is configured. You can choose to turn off this option in your Developer Portal.
- [Importing and exporting taxonomies](#)  
You can import and export taxonomies in the Developer Portal.
- [Restricting access by IP address](#)  
You can restrict access to a role or a user by IP address in the Developer Portal.
- [Restricting or preventing access from search engines](#)  
You can restrict or prevent access to your Developer Portal from search engines. You might want to control the access if you have a development or test site.
- [Toggling the site in and out of maintenance mode](#)  
You can put the Developer Portal site into maintenance mode for short periods of time. During maintenance mode, only the administrator can access the site; all other users who enter the site URL get a maintenance message set by the administrator.
- [Viewing available updates](#)  
You can view available updates in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Adding a page load progress indicator

You can extend your Developer Portal site by adding a page load progress indicator. This option might be useful on servers that have a slow connection to the API Manager, or for customers that are using OIDC and have slow response times from their OIDC provider.

### Before you begin

You must have administrator access to complete this task.

### Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. In the administrator dashboard, click Extend.  
The List tab for the Extend page opens, and the list of installed modules is displayed. The list shows all the modules that are installed. The enabled modules are displayed with a selected check box. The disabled modules are displayed without a selected check box.
3. Enter `page_load_progress` into the search filter.  
The **Page Load Progress** content item is displayed in the list of content.



- By default, the **Page Load Progress** module is enabled. However, if not, select the check box and click Enable. You have now enabled the default settings for the **Page Load Progress** module.
- To change the default settings for the module, navigate to Configuration > User interface > Page Load Progress.
- Decide which settings you want to change, for example, **Time to wait before locking the screen**, and click Save.
 

Note: Changing the settings does not apply to the OIDC buttons. If it is enabled, the OIDC buttons use the page load process indicator, but do not allow any configuration.
- To enable the permissions, navigate to People > Permissions.
- Scroll down to **Page Load Progress**, check the boxes for anonymous and authenticated for **Use Page Load Progress**.

| Page Load Progress            |                                     |                                     |                                     |                                     |                                     |                                     |
|-------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Administer Page Load Progress | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Use Page Load Progress        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

- Click Save permissions.

## Results

You successfully enabled the **Page Load Progress**. Now, the page loading spinner is automatically added to the login and register forms for OIDC buttons.

## What to do next

You can customize the appearance of the modal page and spinner in CSS in a custom theme.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Adding custom fields to user records

You can add custom fields to user records in the Developer Portal.

## Before you begin

You must have administrator access to complete this task.

## About this task

Any data that is added to a custom field for user records will remain in the Developer Portal database.

## Procedure

- In the administrator dashboard, click Configuration > People > Account settings.
- Click the MANAGE FIELDS tab.
- Click Add field.
- Specify the type of data that your field can store by selecting the type from the Add a new field list.
- Enter a label for your field in the Label field.
- Optional: If you want to edit the machine name that is automatically created based on your label, click Edit and enter the new name.
- Click Save and continue.
- Configure any additional values for the new field then click Save field settings.
 

You have added a custom field to your user record.

## What to do next

You can configure the contents of your custom field depending on what type of data you specified it could store.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Checking the site status report

You can check the site status report in the Developer Portal.

## Before you begin

You must have administrator access to complete this task.

## Procedure

---

To check the site status report, complete the following steps:

1. Click Reports in the administrator dashboard.
2. Click Status report.
3. You can review information relating to the site status report from this view.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Clearing the server caches

You can clear the server caches from within the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

1. In the administrator dashboard, click Configuration, then in the Development section, click Performance.
2. Click Clear all caches.

You have cleared the server caches, and a message will be displayed stating: Caches cleared.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring a proxy

You can configure the proxy settings in your Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

The Developer Portal communicates with the management server by way of the consumer and product APIs. For a Developer Portal that is deployed externally to the management server zone, and does not have access to the consumer and product APIs, a proxy can be used.

## Procedure

---

To configure a proxy to use for communicating with management server APIs, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Configuration, > System > IBM API Connect in the administrator dashboard.
4. To configure a proxy, select **Enable Proxy Support**, and make the required changes.

**PROXY CONFIGURATION (EXPERIMENTAL)**

If a proxy is required to allow communication from the portal server to the consumer API.

Enable Proxy Support

**Proxy type**

CURLPROXY\_HTTP

Select what type of proxy to use, CURLPROXY\_HTTP is the default.

**Proxy URL**

Provide the URL of the proxy e.g. <http://proxyserver.domain.com:8080>.

**Proxy Authentication**

If the proxy requires authentication then provide the 'username:password'.

Save configuration

5. When your changes are complete, click Save configuration.

## Results

---

You successfully configured a proxy on your Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring cron to run scheduled tasks

You can configure cron to run scheduled tasks in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

To configure cron to run scheduled tasks, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the SYSTEM heading, click Cron.
3. Use the drop-down list for the module you want to configure and select Edit.
4. From the Run cron every drop-down menu, select the frequency of cron runs for this module.
5. Click Save.

You have configured cron to run scheduled tasks.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring search indexes and servers

You can configure search indexes and servers in the Developer Portal by using the Search API module.

### Before you begin

---

You must have administrator access to complete this task.

## About this task

---

The Search API module provides a generic framework for search capabilities within the Developer Portal, and you can customize the default database server and content index setup that is provided. By default, the search in the Developer Portal runs over the entire OpenAPI and WSDL documentation. It is not possible to restrict the search to specific sections of an Open API or WSDL document. The search can be configured to include any custom fields that were added to the Developer Portal, but it does not include any attached files.

The search results page is a view in the Developer Portal, and so it can also be configured. For more information, see [Configuring the search results view in the Developer Portal](#).

## Procedure

---

To configure the default database server and content index settings, complete the following steps.

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click **Manage** to display it.
3. Click **Configuration**, **Search** and **metadata**, **Search API** in the administrator dashboard.
4. To configure the server settings, click **Edit** against the Database Server and make the required changes. For example:
  - Server name; the displayed name for the server.
  - Description; a description of the server.
  - CONFIGURE DATABASE BACKEND; the minimum number of characters a word must consist of to be indexed, and whether to search on parts of a word. However, wildcard searching can make searches on large sites very slow.
  - AUTOCOMPLETE SETTINGS; how suggestions are computed when autocompletion is used.
5. To configure the index settings, click **Edit** against the Default content index and make the required changes. For example:
  - Index name; the displayed name for the index.
  - Data sources; the data sources of the items to be stored in the index.
  - CONFIGURE THE DATASOURCE; for each data source, set which bundles of content should be indexed, and in which languages.
  - Server; the server that the index should use.
  - Description; a description for the index.
  - INDEX OPTIONS; whether the index is read only, whether to index new or updated items immediately, and setting how many items to index when indexing items during a cron run.
6. When your changes are complete, click **Save**.  
The View page of the updated default database server or default content index is displayed.

## Results

---

You successfully configured the search indexes and servers in the Developer Portal.

## What to do next

---

You can configure how searches are displayed by editing the Search content view. For more information, see [Configuring the search results view in the Developer Portal](#).

## Related information

---

- [Search API module](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring the date and time

---

You can configure the date and time in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To configure the date and time, complete the following steps:

1. If the administrator dashboard is not displayed, click **Manage** to display it.
2. Click **Configuration** in the administrator dashboard.
3. Under the **Regional** and language heading, click **Date and time formats**.
4. Click **Edit** for the format you want to change.
5. Change the settings of the format then click **Save format**.  
You have configured the date and time.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the settings of a Consumer organization

You can configure the settings of your Consumer organization from within your Developer Portal.

### About this task

---

The default settings allow users to delete their accounts and organizations, create more consumer organizations, rename their organization, and change the owner of their organization.

### Procedure

---

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Configuration > System > IBM API Connect.
4. Look in the "CONSUMER ORGANIZATIONS" section.

**CONSUMER ORGANIZATIONS**

Allow users to delete their accounts  
If checked then users will be allowed to delete their accounts.

Allow users to create additional consumer organizations  
If checked then users will be allowed to create additional consumer organizations. Note that self service onboarding must also be enabled in API Manager catalog settings.

Allow users to rename their organization  
If checked then consumer organization Owner and Administrators will be able to rename their consumer organizations.

Allow users to change the owner of their organization  
If checked then consumer organization Owner and Administrators will be able to change the owner of their consumer organizations.

Allow users to delete their organizations  
If checked then Owner or Administrators users will be allowed to delete their consumer organizations.

5. Clear or set the options as needed, and then click Save configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the site default timezone

You can configure the site default timezone in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

To configure which languages are available, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the REGIONAL AND LANGUAGE heading, click Regional settings.
3. From the Default Time Zone drop-down menu, select the default time zone.
4. Click Save configuration.  
You have configured the default time zone.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the site error handling

You can configure the site error handling in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To configure the site error handling, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the DEVELOPMENT heading, click Logging and errors.
3. You can review and configure all information relating to site error handling from this view.  
Note: On a production system, set Logging and errors to **none**, to avoid any security issues.
4. Click Save configuration.  
You have configured the site error handling.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring which buttons are displayed in the WYSIWYG rich text editor

You can configure which buttons are displayed in the WYSIWYG rich text editor in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To configure which buttons are displayed in the WYSIWYG rich text editor, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration, Content authoring, Text formats and editors.
3. Click Configure for the Full HTML profile.
4. In the **TOOLBAR CONFIGURATION** section, edit the buttons that you want to expose.
5. Click Save configuration.  
You have configured the editor buttons that are displayed in the WYSIWYG rich text editor.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring which languages are available

You can configure which languages are available in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To configure which languages are available, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the REGIONAL AND LANGUAGE heading, click Languages.
3. Under the DEFAULT heading, select the radio button for the language you want to set as default.
4. To add a new language, click +Add language. Use the drop-down menu to select the language you want to add.
5. Click Save configuration.  
You have configured the languages that are available.
6. You can also provide your own translations, for strings of the website that are not translated, from the User interface translation view, under Configuration, Regional and Language in the administrator dashboard.

## Results

---

You have configured the available languages for the Developer Portal.

Note: Multilingual API and Product documentation can be created by using an `x-ibm-languages` extension directly in the OpenAPI definition. For more information, see [Using x-ibm-languages to create multilingual API and Product documentation](#).

Note: IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Customizing user account settings

You can customize the user account options that are displayed to users when they register and use the Developer Portal.

### Before you begin

You must have administrator access to complete this task.

### About this task

You can change what fields users of the Developer Portal see when they register to use the site, when they log in, and when they view their user account profile. For instance, you can change the field order display, hide fields, and add custom fields.

As an example, the Consumer organization field is a required entry field when a user creates an account on the Developer Portal. However, you can configure this field to be hidden, in which case the field is automatically completed with a default entry of `firstname lastname` of the new user instead.

### Procedure

To customize the user account settings, complete the following steps.

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Configuration > People > Account settings in the administrator dashboard.  
The Account settings page is displayed.
4. From the Settings tab you can edit the following options:
  - a. CONTACT SETTINGS; enable or disable the personal contact form for new users.
  - b. PASSWORD RESET TIMEOUT; set the timeout in seconds for one-time login links. This setting applies only to the admin user of the Developer Portal.
  - c. ANONYMOUS USERS; set the name to be used to indicate anonymous users.
  - d. ADMININISTRATOR ROLE; assign the administrator role that is automatically given new permissions whenever a new module is enabled. The default is Superuser.  
Note: The Superuser role has the highest level of permissions in the Developer Portal. This role bypasses all content access control, which means that users that have a Superuser role can see all of the site content. For more information, see [Working with roles in the Developer Portal](#).
  - e. LANGUAGE SETTINGS; enable or disable translation.
  - f. REGISTRATION AND CANCELLATION; enable or disable the password strength indicator, and set the action to take when a user account is canceled.
  - g. Notification email address; set the email address to use as the from address for all account notifications.
  - h. Emails; edit the emails that are sent when an account is blocked or canceled, or when a new password is requested.
  - i. Click Save configuration to save your updates.
5. Click the Manage fields tab to customize the fields that are available for storing user data.  
Here you can edit the settings for current fields, delete fields, and add new fields. Remember to save any updates that you make.
6. Click the Manage form display tab to configure how the user forms are displayed.
  - a. Click the Default form mode to configure how the form fields display when a user profile is being edited.
  - b. Click the Register form mode to configure how the form fields display when a new user creates an account.  
For each user form mode you can edit the field settings, enable and disable fields, and drag the fields to change their display order. You can also manage the form modes.  
For example, to hide the Consumer organization field when a new user creates an account, click the Register form mode, and drag the Consumer organization into the Disabled section.
  - c. Click Save to save your updates.
7. Click the Manage display tab to configure how fields are displayed on a user profile page.  
You can edit the field settings, enable and disable fields, and drag the fields to change their display order. You can also manage the display view modes. Remember to save any updates that you make.
8. Click the Translate account settings tab to configure the language options for the user account settings. Remember to save any updates that you make.

### Results

You successfully customized the user account settings in the Developer Portal.

### Related tasks

- [Configuring CAPTCHA](#)

Note: IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Disabling languages

You can extend your Developer Portal site by managing which languages are available for use on the site.

## Before you begin

You must have administrator access to complete this task.

## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.

To disable all languages apart from English.

2. Navigate to Configuration > Regional and language > Languages > Detection and selection.
3. Select the checkbox for the **Selected language** detection method, and deselect all other methods, then click Save settings.

### Interface text language detection

Order of language detection methods for interface text. If a translation of interface text is available in the detected language, it will be displayed.

| DETECTION METHOD               | DESCRIPTION                                    | ENABLED                             | OPERATIONS                |
|--------------------------------|--|-------------------------------------|---------------------------|
| + Account administration pages | Account administration pages language setting. | <input type="checkbox"/>            |                           |
| + URL                          | Language from the URL (Path prefix or domain). | <input type="checkbox"/>            | <a href="#">Configure</a> |
| + Session                      | Language from a request/session parameter.     | <input type="checkbox"/>            | <a href="#">Configure</a> |
| + User                         | Follow the user's language preference.         | <input type="checkbox"/>            |                           |
| + Browser                      | Language from the browser's language settings. | <input type="checkbox"/>            | <a href="#">Configure</a> |
| + Selected language            | Language based on a selected language.         | <input checked="" type="checkbox"/> | <a href="#">Configure</a> |

### Content language detection

Order of language detection methods for content. If a version of content is available in the detected language, it will be displayed.

Customize Content language detection to differ from Interface text language detection settings

[Save settings](#)

To disable one or more languages.

4. In the administrator dashboard, click Extend.  
The List tab for the Extend page opens, and the list of installed modules is displayed. The list shows all the modules that are installed. The enabled modules are displayed with a selected checkbox. The disabled modules are displayed without a selected checkbox.
5. Enter `disable language` into the search filter.  
The Disable Language content item is displayed in the list of content.
6. Select the checkbox for the **Disable Language** module, and click Enable.
7. Navigate to Configuration > Regional and language > Languages > Detection and selection.
8. Select the checkbox to enable URL, and click Save settings.

### Interface text language detection

Order of language detection methods for interface text. If a translation of interface text is available in the detected language, it will be displayed.

| DETECTION METHOD               | DESCRIPTION                                    | ENABLED                             | OPERATIONS                |
|--------------------------------|--|-------------------------------------|---------------------------|
| + Account administration pages | Account administration pages language setting. | <input type="checkbox"/>            |                           |
| + URL                          | Language from the URL (Path prefix or domain). | <input checked="" type="checkbox"/> | <a href="#">Configure</a> |

9. Navigate to Configuration > Regional and language > Languages.
10. Click Edit on the language that you want to disable.
11. Select the checkbox to **Disable language**, then click Save language.



**Edit language** ☆

Edit

Home » Administration » Configuration » Regional and language » Languages

**Language code**  
de

**Language name** \*

**Direction** \*

Left to right

Right to left

Direction that text in this language is presented.

**Disable language**  
This will remove the language from the language switcher and filter out the Simple XML sitemap

**Select language to which we redirect**

This option will redirect to the selected language when a user calls the disabled language

**Save language**

## Results

You successfully disabled your chosen language.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Disabling test tool restrictions

The default OAuth provider contains security mechanisms that prevent the ability to obtain authorization tokens in the Developer Portal test tool when you use Implicit or Authorization Code grant types. If your OAuth provider allows those actions to function correctly from the test tool, you can disable this option.

## Before you begin

You must have administrator access to complete this task.

## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. In the administrator dashboard, click Configuration » System » IBM API Connect.
3. In the **CONFIGURATION** section, deselect the **Optimise OAuth experience in test tool** option

Optimise OAuth experience in test tool

If checked then certain OAuth flows (such as implicit or access code) which cannot be completed from the test tool for technical reasons are optimised to improve usability.

4. Click Save configuration.

## Results

You successfully disabled the **Optimise OAuth experience in test tool** option. You can now use the Developer Portal test tool to test an Implicit or Authorization Code grant type in an OAuth provider API, if your OAuth provider allows those actions to function correctly from the test tool.

## What to do next

You can enable the **Optimise OAuth experience in test tool** option by repeating the task but selecting the option and then clicking Save configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Enabling a cookie compliance banner

You can add a cookie compliance banner to your Developer Portal site by configuring the EU Cookie Compliance module.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

The EU Cookie Compliance module provides a customizable pop-up notification that informs visitors to your site that cookies are being used. The notification enables your site visitors to find out more about cookies, and to decide not to browse the site if they disagree with using cookies. You can also provide a link from your cookie compliance banner to an internal or external privacy policy. For more information about configuring an internal privacy policy, see [Customizing the privacy policy statement](#).

The EU Cookie Compliance module is enabled by default. The following instructions show you how to configure the notification.

## Procedure

---

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. Click Configuration > System > EU Cookie Compliance.  
The Settings tab of the EU Cookie Compliance page is displayed.
4. In the SETTINGS section select Enable banner, otherwise the notification does not display.
5. In the PRIVACY POLICY section, edit the Privacy policy link field to provide a link to your privacy policy or other page that explains cookies to your site visitors. The default link is `/privacy`, which links to the Privacy Policy content that is configured within the Developer Portal. For more information, see [Customizing the privacy policy statement](#).

External links should start with `http://` or `https://`. For example:

```
https://example.com/privacy
```

6. Configure the remaining options as required. For example, you can change the wording, and edit the display size and color.
7. Click Save configuration to save your updates.

## Results

---

You successfully enabled a cookie compliance banner on your Developer Portal site. New visitors to your site are now prompted to accept cookies.

## Related tasks

---

- [Customizing the terms of use statement](#)
- [Customizing the privacy policy statement](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Enabling code languages for code snippets

You can specify the languages that code snippets for APIs can be displayed in.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

1. On the administrator dashboard, click Configuration > System > IBM API Connect.
2. In the API Code Snippets section, select the check boxes for the languages that you want to make available for the code snippets to be displayed in.
3. Click Save configuration.  
The languages that you have enabled are displayed as options for your code snippets.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Enabling code snippets for SOAP APIs

By default, code snippets are only shown for REST APIs. You can enable code snippets for SOAP APIs.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

Code snippets for SOAP APIs use raw HTTP, and do not use any SOAP libraries.

## Procedure

---

1. On the administrator dashboard, click Configuration, System, IBM API Connect.
2. In the API Code Snippets section, select the Display code snippets for SOAP APIs as well as REST APIs check box.
3. Click Save configuration.

The languages that you have enabled are displayed as options for your code snippets.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Forcing new users to accept terms and conditions

You can extend your Developer Portal site by forcing new users to accept the terms and conditions before they are able to create accounts.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. In the administrator dashboard, click Extend.  
The List tab for the Extend page opens, and the list of installed modules is displayed. The list shows all the modules that are installed. The enabled modules are displayed with a selected check box. The disabled modules are displayed without a selected check box.
3. Enter `terms of use` into the search filter.  
The Terms of use content item is displayed in the list of content.
4. Select the check box for the **Terms of use** module, and click Enable.
5. Navigate to Configuration, People, Terms of use.
6. Enter the name of the page that contains your site terms and conditions. If you type `terms`, the system automatically suggests the default **Terms of use (1)** page that is created by default.
7. Select the page that you require, and click Save.

## Results

---

You successfully enabled the **Terms of use** module. Now, when a user tries to create an account on your Developer Portal site, they must agree to the **Terms of use** as a part of the registration.

The image shows a registration form with the following elements:

- Consumer organization \***: A text input field containing "AppDev".
- Password \***: A password input field with a strength indicator below it. The indicator shows a yellow bar (Fair) and a grey bar (Weak).
- Password strength: Fair**: Text indicating the password strength.
- Confirm password \***: A text input field for confirming the password.
- Message box**: A green-bordered box containing the text "Your password meets the password policies required for this site".
- Terms of Use**: A section with a title and a paragraph of text. Below the text is a checkbox labeled "I agree with these terms \*".

## What to do next

You can edit the terms and conditions page. For more information, see [Customizing the terms of use statement](#). The terms are displayed when you create an account and also from the link in the site footer.

You can disable the use of terms and conditions during site registration by disabling the module.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Hiding the admin registry on the login form

You can hide the admin registry as an option on the login form. The use case for this option might be a public facing Developer Portal where admin access shouldn't be shown to the customer.

## Before you begin

You must have administrator access to complete this task.

## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. On the administrator dashboard, click Configuration > System > IBM API Connect.
3. In the Configuration section, select the check box for **Hide the admin registry on the login form**.
4. Click Save configuration.  
The **Hide the admin registry on the login form**. configuration is saved.

## What to do next

The admin registry is no longer shown on the login form. However, it is still accessible directly. The user can navigate to `<site_url>/user/login?registry_url=/admin`.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Hiding the certificate in the header for APIs secured with mutual TLS

You can hide the certificate in the header for APIs secured with mutual TLS. By default, the `x-client-certificate` header is shown in an API when mutual TLS is configured. You can choose to turn off this option in your Developer Portal.

### Before you begin

The parameters for the example API showing the certificate.

Parameters

Header

|                      |   |
|----------------------|---|
| Accept               | application/json                        |
| optional             |   |
| Content-Type         | application/json                        |
| optional             |   |
| X-Client-Certificate | The TLS certificate of your application |
| Required             | string                                  |

Example

```
-----BEGIN CERTIFICATE-----xxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxx  
xxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPL  
ExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxxxEXAMPLExxxxxxxx  
xxEXAMPLExxxx-----END CERTIFICATE-----
```

You must have administrator access to complete this task.

### Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. On the administrator dashboard, click Configuration > System > IBM API Connect.
3. In the Configuration section, deselect the checkbox for **Show certificate in header for APIs secured with mutual TLS**.
4. Click Save configuration.  
The **Show certificate in header for APIs secured with mutual TLS** configuration is saved.  
The parameters for the example API now not showing the certificate:

Parameters

Header

|          |                  |
|----------|------------------|
| Accept   | Permitted values |
| optional |                  |
|          | application/json |

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Importing and exporting taxonomies

You can import and export taxonomies in the Developer Portal.

## Before you begin

You must have administrator access to complete this task.

## Procedure

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click **Manage** to display it.
3. In the administrator dashboard, click **Configuration** » **Content authoring** » **Term CSV Export/Import**.
4. Select **Term CSV Import**, or **Term CSV Export**.

For Term CSV Import

5. Enter your CSV input in the format given in the examples, and select the **Taxonomy** from the drop-down list.

The screenshot shows the 'CSV Term Import' configuration page. At the top, there are two tabs: 'Term CSV Import' (selected) and 'Term CSV Export'. Below the tabs is a breadcrumb trail: 'Home » Administration » Configuration » Content authoring'. The main section is titled 'Input' and contains a large empty text area for entering CSV data. Below the text area, there is a note: 'See CSV Export for an example.' followed by the instruction 'Enter in the form of:'. Two examples of CSV headers are provided: one with fields 'name, description, format, weight, parent\_name, [any\_additional\_fields]' and another with 'tid, uuid, name, description, format, weight, parent\_name[:parent\_name1;parent\_name2;...], [any\_additional\_fields]'. A note states that '[any\_additional\_fields]' are optional and stringified using 'http\_build\_query'. There is a checkbox labeled 'Preserve Vocabularies on existing terms.' which is unchecked. Below this is a 'Taxonomy' dropdown menu with 'forums' selected. At the bottom of the form is a 'Next' button.

6. Click **Next**.
7. Click **Import**.

For Term CSV Export

8. Select the **Taxonomy** from the drop-down list, and check any other options that you want.

The screenshot shows the 'CSV Term Export' configuration page. At the top, there are two tabs: 'Term CSV Import' and 'Term CSV Export' (selected). Below the tabs is a breadcrumb trail: 'Home » Administration » Configuration » Content authoring'. The main section is titled 'Taxonomy' and contains a dropdown menu with 'forums' selected. Below the dropdown are three checkboxes: 'Include Term Ids in export.', 'Include Term Headers in export.', and 'Include extra fields in export.', all of which are unchecked. A note states that fields are stringified using 'http\_build\_query'. At the bottom of the form is an 'Export' button.

9. Click **Export**.

## Results

You successfully imported or exported a taxonomy.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Restricting access by IP address

You can restrict access to a role or a user by IP address in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

By restricting access to a role by IP address, that role is unavailable to users outside of the IP address ranges that you define. If a role restriction is triggered, the user's session is unaffected, but the restricted role is no longer available to the user. Role restriction affects only the availability of the restricted role to users. Role restrictions are available for all roles, except `anonymous user` and `authenticated user`.

By restricting login access of a user by IP address, the user is unable to log in outside of the IP address ranges that you define. You can also specify global IP address ranges, which apply to all users. IP restrictions are checked on every page load. If a user restriction is triggered by an attempt at logging in being denied, then the user is logged out and sent to the 'error page' that is specified by the site administrator.

Note: IP address ranges must be entered in CIDR notation that is separated with semi-colons and no trailing semi-colon. For more information on CIDR notation, see [CIDR format](#).

### Procedure

---

1. In the Developer Portal, click Configuration, > People, > Restrict by IP, then click **Restrict log in by IP**.
2. On the **Global restrictions** tab, enter the address of the page to which the user is redirected to if they are not allowed to log in, in the Login denied error page.
3. Select one of the following options:
  - To restrict access to a role by IP address, complete the following steps:
    - Click Restrict role by IP.
    - Decide which role you want to restrict, the roles that you can restrict include Administrator, Content Author, and Forum Moderator.
    - Enter the IP address range (in CIDR notation) that you do not want to restrict log in for in the field for that role, for example, Forum Moderator role IP range.
    - Click Save configuration.

## Restrict role by IP ☆

General Settings

Restrict login by IP

Restrict role by IP

Home » Administration » Configuration » People » Restrict By IP

✓ The configuration options have been saved.

### Forum Moderator role IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon.

For more information on CIDR notation click [here](#).

Leave field blank to disable IP restrictions for Forum Moderator.

### Content Author role IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon.

For more information on CIDR notation click [here](#).

Leave field blank to disable IP restrictions for Content Author.

### Administrator role IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon.

For more information on CIDR notation click [here](#).

Leave field blank to disable IP restrictions for Administrator.

### Superuser role IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon.

For more information on CIDR notation click [here](#).

Leave field blank to disable IP restrictions for Superuser.

Save configuration

- To restrict login access of a user by IP address, complete the following steps:
  - Click Restrict login by IP, then click **User restrictions**.
  - In the **ADD NEW USER ALLOWED IP RANGE** section, enter a user name in the **Username** field.
  - Enter an IP address Range (in CIDR notation), that you do not want to restrict log in for in the **Allowed IP range** field.
  - Click Save configuration. Your new log in restriction can be seen after the **ADD NEW USER ALLOWED IP RANGE** section.



## Restrict login by IP ☆

General Settings

Restrict login by IP

Restrict role by IP

Global Restrictions

User restrictions

Home » Administration » Configuration » People » Restrict By IP » Restrict login by IP

✓ The configuration options have been saved.

### ADD NEW USER ALLOWED IP RANGE

Username

Allowed IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon. E.G. 10.20.30.0/24;192.168.199.1/32;1.0.0.0/8  
For more information on CIDR notation click [here](#).

### apicdemo1 user IP range

Enter IP Address Ranges in CIDR Notation separated with semi-colons, with no trailing semi-colon. E.G. 10.20.30.0/24;192.168.199.1/32;1.0.0.0/8

For more information on CIDR notation click [here](#).

Leave field blank to disable IP restrictions for apicdemo1.

Save configuration

- To restrict login for all users, complete the following steps:
  - Click Restrict login by IP, then click **Global restrictions** tab.
  - Enter the global IP address ranges (in CIDR notation) in the Restrict global login to allowed IP range field.
  - Click Save configuration.
- 4. To remove an IP restriction, delete the value that is associated with the restriction, then click Save configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Restricting or preventing access from search engines

You can restrict or prevent access to your Developer Portal from search engines. You might want to control the access if you have a development or test site.

### Before you begin

You must have administrator access to complete this task.

### About this task

You might have a development or test Developer Portal site on the internet, but do not want them to be indexed by Google, or other search engines, making them visible to your customers. Google explicitly advises not to use `robots.txt` as a blocking mechanism. A correct solution is to control access by using Metatag.

### Procedure

1. In the Developer Portal, click Configuration, Search and metadata, Metatag.
2. On the **Global restrictions** tab, click Edit.
3. In the Advanced section, select the meta tags that you want to use to restrict or prevent the access:

## Robots

- index – Allow search engines to index this page (assumed).
- follow – Allow search engines to follow links on this page (assumed).
- noindex – Prevents search engines from indexing this page.
- nofollow – Prevents search engines from following links on this page.
- noarchive – Prevents cached copies of this page from appearing in search results.
- nosnippet – Prevents descriptions from appearing in search results, and prevents page caching.
- noodp – Blocks the [Open Directory Project](#) description from appearing in search results.
- noydir – Prevents Yahoo! from listing this page in the [Yahoo! Directory](#).
- noimageindex – Prevent search engines from indexing images on this page.
- notranslate – Prevent search engines from offering to translate this page in search results.

Provides search engines with specific directions for what to do when this page is indexed.

4. Click Save.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## toggling the site in and out of maintenance mode

You can put the Developer Portal site into maintenance mode for short periods of time. During maintenance mode, only the administrator can access the site; all other users who enter the site URL get a maintenance message set by the administrator.

### Before you begin

You must have administrator access to complete this task.

**Important:** Maintenance mode is designed for short-term site maintenance; it is not meant for long-term usage. While a site is in maintenance mode, the database is not updated with new content from API Manager. As soon as the maintenance tasks are finished, you must take the site out of maintenance mode. Do not log out of the Developer Portal while the site is in maintenance mode, as you will not be able to log back in to the UI. In this case, maintenance mode can be turned off only by using the CLI; see the following instructions for details.

### Procedure

To put the site into maintenance mode, complete the following steps in the Developer Portal UI:

1. Log in to your site as the admin user.
2. Click Configuration > Development > Maintenance mode in the administrator dashboard.
3. Select the Put site into maintenance mode checkbox.
4. Enter a site message, or leave the default site maintenance text.  
This message is what will be seen by users who go to the site while it is in maintenance mode.
5. Click Save configuration.  
Your site is now in maintenance mode.

After you finish your maintenance tasks, take the site out of maintenance mode by completing the following steps:

6. Click Configuration > Development > Maintenance mode in the administrator dashboard.
7. Clear the Put site into maintenance mode checkbox.  
Your site is now available to users and out of maintenance mode.

If your site is in maintenance mode and you are locked out of the UI, you must use the command line interface (CLI) to turn off maintenance mode.

8. Use the CLI to log in to your Developer Portal appliance, and move to the directory of the site that you want to update.
9. Run the following command:

```
drush sset system.maintenance_mode 0 --input -format=integer
```

Maintenance mode is now turned off.

Note: If you are using IBM® API Connect V2018 Reserved Instance, and you are locked out of the Developer Portal UI, you must raise a support request on the IBM Cloud support page to get your site updated. For more information, see [Getting help from IBM support](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Viewing available updates

You can view available updates in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

To view available updates, complete the following steps:

1. Click Reports in the administrator dashboard.
2. Click Available updates.
3. You can see what updates are available from this view. However, updates can be applied only by applying the latest available API Connect release; see [Upgrading in a Kubernetes environment](#), or [Upgrading in a VMware environment](#), for more information.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing Developer Portal security

You can manage multiple security elements in the Developer Portal.

- [Configuring CAPTCHA](#)  
You can add a CAPTCHA challenge to any of your Developer Portal pages to protect your site from spam.
- [Configuring reCAPTCHA](#)  
You can configure reCAPTCHA on your Developer Portal site to provide more protection from spam and abuse.
- [Configuring session timeout and limit](#)  
You can specify session timeout settings to control when a user is automatically logged out of the Developer Portal. You can also restrict the number of sessions a single user has available to them.
- [Configuring the timeout length for the password reset link](#)  
You can configure the timeout length for the one-time password reset link that is sent from the Developer Portal to the admin user account.
- [Configuring your site password policy](#)  
You can configure your site password policy to define the constraints that are applied to the passwords of your Developer Portal users.
- [Disabling CORS warnings](#)  
You can disable cross-origin resource sharing (CORS) warnings for unenforced APIs in the Developer Portal.
- [Disabling live testing of APIs](#)  
You can disable live testing of APIs in the Developer Portal to restrict exposure of an API.
- [How to enable penetration testing, and information about Developer Portal cookies](#)  
Many security features are available in the Developer Portal, but if you need to do a penetration test of your Developer Portal site, some of these features might block the scan that is required as part of the testing. This topic provides guidance on the security features that you might need to disable to allow penetration testing of your site, as well as information about Developer Portal cookies.
- [How to manage IP security in the Developer Portal](#)  
The Developer Portal offers the ability to perform various IP address security measures, such as adding and removing specific IP addresses from the banned IP address list, automatically banning client IP addresses by using the Drupal Perimeter Defence module, or managing login security by using flood control.
- [Managing banned IP addresses](#)  
You can ban specific IP addresses from accessing your Developer Portal site.
- [Using flood control for login security](#)  
You can configure login security for your Developer Portal by using flood control.
- [Login security](#)  
You can configure the login security for your Developer Portal at an IP and user level. You can also configure login security for Developer Portal contact forms.
- [Using the Security kit](#)  
You can improve the security of your website by configuring various options that are available in the Security Kit module, in the Developer Portal.
- [Using Honeypot for spam protection](#)  
Honeypot protection provides security mechanisms to protect your Developer Portal site from form submission by spam bots. If spam bot activity is detected, form submission is blocked.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring CAPTCHA

You can add a CAPTCHA challenge to any of your Developer Portal pages to protect your site from spam.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

The types of CAPTCHA challenges that are available include Image and Math based CAPTCHAs. Both types of CAPTCHA challenges are configurable in the CAPTCHA dialog box. By default, your Developer Portal site is configured with the Image based CAPTCHA challenge enabled. You can also configure reCAPTCHA on your site. For more information, see [Configuring reCAPTCHA](#).

## Procedure

---

To configure CAPTCHAS, complete the following steps:

1. If the administrator dashboard is not displayed, click [Manage](#) to display it.
2. Click [Configuration](#) in the administrator dashboard.
3. In the PEOPLE section, click CAPTCHA module settings.  
The CAPTCHA settings window opens.
4. Use the CAPTCHA configuration options provided as required. For example, you can complete the following tasks:
  - a. Specify the CAPTCHA challenge type.
  - b. Specify which forms must include a CAPTCHA challenge.
  - c. Add a description to the CAPTCHA.
  - d. Enable CAPTCHA statistics.
5. Click [Save configuration](#) to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring reCAPTCHA

You can configure reCAPTCHA on your Developer Portal site to provide more protection from spam and abuse.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

reCAPTCHA is a CAPTCHA-like system that is designed to establish that a computer user is human. The types of reCAPTCHA that are available include Checkbox, Invisible, and Android.

Note: To use reCAPTCHA the administrator of your Developer Portal is required to sign up to Google's reCAPTCHA API, and your portal server must have internet access.

## Procedure

---

To configure reCAPTCHA, complete the following steps:

1. If the administrator dashboard is not displayed, click [Manage](#) to display it.
2. Click [Configuration](#), [People](#), [CAPTCHA module settings](#), [reCAPTCHA](#) in the administrator dashboard.  
The reCAPTCHA settings window opens.
3. In the GENERAL SETTINGS section, click [register for reCAPTCHA](#).  
The Google registration site for reCAPTCHA is displayed.
4. Follow the instructions on Google to register your Developer Portal site, and obtain the Site key and Secret key.
5. Return to the reCAPTCHA settings window in the Developer Portal, and enter the Site key and Secret key that you obtained from Google into the GENERAL SETTINGS section.
6. Optional: Update the WIDGET SETTINGS section as required.
7. Click [Save configuration](#) to save your changes.
8. Click the CAPTCHA Settings tab, and in the FORM PROTECTION section change the Default challenge type to reCAPTCHA (from module reCAPTCHA).
9. Click [Save configuration](#).

## Results

---

The reCAPTCHA module is now enabled on your Developer Portal site.

## Related tasks

---

- [Configuring CAPTCHA](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring session timeout and limit

You can specify session timeout settings to control when a user is automatically logged out of the Developer Portal. You can also restrict the number of sessions a single user has available to them.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. To manage the automated logout settings, complete the following steps:
  - a. Click Configuration in the administrator dashboard.
  - b. In the PEOPLE section, click Automated logout settings.

Note: You must check **Enforce auto logout on admin pages** for **autoLogout** to apply when navigating the admin areas of the site.

Enforce auto logout on admin pages

If checked, then users will be automatically logged out when administering the site.

**Save configuration**

- c. Use the session timeout configuration options provided as required. For example, you can complete the following tasks:
    - Specify the length of inactivity time, in seconds, before a user is automatically logged out.
    - Specify session timeout values for individual user roles.  
Note: The default session timeout is 30 minutes.
    - Provide a redirection URL at logout.
    - Provide session expiry message texts.
  - d. Click Save configuration to save your changes.
3. To restrict the number of sessions available to a user, complete the following steps:
    - a. Click Configuration in the administrator dashboard.
    - b. In the PEOPLE section, click Session limit.
    - c. Specify the default maximum number of sessions for a user by entering the number into the Default maximum number of active sessions field.
    - d. Select the required action from the When the session limit is exceeded options.
    - e. From the Logged out message severity list, select the severity of the message that the user receives when they are logged out.
    - f. Optionally, in the ROLE LIMITS section, specify separate session limits for each role.
    - g. Click Save configuration to save your changes.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Configuring the timeout length for the password reset link

You can configure the timeout length for the one-time password reset link that is sent from the Developer Portal to the admin user account.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

By completing the steps in this task, you are configuring the timeout length for the one-time password reset link that is sent to the admin user account by the Developer Portal.

Note:

- Setting the password reset timeout length in the Developer Portal affects only the admin user account.
- You are not affecting the user activation or password reset links that are used by any other identification provider type in API Manager.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.

3. In the PEOPLE section, click Account settings.  
The Account settings window opens.
4. In the Account settings window, navigate to the PASSWORD RESET TIMEOUT section.
5. Enter your required value for the timeout in the Password Reset Timeout field.  
Note: The timeout value is specified in seconds, and is 86400 seconds by default.
6. Click Save configuration.  
You have configured the timeout length for the one-time link that is sent from the Developer Portal for the admin user account.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring your site password policy

You can configure your site password policy to define the constraints that are applied to the passwords of your Developer Portal users.

---

### Before you begin

You must have administrator access to complete this task.

---

### About this task

Any passwords that are stored in the API Manager must comply with the API Manager password policy. You can configure your password policy to be the same as or stricter than the API Manager password policy.

---

### Procedure

To configure your site password policy, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. In the SECURITY section, click Password Policy.  
The Password Policies page opens.
4. In the OPERATIONS column for the default password policy, click Edit.  
Note: You can create a new password policy by clicking Add Policy; this allows you to apply different password policy settings to different roles. If you need only a single password policy, just configure the pre-supplied default policy.
5. Click Next.  
The Configure Constraints page opens.
6. Configure the password constraints as required.
  - a. To configure a constraint, click Edit in the OPERATIONS column for the required constraint.
  - b. To delete a constraint, select Delete in the OPERATIONS column.
  - c. To add a constraint, select the required constraint type from the drop-down list, then click Configure Constraint Settings.
  - d. When you are done, click Next.  
The Apply to Roles page opens.
7. Select the roles to which the password policy settings are to be applied. then click Finish.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Disabling CORS warnings

You can disable cross-origin resource sharing (CORS) warnings for unenforced APIs in the Developer Portal.

---

### Before you begin

You must have administrator access to complete this task.

---

### About this task

By disabling the CORS warnings, users will not receive further warnings regarding the usage of unenforced APIs.

---

### Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. In the SYSTEM section, click IBM API Connect.
4. Ensure that the Display CORS warnings for unenforced APIs check box is cleared.

5. Click Save Configuration.

You have disabled CORS warnings in the Developer Portal. To enable CORS warnings, repeat the steps and ensure that the Display CORS warnings for unenforced APIs check box is selected.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Disabling live testing of APIs

You can disable live testing of APIs in the Developer Portal to restrict exposure of an API.

### Before you begin

You must have administrator access to complete this task.

### Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. In the SYSTEM section, click IBM API Connect.
4. Ensure that the Allow live testing of APIs check box is cleared.
5. Click Save configuration.

You have disabled live testing of APIs. To enable live testing of APIs, repeat the steps and ensure that the Allow live testing of APIs check box is selected.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## How to enable penetration testing, and information about Developer Portal cookies

Many security features are available in the Developer Portal, but if you need to do a penetration test of your Developer Portal site, some of these features might block the scan that is required as part of the testing. This topic provides guidance on the security features that you might need to disable to allow penetration testing of your site, as well as information about Developer Portal cookies.

Penetration testing, or pen testing, is the practice of testing a computer system, network, or web application for potential security vulnerabilities that an attacker might exploit. For certain elements of your pen testing, you might want to disable some of the security features that block scanning of the Developer Portal. The following sections describe the security features in the Developer Portal that you might need to disable. The actual requirements depend on your Developer Portal configuration, and on the type of pen testing that you are running. Note also the final section, which discusses testing for Distributed Denial of Service attacks.

Important:

- If you run a pen test, and you give the automated test tool the admin credentials, the tool submits all of the administration configuration forms as well, and this action is likely to break the Portal site configuration. Best practice is to use different credentials for the testing.
- If you disable security modules, and then run security scans, your Portal site configuration might be broken by the scanning process. Before running any security testing, ensure that you have a valid backup of your Portal site.
- If your Portal site configuration is broken by the security testing, then you must delete the Portal site from the API Manager Catalog settings and re-create it, or restore it from a backup (for information about restoring your site, see [How to restore a Developer Portal service](#)).
- Note that disabling of security modules makes it highly likely that you will find issues that are not actually vulnerabilities, because you have disabled the modules that are designed to protect against these vulnerabilities. Therefore, any issue that is found in these circumstances must be replicated in an environment without the security modules disabled, before contacting IBM Support.
- Do not use the **Enforced** mode of the **Content Security Policy**, this mode is not supported and can break core Developer Portal functionality such as API rendering and the content editor. The **Content Security Policy** must be in **Report only** mode.

Note: You might need to disable the **Honeypot** module to allow your penetration test to run. For more information, see [Using Honeypot for spam protection](#). This topic contains the following sections:

- [CAPTCHA and reCAPTCHA verifications](#)
- [Drupal Perimeter Defense module](#)
- [Login Security module](#)
- [Developer Portal cookies](#)
- [Should I test the Developer Portal for vulnerability to Distributed Denial of Service attacks?](#)

---

### CAPTCHA and reCAPTCHA verifications

CAPTCHA and reCAPTCHA verifications are a type of challenge-response test that are used in web applications to check whether the user is human. By default, your Developer Portal site is configured with an Image based CAPTURE challenge enabled, but you can also configure a Math based CAPTCHA challenge, and Checkbox, Invisible, and Android reCAPTCHA challenges. If you need to disable these verifications before pen testing, see the following instructions.

To disable the CAPTCHA and reCAPTCHA verifications, you must disable the modules as follows:



1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Extend > Disable module.
4. Search through the list of modules, and select the check boxes next to the Image CAPTCHA and reCAPTCHA modules.
5. Click Disable, and click Disable again to confirm.
6. On the Disable tab, select the check box next to the CAPTCHA module, and click Disable. The display shows the forms that the CAPTCHA configuration is removed from.
7. Click Disable again to confirm. Note that the modules must be disabled in this order.

The CAPTCHA and reCAPTCHA modules are now disabled, and you can complete your pen testing. After pen testing, you can enable the modules again by clicking Extend on the administrator dashboard, selecting the check box for each module in the List tab, and clicking Enable.

## Drupal Perimeter Defense module

---

The Drupal Perimeter Defense module (machine name: perimeter) immediately bans the IP addresses of those users that send suspicious requests to your site, so any pen testing scans would be blocked. This module is enabled by default on the Developer Portal. For more information about the module, see the Drupal documentation [Drupal Perimeter Defense module](#).

If you need to disable the Drupal Perimeter Defense module before pen testing, see the following instructions:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Extend > Disable module.
4. Search through the list of modules, and select the check box next to the Drupal Perimeter Defense module.
5. Click Disable, and click Disable again to confirm.

The Drupal Perimeter Defense module is now disabled, and you can complete your pen testing. After pen testing, you can enable the module again by clicking Extend on the administrator dashboard, selecting the check box next to the Drupal Perimeter Defense module in the List tab, and clicking Enable.

## Login Security module

---

The Login Security module (machine name: login\_security) enables the site administrator to improve the security options in the login operation of your Developer Portal site. For example, controlling the number of invalid login attempts before blocking users, and denying access by IP address. This module is enabled by default on the Developer Portal. For more information about the module, see the Drupal documentation [Login Security module](#).

If you need to disable these security options before pen testing, you need to manually modify the configuration of your security options according to your requirements. For example, you might need to change the number of login attempts that are allowed before a user or IP address is blocked. To configure the security options, see the following instructions:

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. On the administrator dashboard, click Configuration > People > Login Security.
4. In the Login Security window, update the security options according to your requirements.
5. Click Save configuration to save your updates.

The Login Security options are now updated, and you can complete your pen testing. After pen testing, you can reconfigure your security options in the Login Security window.

## Developer Portal cookies

---

Cookies are small text files that are placed on a computer by websites when they are visited. The Developer Portal uses cookies that remain on the computer for varying times. Some cookies expire at the end of each session, and some remain longer to enhance the user experience in the future. The list here explains the cookies that are used by the Developer Portal, and why they are used.

Note: Any load balancer that sits in front of the Developer Portal must be a transparent proxy that enables all cookies returned by the Developer Portal to pass through to the browser.

### Drupal session identifier cookie

This is the main cookie that all of the Developer Portal sites have, and it stores the session identification. The session cookie is named in the format `SSESSUUID`, where `UUID` is a randomly generated unique string of characters for that session, and this is the session token itself. The session cookie is always marked with the following flags as instructions to the browser client:

- **Secure** - means that the cookie is sent only over a secure connection (that is, over TLS).
- **HttpOnly** - means that the browser adds this cookie on outbound requests only, and not provide the value to client side javascript.

### session\_store.id cookie

This cookie is used to store information on where a user is in the subscription flow, and any options that they selected, if they haven't yet signed in, so the Developer Portal can redirect them back there later. The `session_store.id` cookie is also marked with the **Secure** and **HttpOnly** flags, and it does not contain any confidential information.

### Drupal.visitor.startSubscriptionWizard cookie

This cookie is used to store information on where a user started in the subscription flow if they haven't yet signed in, so the Developer Portal can redirect them back there later. The `Drupal.visitor.startSubscriptionWizard` cookie is also marked with the **Secure** and **HttpOnly** flags, and it does not contain any confidential information.

### portal\_session\_route cookie

This cookie is used to achieve session persistence, which it does by ensuring that web traffic between the user and the Developer Portal is correctly routed to the server. The `portal_session_route` cookie is used for both HTTP and HTTPS communications, so that routing is meaningful irrespective of the transfer protocol, and ensuring consistent behavior if the switching of protocols is required. The cookie contains only a hash of the pod that served the traffic to maintain session persistence, and does not contain any confidential information. The cookie is provided by the Portal ingress. This cookie is not marked as **Secure** or **HttpOnly**, as the Kubernetes ingress does not yet support these attributes.

### Other cookies



In addition to the cookies listed previously, there are other (usually temporary) cookies that might be created depending on the configuration options of your Developer Portal site. For example, if the site uses OAuth for authentication, then during the OAuth token activation flow a JSON Web Token is sent to the browser. This token is stored in a cookie for a limited lifetime. This action is part of the normal function of an OAuth flow. As the creation of these other cookies depends on your particular Developer Portal configuration, we cannot provide a complete list of cookies here.

## Should I test the Developer Portal for vulnerability to Distributed Denial of Service attacks?

When you run penetration testing, it's important to understand that individual systems are not responsible for (nor capable of dealing with) Distributed Denial of Service (DDoS) attacks. DDoS attacks are launched by huge numbers of external computers simultaneously, with a goal of using up all of the resources of a system, and thereby denying service to legitimate users. These kinds of attacks are usually defended against at the level of the network and edge gateways, and information about how to protect against them is out of the scope of this topic. Although the Developer Portal does have some capabilities, including the modules mentioned previously, for potentially mitigating some of the effects of a DDoS attack, do not rely on these capabilities as the first line of defense; rather they are designed to assist in dealing only with any elements of an attack that might have gotten through the outer network layers of protection.

## Related tasks

- [Configuring CAPTCHA](#)
- [Configuring reCAPTCHA](#)
- [Disabling modules](#)
- [Login security](#)
- [Blocking and unblocking specific users](#)
- [Managing banned IP addresses](#)

## Related information

- [Understanding Drupal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## How to manage IP security in the Developer Portal

The Developer Portal offers the ability to perform various IP address security measures, such as adding and removing specific IP addresses from the banned IP address list, automatically banning client IP addresses by using the Drupal Perimeter Defence module, or managing login security by using flood control.

See the following links for information about how to manage IP security in the Developer Portal:

- [Enable and disable IP security](#)
- [Correctly passing through client IP addresses](#)
- [Managing banned IP addresses](#)

## Enable and disable IP security

The ability to enable or disable IP security related actions can be controlled at the Portal service level by using an internal Developer Portal command. When IP security is enabled, modules such as the Drupal Perimeter Defence module, or flood control, will block client IP addresses suspected of malicious behavior, as expected. When IP security is disabled, all IP related security is switched off. For example, the perimeter module won't block client IP addresses when IP security is disabled. You might want to turn off IP security if you are performing penetration tests, or if you cannot pass through the client IP address from your external load balancer. Note that IP security is enabled by default.

Complete the following instructions to toggle the IP security setting between enabled and disabled.

Note: These instructions assume you have a working Kubernetes environment with the kubectl command-line tool installed, and that you understand how to manage Kubernetes. For more information, see <https://kubernetes.io>.

1. Run the following command to get a list of the pods that are running on the virtual machine where the Developer Portal service is deployed:

```
kubectl get pods -n namespace
```

Identify the `portal-www` pod from the resulting list.

2. Run the following command to set the Developer Portal IP security to enabled:

```
kubectl exec portal-www-pod -c admin -- /opt/ibm/bin/set_ip_security_enabled -e
```

For example:

```
kubectl exec r1234b98d02-apic-portal-www-0 -c admin -- /opt/ibm/bin/set_ip_security_enabled -e
Toggling IP security = 1 on the Developer Portal
Running: drush8 -y @test.com state-set ibm_apim.ip_ban_enabled --input-format=integer 1
IP security successfully set to 1
```

3. Run the following command to set the Developer Portal IP security to disabled:

```
kubectl exec portal-www-pod -c admin -- /opt/ibm/bin/set_ip_security_enabled -d
```

For example:

```
kubectl exec r1234b98d02-apic-portal-www-0 -c admin -- /opt/ibm/bin/set_ip_security_enabled -d
Toggling IP security = 0 on the Developer Portal
Running: drush8 -y @test.com state-set ibm_apim.ip_ban_enabled --input-format=integer 0
IP security successfully set to 0
```

---

## Correctly passing through client IP addresses

---

In order to correctly use modules that make use of client IP banning, such as the Drupal Perimeter Defence module, you must ensure that any external load balancer that fronts the Portal cluster passes through the client IP address. This can be achieved by passing the client IP address through in an '**x-forwarded-for**' header, or by making use of the proxy protocol (provided both the load balancer and the ingress controller are compatible with the proxy protocol, and have the protocol enabled). Failure to correctly pass through the client IP address, results in the load balancer IP address being blocked when a client attempts to send a suspicious request to the portal.

If you are fronting the Portal cluster with an HAProxy setup acting as the load balancer, then you can make use of the proxy protocol by adding the **send-proxy** directive to the end of the Portal server declarations in the HAProxy configuration file, for example:

```
server portal0 portal_host:port check send-proxy
```

You must restart the HAProxy for the change to take effect. Note that if your load balancer IP address has already been blocked, you will need to remove the blocked IP address from the banned list. For more information, see [Managing banned IP addresses](#).

For more information about the Drupal Perimeter Defence module, see [Drupal Perimeter Defense module](#).

---

## Managing banned IP addresses

---

You can manage banned IP addresses for a particular site by using the administrator dashboard for a particular Developer Portal site. For more information, see [Managing banned IP addresses](#).

---

## Related concepts

---

- [How to enable penetration testing, and information about Developer Portal cookies](#)

---

## Related tasks

---

- [Disabling modules](#)
- [Login security](#)
- [Blocking and unblocking specific users](#)
- [Managing banned IP addresses](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Managing banned IP addresses

---

You can ban specific IP addresses from accessing your Developer Portal site.

---

## Before you begin

---

You must have administrator access to complete this task.

---

## Procedure

---

To manage banned IP addresses, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. In the PEOPLE section, click IP address bans.  
The IP address bans window opens.
4. You can complete the following tasks:
  - a. To ban an IP address, enter the address in the IP address field, then click Add.  
The address is added to the BANNED IP ADDRESSES list.
  - b. To remove an IP address from the banned list, click Delete alongside the required address.

Note: There is automatic login "flood" protection built into the Developer Portal. This feature means that an excess of login attempts from the same IP address in one hour will cause that IP to be temporarily banned. For more information, see [Using flood control for login security](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using flood control for login security

You can configure login security for your Developer Portal by using flood control.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Navigate to Configuration > System > Flood Control.
3. Enter values in the following fields in the LOGIN SETTINGS section according to your requirements:
  - Failed IP login limit (min 1) - the number of failed login attempts from the same client IP.
  - Failed IP login window in seconds (0 = Off) - the time window, in seconds, for the number of failed login attempts from the same client IP.
  - Failed User login limit (min 1) - the number of failed login attempts from a user.
  - Failed User login window in seconds (0 = Off) - the time window, in seconds, for the number of failed login attempts from a user.
4. Enter values in the following fields in the CONTACT FORM section according to your requirements:
  - Emails sent limit - the number of emails that can be sent.
  - Emails sent window in seconds - the time window, in seconds, for the number of emails that can be sent.
5. Click Save configuration.

### What to do next

---

To unblock a user, see [Blocking and unblocking specific users](#). To unblock an IP address, see [Managing banned IP addresses](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Login security

You can configure the login security for your Developer Portal at an IP and user level. You can also configure login security for Developer Portal contact forms.

### Before you begin

---

Important: The Login Security module is being removed from future releases. It has been replaced by Flood Control. Existing users of the Login security module should configure Flood Control with the same limits and time windows as specified here. For more information, see [Using flood control for login security](#).

You must have administrator access to complete this task.

### Procedure

---

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. In the PEOPLE section, click Login Security.
4. Enter values in the following fields in the GENERAL SETTINGS section according to your requirements:
  - Track time - the time window to check for security violations.
  - User - the number of login failures before blocking a user.
  - Soft host - the number of login failures before soft blocking a host. The host can still browse the site as an anonymous user.
  - Host - the number of login failures before hard blocking a host. The host cannot access the site at all.
  - Attack detection - the number of login failures before warning about an ongoing attack.
5. Select the check boxes that satisfy your notification requirements in the NOTIFICATION section:
  - Disable login failure error message
  - Notify the user about the number of remaining login attempts
  - Display last login timestamp
  - Display last access timestamp
6. Optional: Edit the EMAIL FOR ONGOING ATTACK DETECTION section.
7. Optional: Edit the EMAIL FOR BLOCKED ACCOUNT section.
8. Optional: You can edit the notification texts that are displayed by configuring the following options:
  - Failed login attempt
  - Banned host (soft IP ban)
  - Banned host (hard IP ban)
  - Blocked user by uid
9. After you have configured your login security options, click Save configuration.

### What to do next

---

To unblock a user, see [Blocking and unblocking specific users](#). To unblock an IP address, see [Managing banned IP addresses](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using the Security kit

You can improve the security of your website by configuring various options that are available in the Security Kit module, in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

The Security Kit module provides your website with various options to mitigate risks of common web application vulnerabilities like Cross-site Scripting, Cross-site Request Forgery and Clickjacking. It also has some options to improve your SSL/TLS security.

### Procedure

---

1. If the administrator dashboard is not displayed, click [Manage](#) to display it.
2. Click [Configuration](#) in the administrator dashboard.
3. In the **SYSTEM** section, click [Security Kit settings](#).
4. In the Security Kit module, configure the security settings for your website as required.
5. When you are done, click [Save configuration](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Using Honeypot for spam protection

Honeypot protection provides security mechanisms to protect your Developer Portal site from form submission by spam bots. If spam bot activity is detected, form submission is blocked.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

Honeypot protection provides the following security mechanisms:

- A hidden field, unseen by users, is added to the form. If a value has been entered in the field when the form is submitted then this indicates that the form was completed by a spam bot, and the submission is blocked. You can specify the name of the hidden field.
- If the form is submitted before a specified time has elapsed (five seconds by default), it is assumed that this is too short a time for a human to have completed the form, and the submission is blocked. You can specify the time length.

Honeypot protection is provided by the Honeypot module, which is enabled by default.

**Note:** If you want to carry out automated testing of your Developer Portal, you might need to disable the Honeypot module, because Honeypot spam protection is designed specifically to block automated Developer Portal usage. For details on how to disable a module, see [Disabling modules](#).

### Procedure

---

To configure Honeypot for spam protection in the Developer Portal, complete the following steps:

1. If the administrator dashboard is not displayed, click [Manage](#) to display it.
2. Click [Configuration](#) in the administrator dashboard.
3. In the **CONTENT AUTHORING** section, click [Honeypot configuration](#).
4. Specify the forms that you want to protect with Honeypot.
  - a. To enable Honeypot protection for all the forms on your Developer Portal site, select [Protect all forms with Honeypot](#).
  - b. To choose which forms you want to protect with Honeypot, clear the [Protect all forms with Honeypot](#) check box, then select the required forms in the **HONEYPOT ENABLED FORMS** section.

By default, Honeypot protection is enabled for all user management forms and all comment forms.
5. To have details of all blocked form submissions written to the log file, select [Log blocked form submissions](#).
6. In the Honeypot element name field, specify the name of the hidden form field.

The default field name is `url`. You need to change the value if your form already has a field of the same name. For the most effective protection, use a generic field name; for example, `email`, `homepage`, or `link`.
7. In the Honeypot time limit field, specify the number of seconds that must elapse before it is assumed that a form is being submitted by a human rather than a spam bot. If the form is submitted before this time has elapsed then the submission is blocked.

The default value is five seconds.  
8. When done, click Save configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## People

You can manage and customize the user experience in the Developer Portal, by altering permissions and assigning roles to specific users.

To learn how to configure the user experience for your Developer Portal site, use the following section links.

- [Working with roles in the Developer Portal](#)  
As a site administrator, you can create and customize site configuration roles in the Developer Portal.
- [Blocking and unblocking specific users](#)  
You can block and unblock specific users in the Developer Portal.
- [Controlling access to Developer Portal content](#)  
In IBM® API Connect, you can restrict access to content for Consumer organizations through an access content list. However, you can also assign permissions in the Developer Portal that override the access content list, and allow users to access and edit all of the content for Products, APIs, and Applications.
- [Creating a developer account to customize API properties](#)  
You can allow specific users in the Developer Portal to customize the properties for certain APIs, including APIs that are not viewable by the administrator, or other customizable features, by creating an internal Consumer organization.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Working with roles in the Developer Portal

As a site administrator, you can create and customize site configuration roles in the Developer Portal.

Rather than assigning individual permissions to each user, permissions are assigned to roles, and then roles are assigned to users. Using roles gives administrators greater control over permissions, and makes it easy to assign and remove roles when necessary.

Administrators can assign the following site configuration roles to users in the Developer Portal:

### Superuser

The Superuser role has the highest level of permissions in the Developer Portal, and is the role that is assigned by default to the admin user, in addition to the Administrator role, when they enable a Developer Portal site in the API Manager. This role bypasses all content access control, which means that users that have a Superuser role can see all of the site content. This level of access can lead to some unexpected behavior when they access content such as their Consumer organizations and their applications. Careful consideration should be given before you assign users to a Superuser role.

### Administrator

An Administrator can use the administrator dashboard to customize and configure the Developer Portal. An Administrator is able to perform any task within the Developer Portal that does not involve the creation of APIs, Products, and Apps.

### Content Author

A Content Author is a curator of content for the Developer Portal. A Content Author can perform the following actions:

- Set taxonomies on the content.
- Provide custom icons for content.
- Upload additional documentation files.
- Create documentation pages in the Developer Portal.

### Forum Moderator

A Forum Moderator can moderate the content of a forum in the Developer Portal.

**Note:** Every account in the Developer Portal must have a unique email address, including the site Admin account. The default email address for the Admin account is the email address of the Catalog owner. It is not possible to create a user account (and associated Consumer organization) with the same email address as the Admin account (or that of the Catalog owner if their email address is different). Any attempts to create an account with the same email address results in the new account not functioning correctly, and returning the following error message when trying to log in: **A user already exists with this email address.**

To learn how to create and customize roles in the Developer Portal, use the following topic links:

- [Creating administrator users for the Developer Portal](#)  
You can create additional administrator users for an Developer Portal.
- [Assigning users to a role](#)  
Assign users to one or more roles in the Developer Portal to enable role specific permissions.
- [Assigning permissions to a role](#)  
You can assign permissions to a new role in the Developer Portal, and update the permissions for current roles.
- [Creating a new role](#)  
You can create a new role within the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Creating administrator users for the Developer Portal

You can create additional administrator users for an Developer Portal.

### Before you begin

You must have administrator access to complete this task.

### About this task

After you enable a Developer Portal site in the API Manager, you are sent an email that contains a one-time log in link for the Developer Portal site Admin user. The Admin user can administer the Content Management System (CMS) capabilities of the Developer Portal. For more information about enabling a site, see [Creating and configuring Catalogs](#).

However, you can also create additional administrative roles for users from within the Developer Portal by assigning the role to users that have access to the Developer Portal. A user with an Administrator role can use the administrator dashboard to customize and configure the Developer Portal. An Administrator is able to perform any task within the Developer Portal that does not involve the creation of APIs, Products, and Apps. You should be careful to ensure that only trusted users are given this access and level of control of your site.

### Procedure

To create additional administrative users, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click People in the administrator dashboard.
3. Select the List tab.
4. Select the check boxes for the target users.
5. From the drop-down list under the Action heading, select Add the Administrator role to the selected user(s).
6. Click Apply to selected items.

### Results

You have assigned the selected users to the Administrator role.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Assigning users to a role

Assign users to one or more roles in the Developer Portal to enable role specific permissions.

### Before you begin

You must have administrator access to complete this task.

### Procedure

To assign users to a role, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click People in the administrator dashboard.
3. Select the List tab.
4. Select the check boxes for the target users.
5. From the drop-down list under the Action heading, select the role that you want to assign.
6. Click Apply to selected items.

### Results

You have assigned the selected users to the required role.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Assigning permissions to a role

You can assign permissions to a new role in the Developer Portal, and update the permissions for current roles.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

To assign or update permissions for a role, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click People in the administrator dashboard.
3. Select the Permissions tab.
4. Assign a permission to a role, by selecting the check box in the relevant column for the permission that you want to assign.
5. Click Save permissions.

### Results

---

You have assigned permissions to the target role.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a new role

You can create a new role within the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

### Procedure

---

To create a new role, complete the following steps:

1. Click People in the administrator dashboard.
2. Select the Roles tab.
3. Click Add role.
4. Enter a name for the new role in Role name, and click Save.
5. Your new role is displayed in the list of roles. You can drag any of the roles to re-order them. It's recommended to order roles from least permissive (for example, Anonymous user) to most permissive (for example, Administrator user). Users who aren't logged in have the Anonymous user role. Users who are logged in have the Authenticated user role, plus any other roles granted to their user account.

### Results

---

You have created a new role in the Developer Portal.

### What to do next

---

The new role is created with the same permissions as an Authenticated user role. You can click the Permissions tab and edit the permissions as required.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Blocking and unblocking specific users

You can block and unblock specific users in the Developer Portal.

### Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To block or unblock a specific user, complete the following steps:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click People in the administrator dashboard.  
A list of users is displayed.
3. Select the check box next to the target user.
4. From the drop-down list under Action, select Block the selected user(s) or Unblock the selected user(s).
5. Click Apply to selected items.

You have blocked or unblocked specific users from the Developer Portal.

Note: If a user reports that they are unable to login, but they do not appear as blocked, then it might be that they have been blocked by Flood Control. To unblock such users, select Configuration > System > Flood unblock. You can configure your Flood Control settings; see [Using flood control for login security](#) for more information.

## Related tasks

---

- [Using flood control for login security](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Controlling access to Developer Portal content

In IBM® API Connect, you can restrict access to content for Consumer organizations through an access content list. However, you can also assign permissions in the Developer Portal that override the access content list, and allow users to access and edit all of the content for Products, APIs, and Applications.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

A user with a Content Author role is a curator of content for the Developer Portal. A Content Author user can perform the following actions:

- Set taxonomies on the content
- Provide custom icons for content
- Upload additional documentation files
- Create documentation pages in the Developer Portal.

By default, Content Author users are assigned the following user permissions:

Edit any Product content

The user can access and edit all of the Products available in the Catalog that the Developer Portal is associated with. The user can also access APIs that belong to those Products.

Edit any API content

The user can access and edit all of the APIs that are available in the Catalog, but not the Products with which they are associated. (Note that this permission is a subset of Edit any Product content; as Content Author users are assigned both permissions they can access and edit all of the Products and the associated APIs that are available.)

The Edit any Application permission grants the user access to all of the Applications that are in the Developer Portal, but is not assigned to any user or role by default.

Note: There are security implications if you grant the Edit any Application permission, and it should be assigned only to trusted roles.

A user can configure, attach, and upload content if they are assigned the appropriate permissions.

## Procedure

---

To check and assign permissions that grant users access to all content, proceed with one of the following options:

- Assign a user to the Content Author role. For more information, see [Assigning users to a role](#).
- Create a custom role, to which you can add the relevant permissions and add users to your new custom role. For more information, see [Creating a new role](#), [Assigning permissions to a role](#), and [Assigning users to a role](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a developer account to customize API properties



You can allow specific users in the Developer Portal to customize the properties for certain APIs, including APIs that are not viewable by the administrator, or other customizable features, by creating an internal Consumer organization.

## Before you begin

---

You must have administrator access to complete this task.

## About this task

---

Every user in the Developer Portal must be a member of a Consumer organization. To enable additional users to be editors of particular content in the Developer Portal, you can create an internal Consumer organization. You can then invite these additional editors to this organization, and assign them a specific role that grants certain editing permissions. For example, the following API properties can be customized by any member of the new Consumer organization that have been assigned the Administrator or Content Author role:

- Image uploading
- Tag assignment
- File attachment
- Custom field editing

## Procedure

---

To create a new Consumer organization within the Developer Portal, complete the following steps:

1. Log in to the Developer Portal as an administrator or Content Author.
2. Click the arrow next to `your_organization_name` from the Developer Portal home page, and select Create organization.
3. In the Organization Title field, type the name of the organization that you want, then click Submit.

To invite the editor users to your new organization, complete the following steps:

4. Switch to your newly created organization by clicking the arrow next to `your_current_organization_name`, and select your new organization name from the list.
5. Click the arrow next to `your_new_organization_name` and select My organization.  
Your new organization page is displayed.
6. For each new user, click Invite, complete their Email details, and click Submit. Note that you can leave the default role of Developer selected and then change the role later, or you can select the Administrator role now. However, be aware that an administrator is able to perform any task within the Developer Portal that does not involve the creation of APIs, Products, and Apps. You should be careful to ensure that only trusted users are given this access and level of control of your site. Each new user will receive an invitation email, and they need to click on the link in the email to activate their Developer Portal account.

To assign the required permissions to the users of your organization, you must log in as an administrator and complete the following steps:

7. Log in to the Developer Portal as an administrator.
8. If the administrator dashboard is not displayed, click Manage to display it.
9. Click People on the administrator dashboard.
10. Select the check box for the user, or users, to whom you want to give new permissions.
11. Under the Action heading, select the relevant role from the drop-down list.  
For example, selecting Add the Content Author role to the selected user(s) means they can edit images and content within the Developer Portal.
12. Click Apply to selected items to update the roles.

## What to do next

---

If you want specific Products or APIs to be customized by the editor users, you must ensure that the relevant Products are visible to the internal Consumer organization that you have created. For more information on how to make a plan visible in API Manager, see [Changing the availability of a Product](#). Note that you can also create new roles and assign specific permissions to those roles. For more information, see [Creating a new role](#) and [Assigning permissions to a role](#).

## Related concepts

---

- [Working with roles in the Developer Portal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Forums

Create and control forums in the Developer Portal.

The Developer Portal lets you create threaded discussion boards, called forums, on your site. To learn how to configure forums, use the following topic links.

- [Creating a new forum](#)  
You can create a new forum in the Developer Portal.
- [Creating new forum containers](#)  
You can create new forum containers in the Developer Portal.
- [Configuring the creation of new forums for each API](#)  
You can choose whether or not to automatically create a new forum when an API is created in the Developer Portal.

- [Locking forum topics](#)  
You can lock forum topics in the Developer Portal.
- [Marking content in a forum as sticky](#)  
You can mark content in a forum as sticky in the Developer Portal.
- [Removing forum posts](#)  
You can remove forum posts in the Developer Portal.
- [Turning off forums in the Developer Portal](#)  
You can turn off forums in the Developer Portal by disabling the Forum module. This action prevents users from viewing, and contributing to, forums.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating a new forum

You can create a new forum in the Developer Portal.

### Before you begin

---

You must have administrator or Forum moderator access to complete this task.

### Procedure

---

To create a new forum, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Forums.
4. Click Add forum.
5. Fill in the Name and Description fields for the new forum.
6. From the Parent drop-down list, select the hierarchical position of the new forum.
7. Click Save. You have created a new forum.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Creating new forum containers

You can create new forum containers in the Developer Portal.

### Before you begin

---

You must have administrator or Forum moderator access to complete this task.

### Procedure

---

To create a new forum containers, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Structure in the administrator dashboard.
3. Click Forums.
4. Click Add container.
5. Fill in the Name and Description fields for the new forum container.
6. From the Parent drop-down list, select the hierarchical position of the new forum container.
7. Click Save. You have created a new forum container.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Configuring the creation of new forums for each API

You can choose whether or not to automatically create a new forum when an API is created in the Developer Portal.

## Before you begin

---

You must have administrator access to complete this task.

## Procedure

---

To configure the creation of new forums for each API, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Configuration in the administrator dashboard.
3. Under the SYSTEM heading, click IBM API Connect.
4. Select or clear the Automatically create a forum per API check box to turn the feature on or off.
5. Click Save configuration.

You have configured the creation of new forums for each API.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Locking forum topics

You can lock forum topics in the Developer Portal.

## Before you begin

---

You must have administrator or Forum moderator access to complete this task.

## Procedure

---

To lock forum topics, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. Click Edit for the forum topic that you want to lock.
4. Click Comment settings. The view expands.
5. Select Closed .
6. Keeping the Published selected, click Save.

You have disallowed any further replies and marked the topic as locked.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Marking content in a forum as sticky

You can mark content in a forum as sticky in the Developer Portal.

## Before you begin

---

You must have administrator or Forum moderator access to complete this task.

## Procedure

---

To mark content in a forum as sticky, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. From the Action drop-down menu, select Make content sticky.
4. Select the check boxes next to the forum topics that you want to make sticky.
5. Click Apply to selected items.

You made the selected forum content sticky.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Removing forum posts

You can remove forum posts in the Developer Portal.

### Before you begin

---

You must have administrator or Forum moderator access to complete this task.

### Procedure

---

To remove forum posts, complete the following tasks:

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Click Content in the administrator dashboard.
3. From the Action drop-down menu, select Delete content.
4. Select the check boxes next to the forum posts that you want to remove.
5. Click Apply to selected items.

You removed the selected forum posts.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Turning off forums in the Developer Portal

You can turn off forums in the Developer Portal by disabling the Forum module. This action prevents users from viewing, and contributing to, forums.

### Before you begin

---

You must have administrator access to complete this task.

### About this task

---

To turn off forums, you must first remove all Forums terms, and then you can disable the Forum module.

### Procedure

---

To turn off forums, complete the following steps.

1. If the administrator dashboard is not displayed, click Manage to display it.
2. Delete all Forums terms by completing the following steps:
  - a. Click Structure > Taxonomy > Forums in the administrator dashboard.  
The list of Forums terms is displayed.
  - b. For each item in the list of terms, select Delete from the drop-down list in the OPERATIONS column, and then click Delete again to confirm.  
All of the Forums terms are deleted.
3. Disable the Forum module by completing the following steps:
  - a. Click Extend > Disable module in the administrator dashboard.
  - b. Select the Forum module, then click Disable.
  - c. The display then shows a list of forum configuration that will also be removed if the Forum module is disabled. Click Disable again to confirm that you agree to this removal.

### Results

---

You successfully turned off forums in the Developer Portal.  
If you want, you can enable the Forum module again by clicking Extend, selecting the Forum module on the List tab, and clicking Enable.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Reports

You can view reports about the general configuration and status of your Developer Portal site.

Use the Reports section in the Developer Portal to view information about the following areas:

#### Available updates

Status report about available updates for your installed modules and themes.

#### Node Type Count

Status report about how much of each content type there is in your portal site, and whether it is published or not. It can also provide totals of how many users there are of each role.

#### Available translation updates

Status report about available interface translations for your installed modules and themes.

#### Field list

Overview of fields on all entity types.

#### Status report

Short overview of your site's parameters, as well as any problems detected with your installation.

#### Views plugins

Overview of the plugins that are used in all views.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Extend

You can extend the Developer Portal by creating and installing custom modules, or installing additional Drupal 9 contributed modules.

Modules provide functionality to your site, and the *core modules* that are included in the standard Developer Portal installation, provide all of the basic functions that most sites need. However, some of the core modules are not enabled by default, so you might want to examine the module list under the Extend menu of your site, to see whether the functionality you require is available from a core module before starting to investigate custom and contributed modules. If you still require additional custom or contributed modules, consider the following topics to help you create and implement modules consistently.

Note: The Developer Portal contains a Rules framework for event based actions. Some of those actions and triggers might exhibit unexpected behavior. A workaround is to create a custom module instead, because the same events and actions are possible there.

- [Custom module development: an introduction to Drupal tools for PHP development](#)

Some guidance about the Drupal tools and environment considerations when developing in PHP, to help when writing custom modules to extend your Developer Portal functionality.

- [Custom module development: background and prerequisites](#)

You can create custom modules in the Developer Portal to extend functionality. The following information provides a getting started developer guide to custom module development.

- [Custom module development: creating a module skeleton](#)

You can extend your Developer Portal site by creating custom modules. The starting point for a new module is to create a module skeleton.

- [Custom module development: using hooks](#)

You can use hooks in custom modules to alter the behavior of Drupal core or other modules in your Developer Portal.

- [Installing custom modules](#)

You can extend your Developer Portal site by installing custom modules that you created, and also installing contributed modules from the Drupal community.

- [Deleting custom modules](#)

You can delete custom modules from your Developer Portal if you want to remove them entirely from your site.

- [Disabling modules](#)

You can disable an entire module in the Developer Portal if you want to improve performance, or remove functionality.

## Related information

- [Extending Drupal](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Custom module development: an introduction to Drupal tools for PHP development

Some guidance about the Drupal tools and environment considerations when developing in PHP, to help when writing custom modules to extend your Developer Portal functionality.

The following sections provide an overview of the key automation tools and environment considerations for PHP development in the Developer Portal. For a complete list of Drupal tools, see [Development tools overview](#) on Drupal.org, however, note that you are not permitted to access the Developer Portal command line interface (CLI).

Important:

- You are not permitted to include any IBM® API Connect modules within any custom modules that you create. Also, directly editing any API Connect themes, modules, included modules, or Drupal core on the file system is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.
- All custom development is your responsibility. Although the use of custom modules and themes is supported, IBM API Connect do not provide support in their development or modification.

## Devel, Devel Generate Kint, and Webprofiler development modules

---

The Devel, Devel Generate Kint, and Webprofiler modules are all packaged within the Devel module. These modules provide the following development support:

- *Devel* - Helper functions for Drupal developers.
- *Devel Generate Kint* - Accelerate development of your site or module by quickly generating nodes, comments, terms, users, and more.
- *Kint* - Pretty print of variables with `kint($my_var)`.
- *Webprofiler* - Port of the Symfony WebProfiler Bundle as a Drupal module.

For more information, see [Devel module](#) on drupal.org, and [Devel](#) on drupalship.org.

---

## Admin Toolbar module

The Admin Toolbar module improves the default Drupal Toolbar by transforming it into a drop-down menu, providing fast access to the administration pages. It also adds extra links to the Drupal icon on the left of the admin toolbar to development tools such as flush all caches, run cron, and run updates.

For more information, see [Admin toolbar module](#) on drupal.org.

---

## Examples for developers project

The Examples for Developers project aims to provide high-quality, well-documented API examples for a broad range of Drupal core functionality.

The project contains many modules that illustrate best practices for implementing Drupal core APIs. For example there are examples for Block, Cache, Config and Content Entity, Cron, Database API, Email, Events, Form API, Field, Field Permission, File, Hooks, Javascript, Node Type, Page, Pager, PHPUnit, Plugin Type, Queue, Simple Test, Stream Wrapper, Table Sort, Testing, and Tour.

For more information, see [Examples for developers project](#) on drupal.org.

---

## Coding standards

The Drupal Coding Standards apply to code within Drupal and its contributed modules. For more information, see [Coding standards](#) on drupal.org.

---

## Related tasks

- [Installing custom modules](#)

---

## Related information

- [📖 Drupal tools for developers](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Custom module development: background and prerequisites

You can create custom modules in the Developer Portal to extend functionality. The following information provides a getting started developer guide to custom module development.

Custom modules are written in PHP, which is an open source server-side scripting language that is used for creating dynamic web pages. For general information on the Drupal tools for PHP development, see [Custom module development: an introduction to Drupal tools for PHP development](#).

The following sections provide an overview of custom module development, with links to learn more about the required development languages and concepts. Creating custom modules affects the source of your Developer Portal site, so it is recommended that you gain Drupal experience and PHP knowledge before you start creating modules.

Important:

- You are not permitted to include any IBM® API Connect modules within any custom modules that you create. Also, directly editing any API Connect themes, modules, included modules, or Drupal core on the file system is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.
- All custom development is your responsibility. Although the use of custom modules and themes is supported, IBM API Connect do not provide support in their development or modification.

---

## Object-Oriented Programming (OOP)

OOP is established as best practice for development, so ensure your knowledge of OOP is good. The PHP documentation on [Classes and Objects](#) is a good place to start. For a general overview of PHP best practices, see [PHP The Right Way](#). The following resources are also helpful:

- [Object-oriented programming](#) (on Wikipedia)
- [Object Oriented Programming with PHP](#) (on phpro.org)
- [Object-Oriented PHP for Beginners](#) (on tutstplus.com)
- [Object Oriented Programming in PHP](#) (on tutorialspoint.com)

Drupal also uses some common design patterns, so you need to ensure you have a basic understanding of these. See the following pattern resources:

- [The Factory Pattern](#) (on [phptherightway.com](#)), and [Late Static Bindings](#) (on [php.net](#))
- [Software design pattern](#) (on Wikipedia)
- [Programming Foundations: Design Patterns](#) (on [lynda.com](#))

---

## PHP namespaces

You must be familiar with the concept of namespaces in PHP. In most cases, the Drupal code is namespaced based on the module that code belongs to.

For example, the namespace for `block.module` is:

```
namespace Drupal\block;
```

When you create Drupal modules, it is important to consider that PHP has a global namespace, and that function names must be unique. You are recommended to prefix the name of any methods with the module name.

For example, if you have a module that is named `custom_module`, a `create` method within it is called `custom_module_create()`.

For more information about namespaces, see the following resources:

- [Drupal namespace standards](#) (on [drupal.org](#))
- [How to use PHP namespaces](#) (on [sitepoint.com](#))
- [PHP Namespaces](#) (on [php.net](#))

---

## Dependency injection

It is important that you have a baseline understanding of dependency injection. Drupal makes heavy use of this concept, so you will need this understanding to be able to access and make use of many of the core APIs.

To find out more about dependency injection, see [Dependency Injection](#) (on [phptherightway.com](#)), as well as the additional articles that are linked to on that page. See also [Services and dependency injection in Drupal](#) (on [drupal.org](#)).

---

## Symfony

Symfony is a set of reusable PHP components and a PHP framework for web projects. Drupal borrows from this framework in order to reduce code duplication across various PHP projects. Much of the code that Drupal uses to handle routing, sessions, and the services container, amongst other things, is borrowed from Symfony.

To understand how Symfony works, see the [Symfony Documentation](#) on [symfony.com](#).

---

## Other useful resources

- <https://api.drupal.org/api/drupal/core%21core.api.php/group/annotation/9.1.x> - list of the different annotation types that Drupal uses for plugin discovery, and to provide additional context/meta-data for the code that's being executed.
- [Plugin API overview](#) - overview about how plugins are used in Drupal.

---

## Related tasks

- [Installing custom modules](#)

---

## Related information

- [🔗 Creating custom modules](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Custom module development: creating a module skeleton

You can extend your Developer Portal site by creating custom modules. The starting point for a new module is to create a module skeleton.

---

## Before you begin

You must have administrator access to complete this task.

You must also have experience in Drupal and PHP development. For more information, see [Custom module development: an introduction to Drupal tools for PHP development](#) and [Custom module development: background and prerequisites](#).

Important:

- You are not permitted to include any IBM® API Connect modules within any custom modules that you create. Also, directly editing any API Connect themes, modules, included modules, or Drupal core on the file system is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.
- All custom development is your responsibility. Although the use of custom modules and themes is supported, IBM API Connect do not provide support in their development or modification.

## About this task

Every custom module creation starts with creating an `.info.yml` file. This file, or module skeleton, is used to store the metadata about the custom theme. After the module skeleton is created, the module can be installed onto your site by using the Extend menu on the administrator dashboard of the Developer Portal UI. However, without any module specific code, your site won't be affected.

The following instructions guide you through creating a custom module skeleton. What to do next in your custom module creation depends on what you want your module to do. Options are shown at the end of this task.

## Procedure

1. Choose a short name, also known as a machine name, for your custom module.

The machine name is used in several files and functions in your module, as well as by Drupal core to programmatically refer to your module. The name must conform to the following rules:

- It must start with a letter.
- It must contain only lower-case letters and underscores. (Note that if upper-case letters are used in the name, Drupal won't recognize any hook implementations in your module.)
- It must not contain any spaces.
- It must be unique within your Developer Portal site.
- It should not be any of the following reserved terms: `src`, `lib`, `vendor`, `assets`, `css`, `files`, `images`, `js`, `misc`, `templates`, `includes`, `fixtures`, `Drupal`.

2. Create a module folder in a location of your choice, for example, `/modules/module_name`.

You don't have to name the folder the same name as your module machine name, however, you must remember to use the machine name programmatically within your module's code and filenames.

3. Create a `module_name.info.yml` file in your module folder.

Where `module_name` is the short name, or machine name, that you chose in Step 1.

4. Insert the metadata for your custom module into the `module_name.info.yml` file as follows:

```
name: Example module name
description: Description of the module.
package: Custom

type: module
core: 8.x
core_version_requirement: '^8 || ^9'
version: 1.0

dependencies:
  - module_name

test_dependencies:
  - module_name

configure: example.settings

php: 7.1

hidden: true
```

where

- **name** (required) is the name of your module, and is the text that is shown on the module administration page in the Developer Portal UI.
- **description** (optional) is a description of your module, and is the text that is shown on the module administration page in the Developer Portal UI.
- **package** (optional) is a key that enables you to group similar modules together.
- **type** (required) is the type of extension that you are creating, for example, a module, theme or profile. In this instance you are creating a module.
- **core** (optional) specifies with which Drupal core version your module is compatible.
- **core\_version\_requirement** (required) specifies which versions of Drupal your module is compatible with. This example specifies that the module is compatible with all versions of Drupal 8 and 9.
- **version** (required) is the version number of your module. Note that **version** is not required if you are publishing your module on Drupal.org.
- **dependencies** (optional) is a YAML list of any modules that your module depends on. They must be namespaced in the format `module_name`, where `module_name` is the module's machine name.
- **test\_dependencies** (optional) is list of any modules (in the same format as **dependencies**) that are needed to run certain automated tests for your module on Drupal's automated test runner (DrupalCI), but are not needed as module dependencies in general.
- **configure** (optional), if your module offers a configuration form, specify the route to this form here.
- **php** (optional) is the minimum PHP version that is required for your module. For the Developer Portal the minimum PHP version is 7.1.
- **hidden** (optional) is an option that, if set, will hide your module from the Extend section of the administrator dashboard on the Developer Portal UI. This option might be useful if your module only contains tests, or if the module is meant as an example to other developers.

5. Save your `module_name.info.yml`.

## Results

You successfully created your module skeleton.

## Example

An example `.info.yml` file:

```
name: User field example
type: module
description: Developer Portal tutorial about a user field example
package: IBM API Connect Tutorials
core: 8.x
core_version_requirement: '^8 || ^9'
version: 8.x-0.0.1
dependencies:
```



```
- ibm_apim
- auth_apic
```

## What to do next

---

You now need to create the appropriate content for your custom module, depending on what you want the module to do. For information about the various options, see [Creating custom modules](#) on Drupal.org. There is also a walkthrough example of creating a basic custom module that is available on Drupal.org, see: [A "Hello World" custom page module](#).

Note: If you are defining URL paths in the routing file of your module, you must consider any potential implications of those paths, and avoid conflicts with other parts of the Developer Portal. For example, you are recommended not to prefix any paths with `ibm_apic`.

After you have created your custom module content, you need to package your custom module files up into a zip, tar, tgz, gz, or bz2 file. This file can then be installed onto your Developer Portal site by using the Extend menu on the administrator dashboard. For more information, see [Installing custom modules](#).

## Related information

---

- 🔗 [Let Drupal know about your module with an .info.yml file](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Custom module development: using hooks

You can use hooks in custom modules to alter the behavior of Drupal core or other modules in your Developer Portal.

Custom modules enable you to extend the functionality of your Developer Portal site. For more information about how to create custom modules, see [Custom module development: an introduction to Drupal tools for PHP development](#), and [Custom module development: background and prerequisites](#).

Hooks are one of the ways that custom modules can use to interact with other modules and with Drupal core subsystems. The following sections provide an overview of hooks, and a list of the IBM® API Connect specific hooks.

Important:

- You are not permitted to include any IBM API Connect modules within any custom modules that you create. Also, directly editing any API Connect themes, modules, included modules, or Drupal core on the file system is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.
- All custom development is your responsibility. Although the use of custom modules and themes is supported, IBM API Connect do not provide support in their development or modification.

## About hooks

---

Hooks define functions that alter the behavior of Drupal core. So one way for custom modules to alter these functions, is to use hooks. Hooks are specially-named functions that a module defines (this is known as *implementing the hook*), these hooks are then discovered and called at specific times to alter or add to the base behavior or data (this is known as *invoking the hook*). Each hook has a name (for example: `hook_batch_alter()`), a defined set of parameters, and a defined return value. Your custom modules can implement hooks that are defined by Drupal core, by API Connect, or by other modules that they interact with. Your custom modules can also define their own hooks, in order to let other modules interact with them. For more information, see [Understanding Hooks](#) on Drupal.org.

Note: If you write a hook implementation function, the function must not throw errors or exceptions, or use `alert` level log messages. Errors, exceptions, and `alert` level log messages can break all the processing in the parent code. For example, if you write a hook that calls an external server when an application is updated, the code needs to have error handling in place to handle the server not being found, or not returning the expected result. Otherwise, Drupal stops parsing the function and the application might not get updated correctly. It can cause webhook and snapshot parsing to abort leaving the portal in an inconsistent state.

For a list of all the hooks that are available on Drupal core, and how to implement them in your custom module, see [Drupal API Hooks](#).

For a list of all the API Connect specific hooks, see the following section.

## Using hooks in custom modules

---

To invoke a hook you need to add a function to your `.module` file in your custom module, and prefix the hook with your custom module name. For example, when using an API Connect hook in a custom module, you must replace the word `hook` in the hook name with the name of your custom module. For example:

`hook_apic_app_create`

should be referenced in your custom module as:

`moduleName_apic_app_create`

See also the following example `.module` file that shows a custom module called `user_field_example`:

```
/**
 * Implementation of hook_form_alter() to alter the Sign up form
 */
function user_field_example_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if ($form_id === 'user_register_form') {
    // add our validator to the #validate array for the user_register_form
    $form['#validate'][] = 'user_field_example_validate_department_code';
  }
}
```

```

/**
 * Validate the Department code field on the Sign up form.
 *
 * Valid entry = DEPnnn
 * where n = single figure digit.
 */
function user_field_example_validate_department_code($form, &$form_state) {
  if (isset($form_state->getValue('field_department_code')[0]['value'])) {
    $dept_code = $form_state->getValue('field_department_code')[0]['value'];
    $valid = preg_match('/^DEP\d{3}$/i', $dept_code);
    if (!$valid) {
      // if the value is not valid then set an inline error on the relevant field
      $form_state->setErrorByName('field_department_code', t('Invalid department code.'));
    }
  }
}

```

## API Connect specific hooks

The following tables list the hooks that are specific to the API Connect Developer Portal. Each section contains a link to the .php file on the API Connect Developer Portal repository on GitHub for those hooks. This file contains examples of how you can use the hooks in your modules.

Note: From IBM API Connect Version 2018.4.1.10, hooks `hook_apic_app_delete` and `hook_consumerorg_delete` are deprecated, and are replaced by the following hooks:

- `hook_apic_app_pre_delete`
- `hook_apic_app_post_delete`
- `hook_consumerorg_pre_delete`
- `hook_consumerorg_post_delete`

### Hooks about Applications

The following table lists the hooks relating to Applications in the Developer Portal. For examples of how to use Application hooks, see [apic\\_app/apic\\_app.api.php](#) on GitHub.

Table 1. Hooks about Applications

| Hook name   | Description   |
|---|---|
| <code>hook_apic_app_create</code>                           | Triggered when an Application is created.   |
| <code>hook_apic_app_update</code>                           | Triggered when an Application is updated.   |
| <code>hook_apic_app_delete</code>                           | Triggered when an Application is deleted. Note: This hook is deprecated from IBM API Connect Version 2018.4.1.10.   |
| <code>hook_apic_app_pre_delete</code>                       | Triggered when an Application is deleted, before the node deletion or cascade has happened.   |
| <code>hook_apic_app_post_delete</code>                      | Triggered when an Application is deleted, after the node deletion or cascade has happened.  |
| <code>hook_apic_app_promote</code>                          | Triggered when an Application is promoted.  |
| <code>hook_apic_app_creds_create</code>                     | Triggered when a new set of credentials is created for an Application.  |
| <code>hook_apic_app_creds_update</code>                     | Triggered when a set of credentials is updated for an Application.  |
| <code>hook_apic_app_creds_delete</code>                     | Triggered when a set of credentials is deleted for an Application.  |
| <code>hook_apic_app_subscribe</code>                        | Triggered when a subscription is created.   |
| <code>hook_apic_app_migrate</code>                          | Triggered when a subscription is migrated to a new Plan.  |
| <code>hook_apic_app_unsubscribe</code>                      | Triggered when an Application is unsubscribed from a Plan.  |
| <code>hook_apic_app_image_create</code>                     | Triggered when a custom Application image is created.   |
| <code>hook_apic_app_image_delete</code>                     | Triggered when a custom Application image is deleted.   |
| <code>hook_apic_app_clientid_reset</code>                   | Triggered when a credential client ID is reset.   |
| <code>hook_apic_app_clientsecret_reset</code>               | Triggered when a credential client secret is reset.   |
| <code>hook_apic_app_modify_getplaceholderimage_alter</code> | Alter the Application placeholder image provided in <code>\Drupal\apic_app\Application::getPlaceholderImage()</code> . Can be used to define a specific placeholder image to use when the consumer has not uploaded their own custom image for their Application. |
| <code>hook_apic_app_modify_getimageforapp_alter</code>      | Alter the Application placeholder image provided in <code>\Drupal\apic_app\Application::getImageForApp()</code> . Can be used to provide a full path to a specific image to use for an Application that overrides any custom image that might have been uploaded. |
| <code>hook_apic_app_modify_clientid_reset_alter</code>      | Alter the client ID provided by API Manager when the ID is reset.   |
| <code>hook_apic_app_modify_clientsecret_reset_alter</code>  | Alter the client secret provided by API Manager when the secret is reset.   |
| <code>hook_apic_app_modify_credentials_create_alter</code>  | Alter the credentials provided by API Manager when a new Application is created.  |
| <code>hook_apic_app_modify_credentials_create_alter</code>  | Alter the credentials provided by API Manager when new credentials are created.   |

### Hooks about Consumer organizations

The following table lists the hooks relating to Consumer organizations in the Developer Portal. For examples of how to use Consumer organization hooks, see [consumerorg/consumerorg.api.php](#) on GitHub.

Table 2. Hooks about Consumer organizations

| Hook name                            | Description  |
|--------------------------------------|--|
| <code>hook_consumerorg_create</code> | Triggered when a Consumer organization is created.   |
| <code>hook_consumerorg_update</code> | Triggered when a Consumer organization is updated.   |
| <code>hook_consumerorg_delete</code> | Triggered when a Consumer organization is deleted. Note: This hook is deprecated from IBM API Connect Version 2018.4.1.10. |

| Hook name                    | Description  |
|------------------------------|--|
| hook_consumerorg_pre_delete  | Triggered when a Consumer organization is deleted, before the node deletion or cascade has happened. |
| hook_consumerorg_post_delete | Triggered when a Consumer organization is deleted, after the node deletion or cascade has happened.  |

Hooks about APIs

The following table lists the hooks relating to APIs in the Developer Portal. For examples of how to use API hooks, see [apic\\_api/apic\\_api.api.php](#) on GitHub.

Table 3. Hooks about APIs

| Hook name            | Description                       |
|----------------------|-----------------------------------|
| hook_apic_api_create | Triggered when an API is created. |
| hook_apic_api_update | Triggered when an API is updated. |
| hook_apic_api_delete | Triggered when an API is deleted. |

Hooks about Products

The following table lists the hooks relating to Products in the Developer Portal. For examples of how to use Product hooks, see [product/product.api.php](#) on GitHub.

Table 4. Hooks about Products

| Hook name           | Description                          |
|---------------------|--------------------------------------|
| hook_product_create | Triggered when a Product is created. |
| hook_product_update | Triggered when a Product is updated. |
| hook_product_delete | Triggered when a Product is deleted. |

## Related tasks

- [Installing custom modules](#)

## Related information

- [Creating custom modules](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

# Installing custom modules

You can extend your Developer Portal site by installing custom modules that you created, and also installing contributed modules from the Drupal community.

## Before you begin

You must have administrator access to complete this task.

## About this task

You can add extra functions to your site by installing new custom or contributed modules. For more information on creating custom modules, see [Custom module development: background and prerequisites](#). Contributed modules are modules that are contributed by Drupal community members. To search for contributed modules, go to [Drupal Download & Extend](#). You can install both types of modules into your Developer Portal site by using the following instructions.

## Procedure

1. If the administrator dashboard is not displayed, click Manage to display it.
2. In the administrator dashboard, click Extend.  
The List tab for the Extend page opens, and the list of installed modules is displayed. The list shows all the modules that are installed. The modules that are displayed with a selected checkbox are enabled, the modules that are displayed without a selected checkbox are not enabled.
3. Click + Install new module, and either complete the Install from a URL field, or click Choose file to upload a *filename.tar.gz* module file from your local computer.
4. Click Install.  
The Update manager confirms that the module was installed successfully.
5. Click Enable newly added modules to return to the List tab for the Extend page.
6. Find your newly added module in the list of modules, select its checkbox, and click Enable.  
Your newly added module is now enabled and available for you to use in the Developer Portal.

Note: The following modules are unsupported and their installation is blocked in the Developer Portal:

- `backup_migrate`
- `content_sync`
- `delete_all`
- `devel_themer`
- `domain`
- `php`
- `theme_editor`

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Deleting custom modules

You can delete custom modules from your Developer Portal if you want to remove them entirely from your site.

### Before you begin

---

You must have administrator access to complete this task.

Note that any custom modules that you want to delete must be disabled first, including disabling any associated sub-modules. For more information, see [Disabling modules](#). Any modules that are shipped with IBM® API Connect cannot be deleted.

### Procedure

---

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. In the administrator dashboard, click Extend.
4. Click the Delete tab.  
The list of disabled custom modules that can be deleted is displayed. Note that any disabled modules that have sub-modules that are still enabled, are prevented from being disabled.
5. After you have found the module that you want to disable, select the check box next to the module, then click Delete.
6. At the confirmation dialogue, select Delete to delete the module completely from your site, or select Cancel to cancel the operation.

### Results

---

You successfully deleted the module.

### What to do next

---

To install a module, see [Installing custom modules](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Disabling modules

You can disable an entire module in the Developer Portal if you want to improve performance, or remove functionality.

### Before you begin

---

You must have administrator access to complete this task.

Note that the disable process removes all data related to a module.

### Procedure

---

1. Log in to the Developer Portal as an administrator.
2. If the administrator dashboard is not displayed, click Manage to display it.
3. In the administrator dashboard, click Extend.  
The List tab for the Extend page opens, and the list of installed modules is displayed. The list shows all the modules that are installed, some of which are enabled (displayed with a selected check box), and some of which are not enabled (where the check box is not selected).
4. Click the Disable tab.  
The list of modules that can be disabled is displayed. Note that any modules that are required by other enabled modules, are prevented from being disabled.
5. Either enter the name of the module that you want to disable in the Filter by name or description field, or scroll through the list of modules.
6. After you have identified the module that you want to disable, select the check box next to the module, then click Disable.
7. At the confirmation dialogue, select Disable to disable the module and remove any data that relates to it, or select Cancel to cancel the operation.

### Results

---

You successfully disabled the module.

### What to do next

---

To enable a module, return to the List tab, select the check box next to the module that you want to enable, and click Enable.

You can also delete custom modules completely from your Developer Portal site; see [Deleting custom modules](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Troubleshooting guide for the Developer Portal

Use this guide to help you diagnose and resolve some of the more common Developer Portal issues in IBM® API Connect Version 2018. For example, why is the Developer Portal not connecting to the Management server, why is my site not created, and why can't I log in?

### About

---

The following sections provide advice about how to diagnose and resolve some common problems. This information can help you to identify whether a specific IBM API Connect Version 2018 component is failing, whether it's an environmental problem, and whether you should raise a support request. Some advice includes reviewing specific log files so you can narrow down the source of the problem. If you need to raise an IBM Support Service Request (formerly known as PMRs) these logs, along with all of the logs that are described on the [IBM API Connect Mustgather - Portal 2018](#), must be supplied to the IBM Support team. If any of the issues relate to interactions with the Management server, then logs from the Management server also need to be supplied, see [Mustgather for Management server 2018](#).

1. [Why is the Developer Portal not connecting to the Management server?](#)
2. [Why is my Developer Portal site not being created?](#)
3. [Why can't I register a user?](#)
4. [Why can't I login?](#)
5. [Why are changes to Products not appearing in the Developer Portal?](#)
6. [Why am I having problems with my Developer Portal user interface?](#)
7. [Why am I have problems with my Kubernetes system?](#)
8. [Why am I having problems with my Developer Portal backup?](#)
9. [Why am I having problems installing Drupal 8 based custom modules or sub-themes into the Drupal 9 based Developer Portal?](#)

### Why is the Developer Portal not connecting to the Management server?

---

Problems with connecting to the Management server can be due to connectivity or TLS issues. You should check the following areas:

- Can the Management server actually reach the Developer Portal?
- Is SSL pass-through enabled or required on any load balancer that sits between the Management server and the Developer Portal? For more information, see [Setting the security mode of the Portal Director](#).

If you need to open a Service Request, include the following information:

- The [IBM API Connect Mustgather - Portal 2018](#) logs
- Details of the date and time that the connection attempts were made.

### Why is my Developer Portal site not being created?

---

If you are having problems creating a site, you should take the following actions to help diagnose the issue:

1. Tail the nginx container log to confirm that the site create request reached the portal, for example:

```
kubectl logs rb93d4fb118-apic-portal-nginx-7c7b464fb6-vvx5m
```

2. Look in the container log for a site create request, for example:

```
POST /portal-service-configuration-create HTTP/1.1" 200
```

- a. If you don't find a site create request in the log, then the create request is not reaching the Developer Portal from the Management. Follow the instructions in [Why is the Developer Portal not connecting to the Management server?](#) to diagnose the connectivity issues.
- b. If a site create request is in the log, then further diagnostics can be found in the admin pod log, for example:

```
kubectl logs -f rb93d4fb118-apic-portal-www-p4bhj -c admin
```

Look in the admin pod log for messages that refer to `create_site`, as well as the name of the site.

### Why can't I register a user?

---

If you are having problems registering a user, take the following actions to help diagnose the issue:

- The user registration links in the emails are only valid for 24 hours, so check that the system time is correctly configured across all of the pods.
- If emails aren't being received, check that the email server is correctly configured; see [Configuring an email server for notifications](#). Also, if possible, check whether other emails such as publish approval alerts or password reset emails are being received.

If you need to open a Service Request, please take the following steps:

1. Note which user registry type is being used, such as Local User Registry (LUR), or LDAP user registry.
2. Enable the debug REST server and method trace in the Developer Portal UI. Click Configuration > Development > IBM Development on the administrator dashboard. Then select the Enable method entry / exit trace and Enable API Manager REST interface debug check boxes. Click Save configuration.
3. Re-create the issue, making a note of the time, the username, and any messages displayed on the screen.
4. Gather the web container log for the www pod, for example:

```
kubectl logs portal-apic-portal-www-zhqxf web > web.log
```

For more information, see [IBM API Connect Mustgather - Portal 2018](#).

## Why can't I login?

If users are having problems with logging in to the Developer Portal, please take the following steps before opening a Service Request:

1. Note which user registry type is being used, such as Local User Registry (LUR), or LDAP user registry.
2. Enable the debug REST server and method trace in the Developer Portal UI. Click Configuration > Development > IBM Development on the administrator dashboard. Then select the Enable method entry / exit trace and Enable API Manager REST interface debug check boxes. Click Save configuration.
3. Recreate the issue, making a note of the time, the username, and any messages printed to the screen.
4. Gather the web container log for the www pod, for example:

```
kubectl logs portal-apic-portal-www-zhqxf web > web.log
```

For more information, see [IBM API Connect Mustgather - Portal 2018](#).

## Why are changes to Products not appearing in the Developer Portal?

To determine whether the problem is with the Portal server (either the nginx or the www pod), the Management server, or the network, please make the following diagnostic checks.

Portal nginx pod

1. Tail the nginx pod log on the Portal, for example:

```
kubectl logs -f portal-apic-portal-nginx-866c977c4c-n9gbf
```

2. Make a publish request to the Catalog (note that the request must be a Publish or Retire operation, not a Stage operation).
3. In the log being tailed, see if the following message appears:

```
...POST /event-create...
```

If the POST message appears in the nginx log, then the publish webhook that was sent from the Management server did arrive at the Portal. If there is an error message in place of a POST message, then the problem is likely to be with the nginx pod. If there is no message at all, then it's likely that the webhook from the Management server is not reaching the Portal and there is a network problem.

Portal www pod

1. Tail the www pod admin container log on the Portal, for example:

```
kubectl logs -f portal-apic-portal-www-dtbzw -c admin
```

2. Make a publish request to the Catalog (note that the request must be a Publish or Retire operation, not a Stage operation).
3. In the log being tailed, see if the following message appears:

```
Received webhook event 'product_lifecycle' for catalog
```

If no such message appears, it suggests that the webhook is either not arriving at the Portal, or not getting beyond the nginx pod (see the previous section [Portal nginx pod](#)). Any errors that the admin container has in processing the webhook will be logged in the www pod admin container log.

Management server apim pod

1. Tail the apim pod log on the Management server, for example:

```
kubectl logs -f apiconnect-apim-v2-86b464c4f5-4cx6r
```

2. Make a publish request to the Catalog (note that the request must be a Publish or Retire operation, not a Stage operation).
3. In the log being tailed, see if there any error messages.

Search for the word 'sending', you should see the message **webhook: sending successful** if the webhook is sent, or error messages indicating why if the webhook is not sent.

## Why am I having problems with my Developer Portal user interface?

If you are having user interface (UI) problems, make the following checks:

- Try clearing the cache from your browser, try using private browsing, and try using different browsers - see whether there are any differences in behavior.
- Check your browser window size and screen resolution, and try increasing and decreasing the settings.
- Investigate the browser developer tools console and network tabs, as they might provide some diagnostic clues.
- Try clearing the Drupal caches from within the Developer Portal; see [Clearing the server caches](#) for details.

If you need to open a Service Request, along with the [IBM API Connect Mustgather - Portal 2018](#) logs, also include details of the time and date when the problem was reproduced. In addition, an export of the browser developer tools console and network output would be helpful.

## Why am I have problems with my Kubernetes system?

Kubernetes provides a DNS server as one of the pods in the kube-system namespace (unless you have configured your Kubernetes setup differently), and sometimes this DNS server can silently fail. When the Portal www or db pods restart, they check that their own hostname and IP address is in the service that is defined for the pod. If the DNS server is down, this check fails. An indication that the DNS server is down is seeing the following log lines printed endlessly in the admin or db containers:

```
[ admin stdout] 2cd141dc:2cd141dc:2cd141dc 2018-10-08 07:27:54: service-ready:
[ admin stdout] 2cd141dc:2cd141dc:2cd141dc 2018-10-08 07:27:54: service-all:
[ admin stdout] 2cd141dc:2cd141dc:2cd141dc 2018-10-08 07:27:54: Waiting for sleep in service all check. Pid(s) 352 Seconds
24
[ admin stdout] 2cd141dc:2cd141dc:2cd141dc 2018-10-08 07:27:59: Finished waiting for sleep in service all check. Pid(s)
352 Seconds 29
```

```
[ admin stdout] 2cd141dc:2cd141dc:2cd141dc 2018-10-08 07:28:01: WARNING 5: r4c09b98d02-apic-portal-admin-all doesn't include this pod or r4c09b98d02-apic-portal-director does. Checking again in 5 seconds.
```

To fix this problem, delete the DNS pod in the kube-system namespace to force it to restart, for example:

```
kubect1 -n kube-system delete pod kube-dns-b76db4f7f-n41vt
```

Where `kube-dns-b76db4f7f-n41vt` is the full name of the DNS pod.

Note: To find out the full name of the DNS pod in your Kubernetes system, run the following command:

```
kubect1 -n kube-system get pods
```

This command will return data like the following example:

```
$ kubect1 -n kube-system get pods
NAME READY STATUS RESTARTS AGE
default-http-backend-5bccfbd8c-mjnnb 1/1 Running 0 4d
etcd-minikube 1/1 Running 0 4d
kube-addon-manager-minikube 1/1 Running 0 4d
kube-apiserver-minikube 1/1 Running 0 4d
kube-controller-manager-minikube 1/1 Running 0 4d
kube-dns-6f4fd4bdf-7zslq 3/3 Running 0 4d
kube-proxy-mcdmj 1/1 Running 0 4d
kube-registry-proxy-99qml 1/1 Running 0 4d
kube-registry-v0-zt8pv 1/1 Running 0 4d
kube-scheduler-minikube 1/1 Running 0 4d
kubernetes-dashboard-77d8b98585-dczcs 1/1 Running 0 4d
nginx-ingress-controller-57bcfc76d6-z775d 1/1 Running 0 4d
storage-provisioner 1/1 Running 0 4d
tiller-deploy-587df449fb-bpwhq 1/1 Running 0 4d
```

## Why am I having problems with my Developer Portal backup?

If you attempt to back up your portal system or site:

```
apicup subsys exec <portal_subsystem> backup-system|backup-site|backup-all
```

You might encounter the following message:

```
curl: (6) Could not resolve host: apic-portal-apic-portal-director
```

This error is the result of a failure of the SFTP backup file transfer to the backup system, caused by missing authentication with the portal node. To correct this error, `ssh` from the backup system to the portal node, and accept the authentication key when prompted:

1. Enter the portal-www admin container, for example:

```
kubect1 exec -it WWW_POD_NAME -c admin bash
```

2. `ssh` to your backup server and accept any authentication prompts, for example:

```
ssh BACKUP_SERVER_HOSTNAME
```

When you have successfully authenticated with your backup server, you can close the session. This action needs to be done only once. Portal back ups can then run without any issues.

## Why am I having problems installing Drupal 8 based custom modules or sub-themes into the Drupal 9 based Developer Portal?

From IBM API Connect 2018.4.1.17, the Developer Portal is based on the Drupal 9 content management system. If you want to install Drupal 8 custom modules or sub-themes into the Drupal 9 based Developer Portal, you must ensure that they are compatible with Drupal 9, including any custom code that they contain, and not using any deprecated APIs, for example. There are tools available for checking your custom code, such as [Drupal check](#) on GitHub, which checks Drupal code for deprecations.

For example, any Developer Portal sites that contain modules or sub-themes that don't contain a Drupal 9 version declaration will fail to upgrade, and errors like the following output will be seen in the `admin` logs:

```
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: Checking theme: emeraldgreen
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '' found for emeraldgreen
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: Checking theme: rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '8.x' found for rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: Found themes incompatible with
Drupal 9: emeraldgreen rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:34:49: check_d9_compat: ERROR: /tmp/restore_site.355ec8 is NOT
Drupal 9 compatible
...
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: Checking module: custom_mod_1
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '' found for custom_mod_1
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: Checking module: custom_mod_2
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Incompatible
core_version_requirement '8.x' found for custom_mod_2
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: Found modules incompatible with
Drupal 9: emeraldgreen rubyred
[ queue stdout] 14834 729319:355ec8:a7d29c 2021-09-04 20:44:49: check_d9_compat: ERROR: site1.com is NOT Drupal 9
compatible
```

To fix version compatibility errors, all custom modules and sub-themes should declare a `core_version_requirement` key in their `*.info.yml` file that indicates Drupal 9 compatibility. For example:

```
name: Example module
type: module
description: Purely an example
core: 8.x
core_version_requirement: '^8 || ^9'
package: Example module
```

```
# Information added by Drupal.org packaging script on 2020-05-31
version: '8.x-1.3'
project: 'example_module'
datestamp: 1590905415
```

This example specifies that the module is compatible with all versions of Drupal 8 and 9. For more information, see [Let Drupal know about your module with an .info.yml file](#) on the drupal.org website.

If you have a backup of a site that you need to restore, and are getting the version compatibility error, but the module or theme \*.info.yml file cannot be changed easily, then you have two options. Either:

- Add an environment variable for the `www` pod of the `admin` container stating `SKIP_D9_COMPAT_CHECK: "true"`. However, if you choose this method, you must be positive that all of the custom modules and themes for your sites are Drupal 9 compatible, as otherwise the sites may end up inaccessible after the upgrade or restore. Add the environment variable by using the following instructions:
  - Create an extra values file to contain the environment variable, as follows:

```
apic-portal-www:
  admin:
    env:
      SKIP_D9_COMPAT_CHECK: "true"
```

Save the file as `d9compat.yaml`, and run the following command:

```
apicup subsys set <portal_subsystem_name> extra-values-file d9compat.yaml
```

Then, update the portal with the updated setting by running the following command:

```
apicup subsys install <portal_subsystem_name>
```

Or:

- Extract the site backup, edit the relevant files inside it, and then tar the backup file again. Note that this procedure will overwrite the original backup file, so ensure that you keep a separate copy of the original file before you start the extraction. For example:

```
1. mkdir /tmp/backup
2. cd /tmp/backup
3. tar xfz path_to_backup.tar.gz
4. Edit the custom module and theme files to make them Drupal 9 compatible, and add the correct core_version_requirement setting.
5. rm -f path_to_backup.tar.gz
6. tar cfz path_to_backup.tar.gz
7. cd /
8. rm -rf /tmp/backup
```

## Related information

---

- [IBM Support](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Setting the security mode of the Portal Director

You can set the security mode of the Portal Director for a Developer Portal service in your API Connect on-premise cloud.

### About this task

---

By default, the Portal Director rest server is started in **secure** mode, meaning that requests to it are carried out using mutual TLS (mTLS). In order for this process to work, any load balancer on the Portal Director must allow SSL pass-through so that client certificates that are sent on incoming requests reach the Director and can be checked. In certain scenarios, it might not be possible for a load balancer to provide SSL pass-through. In that case, the Portal Director must be set to **Insecure** mode so that client certificates on incoming requests are not checked.

**Warning:** It is not recommended to run Portal Directors in production Developer Portal services in **Insecure** mode.

To configure the security mode of the Portal Director belonging to a Developer Portal service in your cloud, complete these steps:

### Procedure

---

1. Get a list of the pods that are running on the virtual machine where the Developer Portal service is deployed.

```
kubectl get pods -n <namespace>
```

2. Identify the `portal-www` pod from the resulting list then set the Portal Director's security mode to **Insecure** with the following command:

```
kubectl exec <portal-www-pod> -c admin -- /opt/ibm/services/node_modules/portal/client/client
director_security_mode --secure false
```



## Results

---

The Portal Director is now set to **Insecure** mode. Any new incoming requests are not checked for a valid client certificate.

## What to do next

---

To set the Portal Director back to **Secure** mode, use the following command:

```
kubect1 exec <portal-www-pod> -c admin -- /opt/ibm/services/node_modules/portal/client/client
director_security_mode --secure true
```

For more information on security settings, including cookies, see [How to enable penetration testing, and information about Developer Portal cookies](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Migrating your Developer Portal from Version 5 to Version 2018

Guidance on how to migrate a Developer Portal site from IBM® API Connect Version 5, to IBM API Connect Version 2018.

In IBM API Connect Version 2018 the Developer Portal is based on the open source Drupal 9 content management software, which enables more advanced, flexible, and powerful customization capabilities. However, in IBM API Connect Version 5 the Developer Portal is based on Drupal 7. The migration tooling in Version 2018 will create a new Developer Portal site, and migrate all of the associated APIs and Products, user information, and subscriptions over. However, customizations such as modules and themes cannot be automatically migrated, as the Drupal 9 baseline is very different to Drupal 7. For more information about upgrading from a Drupal 7 baseline to a Drupal 9 baseline, see [Upgrading Drupal](#) in the Drupal documentation.

The following sections take you through the migration process for the Developer Portal, and provide guidance on how to manually migrate any specific customizations to Drupal 9.

- [Migration process for the Developer Portal](#)
- [Portal Delegated User Registries](#)
- [Custom themes](#)
- [Custom modules](#)
- [Custom Rules](#)
- [Custom content types and fields](#)
- [Advanced theme development](#)
- [Changing the page layout](#)
- [Changing the front page](#)
- [Features no longer available in the Developer Portal](#)

Note: The ability to migrate is available only from IBM API Connect Version 5.0.8.7 onwards. If your deployment is at an earlier version, you must upgrade before migrating. See [Upgrading your API Connect cloud](#) for information about upgrading in Version 5. For more information about the supported versions for migration, see [Migrating a Version 5 deployment to Version 2018](#).

---

## Migration process for the Developer Portal

The migration process for the Developer Portal migrates all of the following information to Version 2018 for each site in your cluster:

- User information (if Portal Delegated User Registry is enabled, the process is slightly different; see [Portal Delegated User Registries](#)).
- Products and APIs.
- Consumer organizations.
- Applications.
- Subscriptions.

For information about the main migration process, see [Migrating a Version 5 deployment to Version 2018](#).

---

## Portal Delegated User Registries

If your IBM API Connect Version 5 Developer Portal uses Portal Delegated User Registry (PDUR), you must first export the user information that is stored in the Version 5 Developer Portal local database. This user information can then be fed into the migration tooling so that the correct user registries can be created in Version 2018.

For more information, see [Exporting Portal Delegated User Registry user information](#).

---

## Custom themes

If your Version 5 Developer Portal is using a custom theme, you'll need to create the custom theme again for the Drupal 9 baseline. Start by creating a sub-theme of the new Version 2018 Developer Portal site by using the theme generator, and then manually customize the sub-theme to your specifications for Drupal 9 by using Cascading Style Sheets (CSS), or Sass CSS (SCSS). Note that using SCSS in your theme makes it far simpler to change colors. We've also added a color template theme generator, which means you can quickly generate a base color theme on which to build your customized site branding. For more information, see [Creating a sub-theme](#).

Note, directly editing the API Connect theme is not permitted or supported, as edited versions of these files are overwritten when a fix pack or iFix is installed.

Other useful resources:

- [Tutorial: Creating a custom theme for the Developer Portal](#) - takes you through how to use the theme generator, customize a theme, and install a custom theme.

- [Changing the site logo](#) - how to change your site logo.
- [Changing the shortcut icon](#) - how to change the shortcut icon (favicon).
- [Creating sub-themes](#) - Drupal documentation about creating sub-themes.
- [Theming differences between Drupal 6, 7 & 8](#) - Drupal documentation about theming differences.

## Custom modules

---

If your Version 5 Developer Portal is using any custom modules, you will need to create these custom modules again for the Drupal 9 baseline. For more information on writing custom modules, see [Extend](#).

Other useful resources:

- [Creating custom modules](#) - Drupal documentation about creating custom modules.
- [Getting started creating custom modules](#) - Drupal documentation giving background information and prerequisites needed for custom module development.
- [Understanding hooks](#) - Drupal documentations about hooks.
- [Drupal API hooks](#) - Drupal documentation about the hooks that are available.

## Custom Rules

---

If your Version 5 Developer Portal is using any custom Rules, you will need to create them as custom modules for the Drupal 9 baseline. You cannot export the Drupal 7 Rules, and then import them into the Drupal 9 baseline, as the events and actions are implemented in a different way. For more information about custom modules, see the section about [Custom modules](#).

## Custom content types and fields

---

If you've created custom content in your Version 5 Developer Portal, you can export the following custom content types:

- Any custom fields added to APIs, Applications, Products, Consumer Organizations, or users.
- Any custom image files for Applications, APIs, or Products.
- Any attachment files added to APIs or Products.

When the migration tooling is available, you can then import this content into your Version 2018 Developer Portal. For information about exporting custom content from Version 5, see [How to export custom Developer Portal content](#).

Any other custom types or fields content that you've created in Version 5, will need to be created again for the Drupal 9 baseline. For information about creating custom content types, see the following topics:

- [Creating content types](#)
- [Adding fields to content types](#)
- [Adding custom fields to user records](#)

## Advanced theme development

---

If your Version 5 Developer Portal is using modified content type templates, you must create these templates again for the Drupal 9 baseline. In Drupal 9 the templates are Twig files that define the actual HTML output for a particular piece of content. You can override a Twig template by copying the original template into the templates folder of your custom sub-theme, and then editing the template to your specification. The Developer Portal then uses your template in preference to the original one. For more information, see [Applying a modified content type template](#), and [Creating a sub-theme](#). You can also follow a guided example of how to create a new theme, see [Tutorial: Creating a custom theme for the Developer Portal](#).

Note:

Modifying Twig templates allows very fine-grained control over the structure of pages. However, it also means you might miss new features and defect fixes that are made to the original templates, as your Developer Portal is using your overrides instead of the originals. So, if you override a template, it is your responsibility to check the templates in the latest API Connect Version 2018 releases, and to ensure that any equivalent changes that are needed to your overrides to maintain functional equivalence are made.

## Changing the page layout

---

If your Version 5 Developer Portal is using a different page layout, you will need to create this layout again for the Drupal 9 baseline. The page layout is defined by configuring which blocks appear in which regions. Blocks are controlled by using the Structure\_>Block layout options on the Developer Portal administrator dashboard.

For information about how to configure the page layout, see [Adding and changing the blocks displayed on Developer Portal pages](#).

## Changing the front page

---

If your Version 5 Developer Portal is using a different front page, you will need to create this front page again for the Drupal 9 baseline. The front page is configured from the Structure\_>Pages options on the Developer Portal administrator dashboard.

For information about how to configure the front page, see [Configuring the front page](#), and [Adding custom pages](#).

Further options that you might want to consider when designing your front page include:

- Change the banner block content or image: [Changing the front page banner block](#).
- Display featured API Products by creating a featured content block: [Changing the front page Featured Content block](#).
- Create custom blocks with your own HTML content: [Adding and changing the blocks displayed on Developer Portal pages](#).
- Set up the social block to include your organization Twitter feed: [Integrating Twitter data into the social block](#).

## Features no longer available in the Developer Portal

---

The following features that are available in IBM API Connect Version 5, are not available in IBM API Connect Version 2018:

- Security questions - these can now be created inside an OpenID Connect provider.
- Two-factor authentication - this can now be performed inside an OpenID Connect provider.
- Support tickets - no Drupal 9 equivalent module is currently available.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Developer Portal tutorials

Tutorials for using the Developer Portal.

The following guided examples help you through some of the most common user scenarios in the Developer Portal, as well as some of the more complex areas.

| Developer Portal tutorials by category   |
|--|
| For API consumers<br><a href="#">Create an Application</a>   |
| User authentication<br><a href="#">Adding a field to the sign up form</a><br><a href="#">Adding validation to a field on the sign up form</a><br><a href="#">Adding a field group to group fields in a content type</a>  |
| Getting started configuring the Developer Portal<br><a href="#">Creating the Portal</a><br><a href="#">Creating a private Portal site</a><br><a href="#">Customizing themes</a>  |
| Advanced customization<br><a href="#">Changing the front page</a><br><a href="#">Grouping products by category</a><br><a href="#">Adding a custom block to the front page</a><br><a href="#">Adding a custom block to a page other than the front page</a><br><a href="#">Using avatars</a><br><a href="#">Using image optimization</a><br><a href="#">Adding a Back to top button</a><br><a href="#">Adding a zoom effect</a><br><a href="#">Creating a drop-down menu link</a><br><a href="#">Displaying blog posts on the front page</a><br><a href="#">Displaying tweets on the front page</a><br><a href="#">Changing the profile page to display firstname lastname, instead of username</a><br><a href="#">Using a custom weighting sort order on the product list page</a><br><a href="#">Configuring the RobotsTxt file</a> |

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Creating the Developer Portal

### About this task

In this tutorial, you are going to complete the following lessons:

### Enabling the Developer Portal through the API Manager

By proceeding with the following steps, you create an administrator account:

1. Log in to the API Manager
2. Click Manage, then click Add, and select Create a catalog.
3. From the drop-down **Select user**, then enter a **Title** for your catalog, for example `Production`.

← Create Catalog

### Create Catalog

Enter the catalog summary details; you can fully configure the catalog after you create it

**Select user**

apic demo (apicdemo1), apicdemo1@outlook.com

**Title**

Production

**Name**

production


Cancel Create

4. Click Create.


IBM API Connect  
API Manager

## Manage

A catalog hosts a collection of API products that are visible in the associated developer portal when published




Production



Sandbox  
Sandbox Catalog

5. Click your new catalog, for example **Production**.

6. In Production, click  Settings.
7. Click Gateway Services. If it says **No items found**, click Edit, select **gateway\_service\_1**, and click Save.

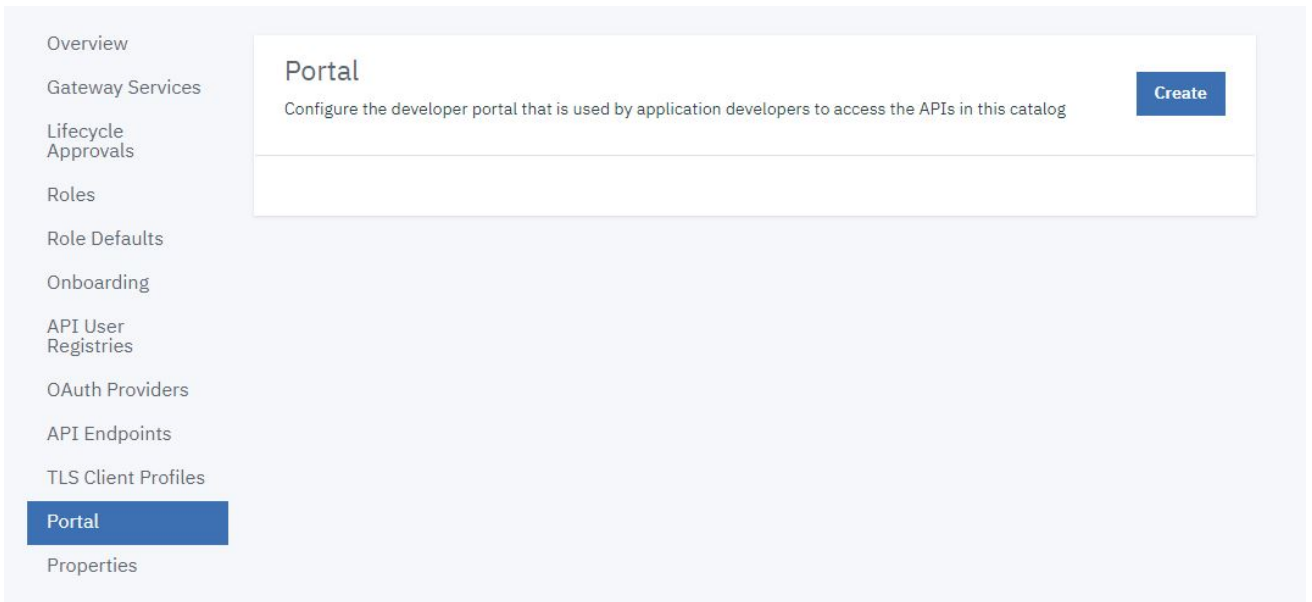
← Manage / Production

### Enable Gateway Services

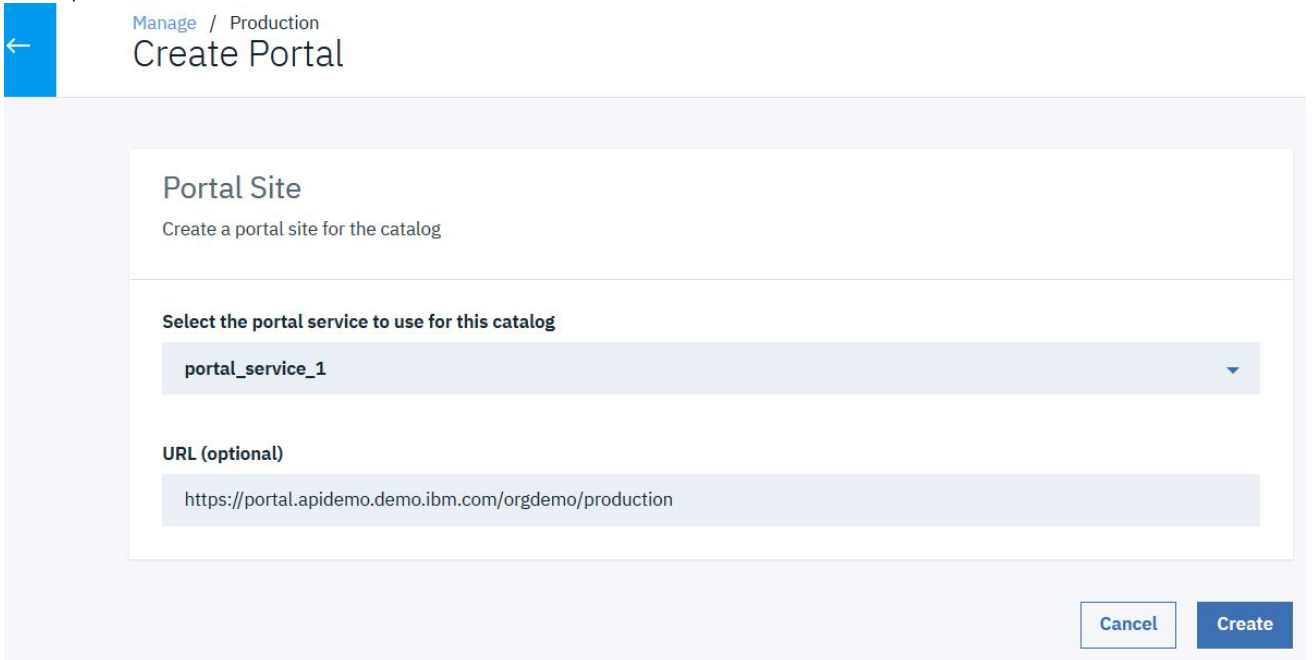
| <input checked="" type="checkbox"/> | TITLE             | TYPE                  |
|-------------------------------------|-------------------|-----------------------|
| <input checked="" type="checkbox"/> | gateway_service_1 | DataPower API Gateway |

Cancel Save

8. Click Portal.
9. Click Create.



10. Select a portal service to use. Click Create.



After a few minutes, you receive an email with a link to your Developer Portal site for that catalog. The link is a single use only link for the administrator account.

Administrator ([admin](#)).

Your Developer Portal site has been created. You can log in as the '[admin](#)' user by using the following one-time log in link. After you have logged in with the following link, you must change the password for the '[admin](#)' user account immediately.

[https://portal.apidemo.demo.ibm.com/production/production/user/reset/1/1566308312/5QdHaGExu2\\_M5cAx1FcchbpvDhf9xHLiyX\\_I9ZMYIQ/new](https://portal.apidemo.demo.ibm.com/production/production/user/reset/1/1566308312/5QdHaGExu2_M5cAx1FcchbpvDhf9xHLiyX_I9ZMYIQ/new)

11. Click the email link. Click Sign in.

This is a one-time login for *admin*.  
 Click on this button to log in to the site and change your password.  
 This login can be used only once.

[Sign in](#)

12. Change the password. Click Submit.

### Change your password

Change your password.

**Password \***

Password strength: Fair

**Confirm password \***

[Submit](#)

The portal site for the Production catalog is now active. The admin account cannot create applications or subscriptions. You must create a new account to complete those tasks.

Your password has been changed.

**Brace yourselves. APIs are coming.**  
 Explore, subscribe to and be creative with our APIs. We can't wait to see what you come up with!

[Explore API Documentation](#)

**Sign up**

[Create a new account](#) and get started with our APIs. It's free to join.

**Explore our APIs**

Take a look at our [API products](#) and quickly find APIs to construct a fully featured application

**Create**

Subscribe to a plan and create your application to make use of our APIs.

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Enabled a Developer Portal site and created an administrator account.

---

## What to do next

- [Create a Developer Account and Application](#)

---

## Related information

- [Importing an API](#)

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

# Tutorial: Creating a private Developer Portal site

You can create a private version of the Developer Portal, so that authenticated users only can access content on the site. You might want to do this if you want to restrict public access to your content unless they are logged in.

---

## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

---

## About this tutorial

In this tutorial, you are going to restrict access for all the content to authenticated users only. This gives access to all users from **Content Author** to **Superuser**, as they are all required to authenticate to access their roles. You will then add the login form to the home page, so that public users will be able to view the public home page that contains the login form only.

This tutorial takes you through the following steps:

1. [Export your configuration settings](#).
2. [Restrict all permissions to authenticated users](#).
3. [Restrict blocks to authenticated users](#).
4. [Add login form to the home page](#).

---

## Export your configuration settings

In case you make an error, or want to revert all of the changes made during this tutorial, you can back up your configuration settings to your machine before you make any changes.

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Configuration > Development > Configuration synchronization > Export.
4. Click **Export**.

---

## Restrict all permissions to authenticated users

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to People > Permissions.
3. Clear all check boxes that are associated with **ANONYMOUS USER**.
4. Click **Save permissions** at the end of the page.

---

## Restrict blocks to authenticated users

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Structure > Block layout.
3. Find the row that is labeled **Main menu** and click **Configure**.
4. Find the setting **Roles** under the category **Visibility**.
5. Check the box **Authenticated user**.
6. Click **Save block**.
7. Repeat from step 3 for the **Footer menu**.
8. Click **Save blocks**.

---

## Add login form to the home page

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Structure > Pages.

3. Click **Edit** beside the **Welcome** page.
4. Navigate to **Panels**, **Content**.
5. Click + **Add new block**.
6. Find and click **User login**, under the **FORMS** section.
7. Clear the **Display title** check box.
8. Click **Add block**.
9. Rearrange the new login block so that it is underneath the **Welcome Banner**.
10. Click **Update and save**.

## What you did in this tutorial

---

On your Developer Portal, you restricted access for all the content to authenticated users only. Now when you access your home page as an unauthenticated user, you see that the login form appears underneath the welcome banner. Navigating to other URLs on the site presents an **unauthorized** page to appear.

When you log in, the login form disappears from the home page and you are able to access all of the pages that are visible to your role.

## What to do next

---

If you exported your configuration and now want to revert to your configuration as it was before this tutorial, then complete these steps:

1. Navigate to **Configuration**, **Development**, **Configuration synchronization**, **Import**.
2. Click **Chose file** and select your exported backup.
3. Click **Upload**.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

# Tutorial: Creating Accounts and Applications on the Developer Portal

---

This tutorial shows you how to create a developer account and register an application on the Developer Portal.

## Before you begin

---

You must have a Developer Portal installed, and created and published at least one API product.

If you do not have a Developer Portal available, complete the tutorial [Creating a Developer Portal](#).

If you do not have an API product available, complete the tutorial [Importing an API](#).

## About this task

---

In this tutorial you are going to complete the following lessons:

- [Creating a developer account in the Developer Portal](#)
- [Creating a developer application in the Developer Portal](#)
- [Testing the Branches API in the Developer Portal](#)

Note: The administrator account cannot create an App, or register to an App. To create or register to an App, you must have a developer account. A developer account that has been assigned administrator privileges can create and register Apps.

## Creating a developer account in the Developer Portal

---

Take the following steps to create a developer account:

1. Open a Developer Portal at the previously specified URL, then click **Create an account** and complete the fields. If the **Create an account** link is not visible, log out as **Admin** from the Developer Portal first.  
Note: Your email is used as your user name for the Developer Portal. The email address must be different from the address used for creating the administrator account.
2. Go to the home page of the Developer Portal
3. Click **Create an account**.
4. Complete the sign up form. The email address you provide must be functional. Click **Sign up**. A confirmation message is sent to the email address provided in the form.




Consumer organization \*

Password \*

Password strength: Fair

Confirm password \*

Your password meets the password policies required for this site



What code is in the image? \*

[Sign up](#)

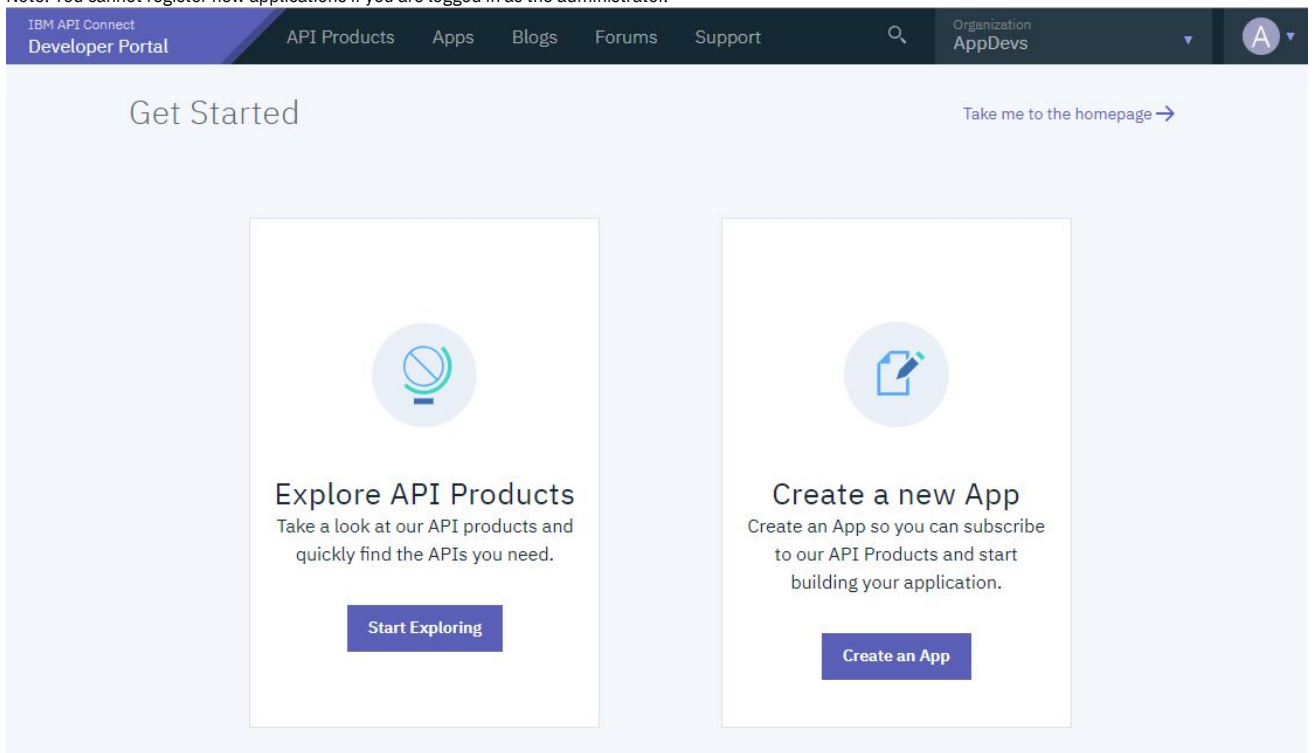
Already have an account? [Sign in](#)

5. Click the account activation link in the confirmation email to activate your account.
6. To use the Developer Portal, click Login and sign in with the user credentials you specified.

## Creating a Developer Portal Application

To create and register a new App:

1. In the Developer Portal, click Create an App.  
Note: You cannot register new applications if you are logged in as the administrator.



The screenshot shows the IBM API Connect Developer Portal interface. The top navigation bar includes 'IBM API Connect Developer Portal', 'API Products', 'Apps', 'Blogs', 'Forums', 'Support', a search icon, and a user profile icon for 'Organization AppDevs'. The main content area is titled 'Get Started' and features two prominent cards. The first card, 'Explore API Products', includes a magnifying glass icon and a 'Start Exploring' button. The second card, 'Create a new App', includes a document icon and a 'Create an App' button. A link 'Take me to the homepage' is located in the top right corner of the main content area.

2. Enter the following values for the application that is being registered.

Table 1. Values for registering the application

| Field Name                        | Value              |
|-----------------------------------|--------------------|
| Title                             | AppOne             |
| Application OAuth Redirect URL(s) | http://example.com |

Create a new application

Title \*

AppOne

Description

Application OAuth Redirect URL(s)

http://example.com

Add another item

Cancel Submit

3. Click Submit.

4. The opportunity to discover the key and secret for the new app is presented. Click Show to see and take note off these values.

Application created successfully.

## API Key and Secret

The API Key and Secret have been generated for your application.

Key

7aae01f7a71eeb5ca59ed0d9ae31a6ac  Show

Secret

a458cbf26c655381d2812f184eea8266  Show

The Secret will only be displayed here one time. Please copy your API Secret and keep it for your records.

Continue

5. Click Continue.

Applications



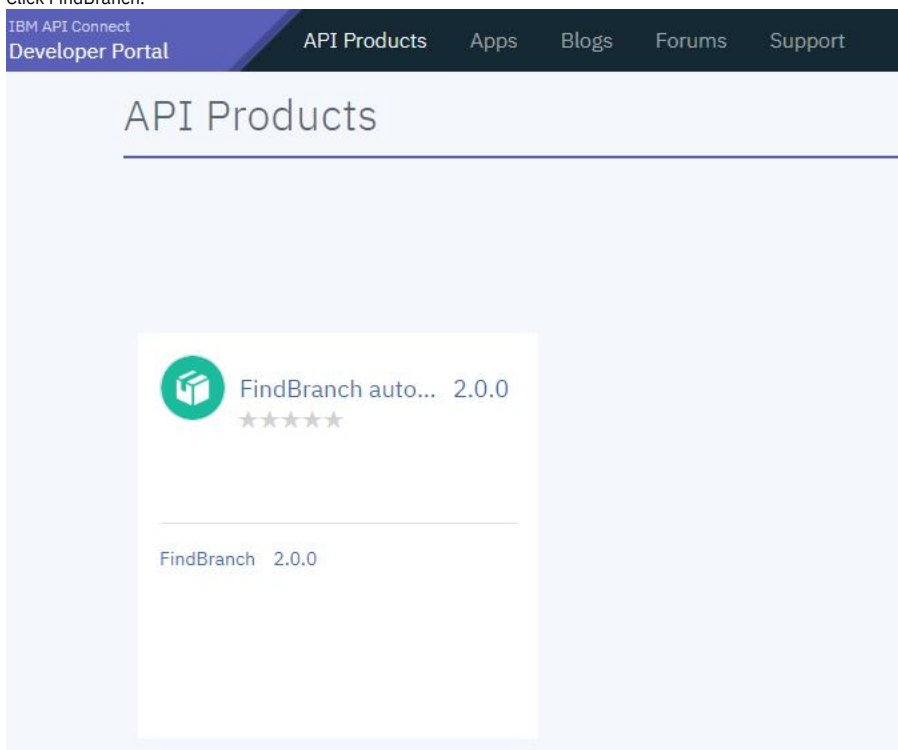
Dashboard Subscriptions

Application OAuth Redirect URL(s)  
http://example.com

You see a usage statistics page.

## Subscribing to an API in the Developer Portal

1. Click API Products.
2. Click FindBranch.



3. Click Subscribe.

## APIs

 FindBranch 2.0.0

## Plans

### Default Plan

Subscribe

View details ▾

4. Under the Application heading, click Select App for your new application.

## Select Application



AppOne

Select App

5. Click Next.

## Confirm Subscription

A subscription will be created for the product, plan and application you selected.

**Product**  
FindBranch auto product

**Plan**  
Default Plan


**Application**  
AppOne


[Previous](#) [Next](#)

6. Click Done.

## Subscription Complete

Your application is now subscribed to the selected plan.

 FindBranch auto p...  
Default Plan

 AppOne

[Explore more products](#)


[Done](#)

## Testing An API in the Developer Portal

1. Click API Products in the Developer Portal dashboard.
2. Click a Product, such as FindBranch auto product , then select the FindBranch API from the list.

### FindBranch auto product

APIs

 FindBranch 2.0.0

3. Click GET /details.

Filter

**Overview**

GET /details

Definitions

## Overview

Download Open API Document

Type REST

4. Click Try it.

## GET : /details

**Details**

Try it

**GET**      **Production, Development:**      [https://\[redacted\]/tutorial/sandbox/findbran](https://[redacted]/tutorial/sandbox/findbran)

Security

**clientID**

X-IBM-Client-Id      apiKey located in header

5. Click Send.

**Details**

**Try it**

**GET**      **Production, Development:**      [https://\[redacted\]/tutorial/sandbox/findbran](https://[redacted]/tutorial/sandbox/findbran)

Security

**Identification**

---

**Client ID**

AppOne
▼

Parameters

▼ Header

---

**Accept**

application/json
▼

[Reset](#)    [Send](#)

Note: If no response is received, navigate to the URL that is displayed at the beginning of the Try this operation section, in a new browser tab. Accept the security certificate, and then call the operation again.

6. A returned response of 200 OK and the message body are displayed, indicating that the REST API operation call was successful.

[Reset](#) [Send](#)

---

Request

```
GET https://[REDACTED]/tutorial/sandbox/findbranch/details
Headers:
Content-Type: application/json
Accept: application/json
X-IBM-Client-Id: bdd47d45d7775a24e2326dc46dbb9dfe
```

Response

```
Code: 200 OK
Headers:
content-type: application/json; charset=utf-8
x-global-transaction-id: 196c55655b0307b1000b1da1
[
  {
    "id": "02e91060-4d49-11e8-a6f5-b75dbc3b13de"
  },
  {
    "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
```

---

## What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a developer account in the Developer Portal.
- Created and registered a new App, and subscribed it to a Plan.
- Tested an API in the Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Creating a custom theme for the Developer Portal

Customize the appearance of your Developer Portal home page, by generating a sub-theme, configuring the overrides.css file, and installing the newly customized theme.

---

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

Important: You're not permitted to include any IBM® API Connect code within any custom themes that you create.

---

### About this tutorial

The way to create a custom theme is to create a custom sub-theme of the standard API Connect theme that the Developer Portal uses by default. The sub-theme inherits all of the resources of the parent theme, and you can then override specific resources to configure the required customizations.

This tutorial takes you through the following lessons:

1. [Creating a sub-theme](#) by using the theme generator.
2. [Customizing the overrides.css file](#).
3. [Installing and enabling your customized theme](#).

---

### Creating a sub-theme

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Click Appearance > Generate sub-theme. The Generate sub-theme window is displayed.
4. Enter a Sub-theme name, and select CSS for the Sub-theme type. (If you prefer, you can select SCSS, but this extension to CSS is for advanced theme developers, and isn't covered by this tutorial.)

## Generate sub-theme ☆

List Generate Update Settings

Home » Administration

The first step in customizing the branding of your Developer Portal is to create a custom sub-theme.

The sub-theme inherits all of the resources of the parent theme, and you can then override specific resources in the overrides.css file to configure your customizations. For more information, see: [Knowledge Center](#)

Complete the form below and you will be presented with a custom sub-theme to download.

**Sub-theme name \***

banka\_theme

A custom theme name, for example: 'mycustom\_theme' or 'banka\_theme'. The name does not need to end in '\_theme' but it is a common convention.

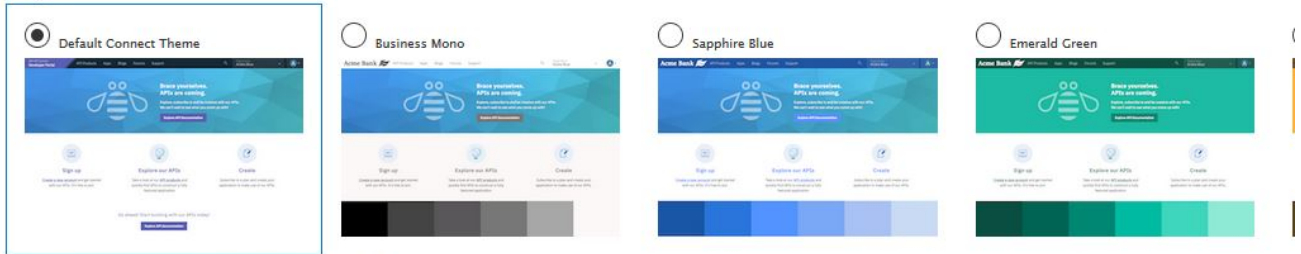
**Sub-theme type**

CSS

SCSS

Your sub-theme can be setup to use either CSS or SCSS. SCSS is an extension to CSS and is for more advanced theme developers.

**Template**



Your sub-themes can use one of several different base templates, either the default connect\_theme, or one of several different color variants.

Generate

5. Select the Default template to base your sub-theme on. You can create a sub-theme based on a color template, however, for this tutorial we'll use the default connect\_theme template.
6. Click Generate.
7. Download the generated sub-theme to a location of your choice, and extract all the files from the .zip file.

## Customizing the overrides.css file

1. Open the overrides.css file in your chosen editor (the file is located in the css folder of the sub-theme that you downloaded).
2. Customize your sub-theme by entering the following elements into the overrides.css file:

```
/* Body of the home page */
body.path-frontpage.contexthome {
  color: #152935;
  background-color: #d1f0f7;
}

/* Header of the home page */
.navbar {
  background-color: #a0a0a0;
}

/* Footer of the home page */
footer.footer {
  background-color: #dee0e2;
}

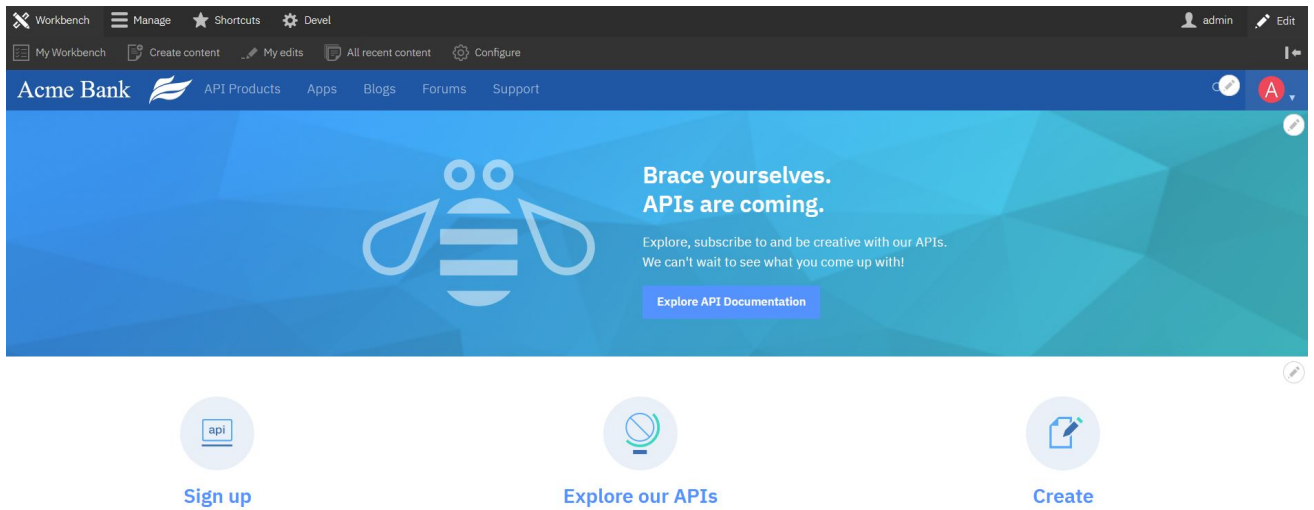
footer.footer ul.nav > li > a {
  color: #454A4C;
}
```

3. Save the overrides.css file.

## Installing and enabling your customized theme

1. After you have finished updating the overrides.css file, compress all the theme files back into the .zip sub-theme file that you downloaded originally. Note that you can also include your own custom logo and favicon in your theme files, and these are then included as part of your theme when it is installed. These files must use the naming convention logo.svg, and favicon.ico, respectively. You can also change your logo and favicon at a later time, see [Changing the site logo](#) and [Changing the shortcut icon](#).
2. In the Developer Portal, click Appearance > Install new theme. The Install new theme window is displayed.
3. In Upload a module or theme archive to install click Choose file, and select your newly updated compressed theme file.
4. Click Install to install the theme onto your site.
5. Click Enable newly added themes, and find your new theme in the list of Disabled themes. Click Enable and set as default to set your new custom sub-theme as the default theme for your site.
6. Return to the Developer Portal home page by clicking Back to site in the upper left of the screen. You can now see your custom theme.





## What you did in this tutorial

In this tutorial, you completed the following tasks:

- Created a sub-theme of your Developer Portal site by using the theme generator.
- Configured some customized home page elements in the `overrides.css` file of your sub-theme.
- Uploaded and installed your customized theme in your Developer Portal site.
- Successfully customized the appearance of your Developer Portal home page.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Using avatars

You can use an avatar generator to better represent the branding of your site, and add a sense of community for your users, than the default name initial that is provided in the Developer Portal.

### Before you begin

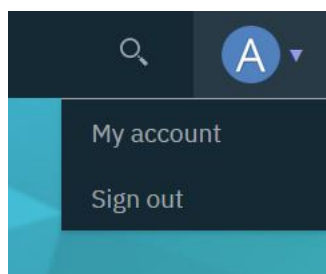
You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

In this task, you add the **Robohash** avatar generator to your Developer Portal site, and assign it as the primary avatar generator for all users.

This tutorial takes you through the following steps:

1. [Install the Robohash avatar generator.](#)
2. [Set Robohash as the preferred avatar generator for your site.](#)
3. [Configure the permissions for the Robohash avatar generator.](#)



The avatar of the admin user displays like this before the block is added.

### Enable the Robohash avatar generator

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend**.

- In the text box, enter `avatar` in the **Block description**.
- Check the **Robohash** box, click Enable.

[+ Install new module](#)

avatar

Enter a part of the module name or description

Below is a list of installed modules. Modules that are installed and enabled are shown with a selected check box. To enable a module,

▼ AVATARS

|                                     |                      |   |
|-------------------------------------|----------------------|---|
| <input type="checkbox"/>            | <b>Adorable</b>      | ▶ Adds an Adorable.io avatar generator for Avatar Kit.        |
| <input checked="" type="checkbox"/> | <b>Avatar Kit</b>    | ▶ Provides a selection of generated avatars for users.        |
| <input type="checkbox"/>            | <b>Gravatar</b>      | ▶ Adds Gravatar.com avatar generators for Avatar Kit.         |
| <input checked="" type="checkbox"/> | <b>Letter Avatar</b> | ▶ Adds a letter avatar generator based on the name of a user. |
| <input checked="" type="checkbox"/> | <b>Robohash</b>      | ▶ Adds Robohash avatar generators for Avatar Kit.             |

[Enable](#)

When the module is enabled, you receive a **Module Robohash has been enabled** message.

## Set Robohash as the preferred avatar generator for your site

- If the administrator dashboard isn't displayed, click Manage to display it.
- Navigate to Configuration > People > Avatar settings.
- Click **+ Add avatar generator**.
- Enter `Robohash` as the label.
- Check **Robohash** for the avatar generator.
- Click Save

**Label \***

Robohash

**Avatar generator \***

**Letter Avatar**

Letter generated by the LetterAvatar library.

**Robohash**

Robots and monsters from Robohash.org

**User preference**

Avatar generator calculated based on user preference.

**User upload**

Image uploaded by the user to the site.

[Save](#)

- You are then taken to the **Edit avatar generator Robohash** page. Select **Type: Robot Heads** and **Background: Transparent**. Click Save.

**Label \***

Robohash

**Type**

- Robots
- Robot Heads
- Monsters

**Background**

- Transparent
- Places
- Patterns

Save

[Delete](#)

8. You are now on the avatar generator page. Drag **Robohash** to be first in the list, and check **Enabled**, then click Save configuration.

You have now set **Robohash** as the preferred avatar generator for your site.

## Configure the permissions for the Robohash avatar generator

The default setting for the **Robohash** avatar generator is to generate avatars for users with **superuser** access only. In this example, the permissions are changed so that all **authenticated users** can access **Robohash**.

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to People > Permissions.
3. Find **Use Robohash** under the permission type **Avatar Kit**.
4. Check **AUTHENTICATED USER** for **Use Robohash** to enable the generator for all authenticated users.

Notice that all the boxes to the right of this user type are then automatically checked because all of those users have the same permissions as an authenticated user, or higher.

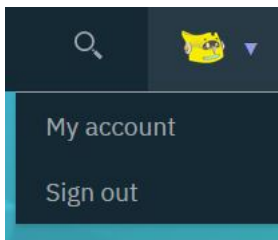
| PERMISSION  | ANONYMOUS USER           | AUTHENTICATED USER                  | FORUM MODERATOR                     | CONTENT AUTHOR                      | ADMINISTRATOR                       | SUPERUSER                           |
|---|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Change own logout threshold   | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| <b>Avatar Kit</b>   |                          |                                     |                                     |                                     |                                     |                                     |
| Administer Avatar Kit<br><i>Warning: Give to trusted roles only; this permission has security implications. Manage settings for Avatar Kit.</i> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Use Letter Generator<br>User can select Letter Generator avatar generator.  | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Use Robohash<br>User can select Robohash avatar generator.  | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

5. Click Save permissions.

## What you did in this tutorial

The avatars of all users now have the **robohash** avatar.

You can check that the avatar of the admin user has changed by looking at your site page.



You can check that the avatars of other users have changed by finding their published content and looking for their avatar. Or, you can navigate to Manage > People, click a user, and see their avatar on their profile page.

**First Name**  
firstnametest

**Last Name**  
lastnametest

**Member for** 2 weeks 1 day

**Picture**



---

## What to do next

You can disable the **robohash** avatar generator.

1. Navigating to Extend > Disable module.
2. Search for **avatar** in the filter text box.
3. Check **Robohash**, click Disable
4. Click Disable.

You can also disable the **robohash** avatar generator by clearing it within Configuration > People > Avatar settings, or, lowering its preference by dragging it lower in the list order. Click Save configuration.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Using image optimization

You can use image optimization on your Developer Portal site to enhance the use of images.

Optimized images do not overload your website and significantly increase its performance compared to using the original images. Your customer satisfaction and brand reputation improve with fast-loading and trimmed visual assets. Automatic optimization can save time and effort on editing and content moderation.

---

## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

---

## About this tutorial

In this task, you create an image style that is called `User picture`, and give it a scale and crop effect. Then, use the `User picture` as the **Picture** field in the user account.

This tutorial takes you through the following steps:

1. [Create the image style `User picture`](#).
2. [Edit the image style `User picture`](#).
3. [Assign `User picture` to the `picture` option in `Account settings`](#).

## Create the image style `User picture`

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Configuration > Media > Image styles**.
4. Click **+ Add image style**.
5. Enter `User picture` in the **Image style name** block. Click **Create new style**

### Add image style ☆

[Home](#) » [Administration](#) » [Configuration](#) » [Media](#) » [Image styles](#)

**Image style name \***

Machine name: `user_picture` [\[Edit\]](#)

**Create new style**

You created the image style `User picture`.

## Edit the image style `User picture`

When you edit this style, you choose the width and height settings. Also, the part of the picture to retain during the crop, for example, the center.

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Configuration > Media > Image styles**.
3. Click **Edit** on the `User picture` style name.

### Edit style `User picture` ☆

**Edit**

[Translate image style](#)

[Home](#) » [Administration](#) » [Configuration](#) » [Media](#) » [Image styles](#)

**Preview**

original (view actual size)



600px

800px

**Image style name \***

Machine name: `user_picture` [\[Edit\]](#)

**EFFECT**



Select a new effect ▼

**Add**

**Save**

[Delete](#)

4. Select **Scale and crop** from **Select a new effect**, then click **Add**.
5. Enter **300** in the **Width** and **Height** boxes. Leave the **Anchor** as the middle setting. Click **Add effect**.

Scale and crop will maintain the aspect-ratio of the original image, then crop the larger dimension.

**Width \***  
300 pixels

**Height \***  
300 pixels

**Anchor**

|                       |                                  |                       |
|-----------------------|----------------------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/>            | <input type="radio"/> |
| <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/>            | <input type="radio"/> |

The part of the image that will be retained during the crop.

**Add effect** **Cancel**

6. Click Save.



You edited the image style **User picture**.

## Assign User picture to the picture option in Account settings

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Configuration > People > Account settings > Manage display, then click Compact tab.

**Manage display** ☆

Settings Manage fields Manage form display **Manage display** Translate account settings

Default **Compact**

Home > Administration > Configuration > People > Account settings > Manage display

+ Add group

| FIELD   | LABEL      | FORMAT |   | Show row weights |
|---------|------------|--------|---|------------------|
| Picture | - Hidden - | Image  | Image style: Thumbnail (100x100)<br>Linked to content | ⚙️               |

3. On the **Picture** row, click ⚙️.
4. Select **User picture** from the **Image style** drop-down.
5. Click Update and Save.

You set the **User picture** image style for all accounts.

## What you did in this tutorial

You changed your account settings so that if the user displays a picture it is within the limits of the image style **User picture**.

To check the picture used by a user, navigate to Manage > People, click a user, and see their picture on their profile page. Right-click the image, then click **View image info**, you can see that the dimensions are scaled.

|                  |   |
|------------------|---|
| Type:            | JPEG Image                              |
| Size:            | 29.95 KB (30,667 bytes)                 |
| Dimensions:      | 620px × 413px (scaled to 240px × 160px) |
| Associated Text: | Profile picture for user apicdemo1      |

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tutorial: Adding a Back to top button

You might want to add a Back to top button, as a navigation enhancement to help your Developer Portal users get back to the top of the page that they are viewing, instead of scrolling.

### Before you begin

---

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial.

### About this tutorial

---

In this tutorial, you add a Back to top button to your Developer Portal.

Before you begin, ensure that the **Back to top** module is installed and enabled:

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend**.
4. In the text box, enter `back to top` in the **Block description**.
5. Check the **Back To Top** box, click **Enable**.

### Configure the Back to top module

---

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Configuration > User interface > Back to Top**.

**Back To Top** ☆

Home > Administration > Configuration > User interface

Prevent on mobile and touch devices  
Do you want to prevent Back To Top on touch devices?

Prevent on administration pages and node edit  
Do you want to prevent Back To Top on admin pages?

Prevent on front page  
Do you want to prevent Back To Top on front page?

**Trigger**  
  
Set the number of pixel which trigger the Back To Top button default: 100

**Placement**  
  
Where should the Back To Top button appear?

**Button text**  
  
Set the text of the Back To Top button

3. Change any settings as required, or keep the defaults.
4. Click **Save configuration**

You now have a Back to Top button.

Anonymous (not verified)

Published

08/02/2019 – 07:53

Edit ▾

1 2 Next > Last >

Back to top

### What you did in this tutorial

---

You added a Back to Top button, as a navigation enhancement to help your users get back to the top of a page that they are viewing, instead of scrolling.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tutorial: Adding a zoom effect

You can configure an **Image to Intense** option on your Developer Portal to give a zoom effect to your images.

Adding a zoom effect to the images on your Developer Portal can enhance the user experience.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

In this task, you configure the image field of Article content types to use the zoom effect. Then, create an Article that uses an image, and test the zoom effect.


Before you begin, ensure that the **Intense images** module is installed and enabled:

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend**.
4. In the text box, enter `Intense images` in the **Block description**.  
Note: If the **Intense images** module does not appear in the list see, [Installing custom modules](#).
5. Check the **Intense images** box, click **Enable**.

This tutorial takes you through the following steps:

1. [Configure the image field of Article content types to use the zoom effect.](#)
2. [Create an Article with an image.](#)
3. [Test the zoom effect.](#)

### Configure the image field of Article content types to use the zoom effect

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Structure > Content types > Article > Manage** display.
4. In **Image field Format**, select **Blazy** from the drop-down.
5. Click the edit icon .
6. In **Media Switcher** select **Image to Intense**, then click **Update**.

[+ Add group](#)

| FIELD | LABEL      | FORMAT |
|-------|------------|--------|
| Image | - Hidden - |        |

Format settings: **Blazy**

|                       |                         |                    |                          |
|-----------------------|-------------------------|--------------------|--------------------------|
| Image style           | Large (480x480)         | Responsive image   | - None -                 |
| <b>MEDIA SWITCHER</b> |                         |                    |                          |
| Media switcher        | <b>Image to Intense</b> | Use CSS background | <input type="checkbox"/> |
| Lightbox image style  | - None -                | Thumbnail style    | - None -                 |
| Aspect ratio          | - None -                | Lightbox caption   | - None -                 |

7. Click **Save**.

You changed the **Image field Format** for Article content types to use the zoom effect.

### Create an Article with an image

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Content > Add content > Article**.
3. Enter a title for your article, upload an image, and provide **Alternative text** for the image. Click **Save**.

You created an article and added an image.

### Test the zoom effect



1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Click **Content**.
3. Click the Article that you created.
4. Hover your cursor over the article's image, the cursor turns into a cross, and if you click the image it enlarges.

You tested the zoom effect on the article that you created.

## What you did in this tutorial

You successfully configured the image field of Article content types to use the zoom effect. Then, created an Article that uses an image, and tested the zoom effect.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Creating a drop-down menu link

You can create and use drop-down menus, or nested menus, to enhance the use of your Developer Portal .

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

In this tutorial, you create a drop-down menu link on the **Main navigation** menu.

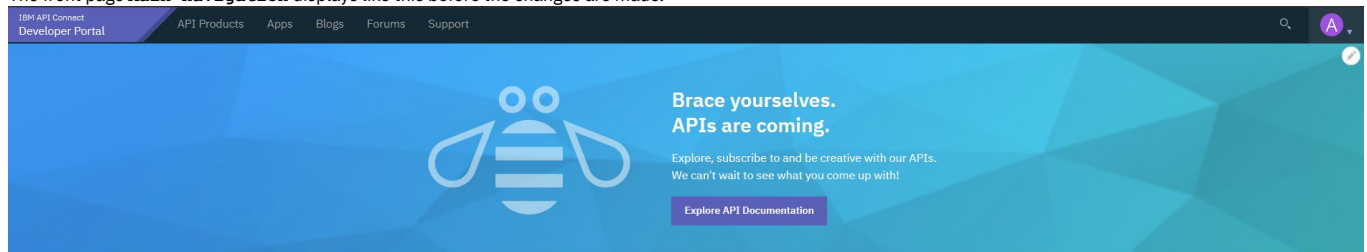
Note:

Although drop-down menu links can be created in the Developer portal, the drop-down menu doesn't expand by default. To make the drop-down menu expand, you need to use the **Superfish** module.

This tutorial takes you through the following steps:

1. [Add menu links to the main navigation menu.](#)
2. [Make the drop-down menu expand by using the Superfish module.](#)

The front page **Main navigation** displays like this before the changes are made.



### Add menu links to the main navigation menu

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Structure > Menus > Main navigation > Add link**
4. Enter **API's** in the **Menu link title**. Enter **/api** in the **Link**. Check **Show as expanded**, then click **Save**.  
Note: Ensure **Show as expanded** is checked as this is the parent link for the submenu links.
5. Add another menu link to the main navigation menu. Enter **API One** in the **Menu link title**. Enter **/product/10/api/5** in the **Link**. Select -- **API's** in the **Parent link**, then click **Save**.

## Add menu link ☆

[Home](#) » [Administration](#) » [Structure](#) » [Menus](#) » [Main navigation](#)

### Menu link title \*

The text to be used for this link in the menu.

### Link \*

- The location this menu link points to.
- Start typing the title of a piece of content to select it. You can also enter an internal path such as `/node/add` or an external URL such as `http://example.com`. Enter `<from`

 Enabled

A flag for whether the link should be enabled in menus or hidden.

### Description

Shown when hovering over the menu link.

 Show as expanded

If selected and this menu link has children, the menu will always appear expanded. This option may be overridden for the entire menu tree when placing a menu block.

### Parent link

The maximum depth for a link and all its children is fixed. Some menu links may not be available as parents if selecting them would exceed this limit.

### Weight

Link weight among links in the same menu at the same depth. In the menu, the links with high weight will sink and links with a low weight will be positioned nearer the top.

6. Add another menu link to the main navigation menu. Enter `API Two` in the **Menu link title**. Enter `/product/10/api/9` in the **Link**. Select `-- API's` in the **Parent link**. Select `1` in **Weight**, then click **Save**.

## Add menu link ☆

[Home](#) » [Administration](#) » [Structure](#) » [Menus](#) » [Main navigation](#)

### Menu link title \*

The text to be used for this link in the menu.

### Link \*

- The location this menu link points to.
- Start typing the title of a piece of content to select it. You can also enter an internal path such as `/node/add` or an external URL such as `http://example.com`. Enter `<from`

 Enabled

A flag for whether the link should be enabled in menus or hidden.

### Description

Shown when hovering over the menu link.

 Show as expanded

If selected and this menu link has children, the menu will always appear expanded. This option may be overridden for the entire menu tree when placing a menu block.

### Parent link

The maximum depth for a link and all its children is fixed. Some menu links may not be available as parents if selecting them would exceed this limit.

### Weight

Link weight among links in the same menu at the same depth. In the menu, the links with high weight will sink and links with a low weight will be positioned nearer the top.

7. Navigate to Structure > Menus > Main navigation. The hierarchy of the menu links now looks like this:

**+ Add link**

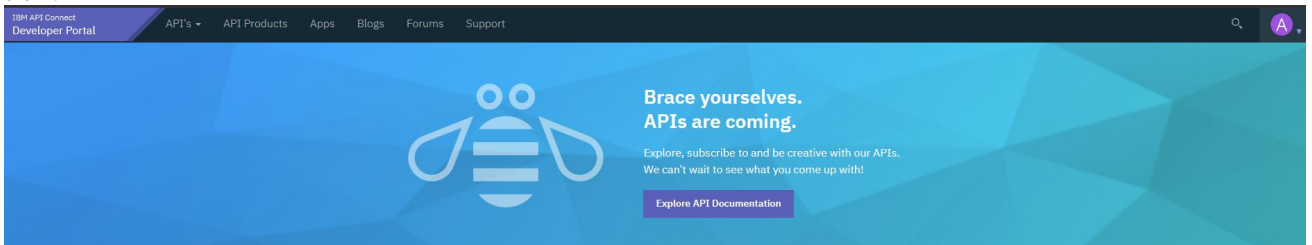
**Title \***  
Main navigation Machine name: main

**Administrative summary**  
Site section links

**Menu language**  
English

| MENU LINK    | ENABLED                             | OPERATIONS |
|--------------|-------------------------------------|------------|
| Home         | <input checked="" type="checkbox"/> | Edit       |
| API's        | <input checked="" type="checkbox"/> | Edit       |
| API One      | <input checked="" type="checkbox"/> | Edit       |
| API Two      | <input checked="" type="checkbox"/> | Edit       |
| API Products | <input checked="" type="checkbox"/> | Edit       |
| Apps         | <input checked="" type="checkbox"/> | Edit       |

8. Navigate to the home page. The **APIs** menu link that you added into the main navigation menu does appear but the submenus links don't expand when you hover over it.



## Make the drop-down menu expand by using the Superfish module

To make the drop-down menu expand when you hover over it, you need to use the **Superfish** module and configure the **Block Layout**.

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Extend. Search for **superfish**, check the **Superfish** box, click Enable.

**+ Install new module**

superfish

Enter a part of the module name or description

Below is a list of installed modules. Modules that are i

**USER INTERFACE**

**Superfish**

**Enable**

3. Navigate to Structure > Block layout.

4. Click Configure, and select Disable for the **Main menu** block.

| BLOCK                                     | CATEGORY | REGION           | OPERATIONS   |
|---|----------|------------------|--|
| Navigation <span>Place block</span>       |          |                  |  |
| Site branding                             | System   | Navigation       | Configure  |
| Main menu                                 | Menus    | Navigation       | Configure<br>Disable<br>Translate<br>Remove<br>Configure |
| Navigation Right <span>Place block</span> |          |                  |  |
| User menu                                 | Menus    | Navigation Right |  |

5. Click Place block for Navigation, Then search for `main navigation` in the box. The filtered results might show two or more options. Click Place block for the **Main navigation** with the Category **Superfish**.

**Place block**

[+ Add custom block](#)

main navigation

| BLOCK           | CATEGORY  | OPERATIONS  |
|-----------------|-----------|-------------|
| Main navigation | Menus     | Place block |
| Main navigation | Superfish | Place block |

6. Deselect **Display title**, click Save block.

**Configure block**

**Block description:** Main navigation

**Title \***

Main navigation Machine name: mainnavigation [Edit]

This field supports tokens. [Browse available tokens.](#)

Display title

[▶ MENU LEVELS](#)

**▼ BLOCK SETTINGS**

**Menu type**

Horizontal (single row)

Horizontal (double row)

Vertical (stack)

*(Default: Horizontal (single row))*

**Style**

None

*(Default: None)*

Add arrows to parent menus

Drop shadows

**Slide-in effect**

Vertical

*(Default: Vertical)*

The plugin provides a handful number of animation effects, they can be used by uploading the

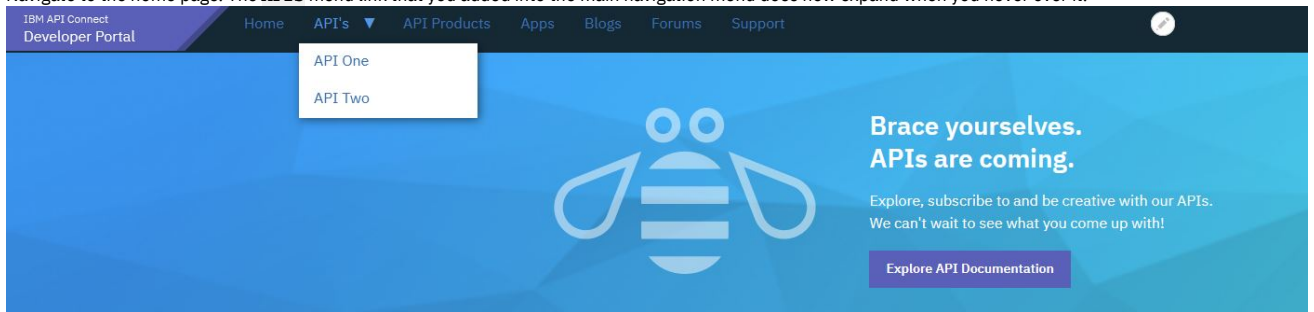
[Save block](#)

7. Drag **Main navigation** to be under **Site branding**.

| BLOCK                               | CATEGORY  | REGION       | OPERATIONS  |
|-------------------------------------|-----------|--------------|-------------|
| Navigation <span>Place block</span> |           |              |             |
| + Site branding                     | System    | Navigation ▾ | Configure ▾ |
| + Main navigation                   | Superfish | Navigation ▾ | Configure ▾ |
| + Main menu (disabled)              | Menus     | Navigation ▾ | Enable ▾    |

8. Click Save blocks.

9. Navigate to the home page. The **APIs** menu link that you added into the main navigation menu does now expand when you hover over it.



## What you did in this tutorial

You created a drop-down menu link on the **Main navigation** menu.

## What to do next

You can consider any other uses of the drop-down menu to enhance your users experience on your Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Displaying blog posts on the front page

You can customize the appearance of the front page of your Developer Portal to display recent blog posts. You might use the blog posts to provide your users with information, such as service availability or marketing campaigns.

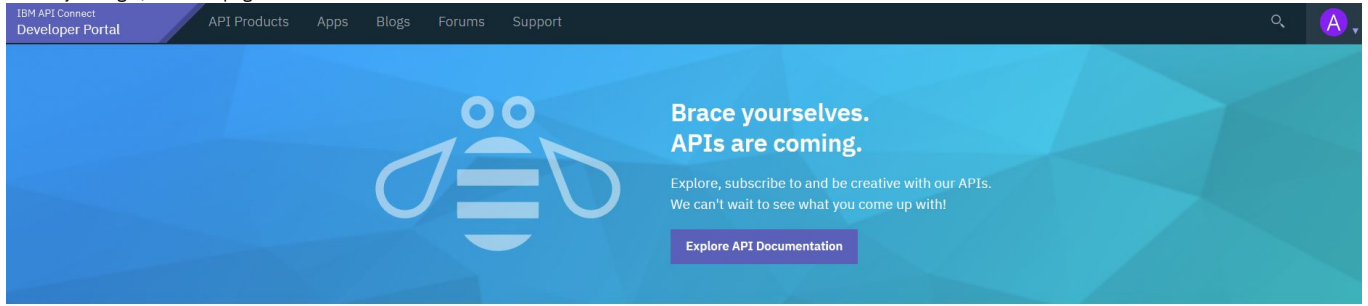
## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

## About this tutorial

In this task, you create a view, customize that view to show blog posts, then embed that view on the front page of your Developer Portal site.

Before you begin, the front page looks like this:



### Sign up

[Create a new account](#) and get started with our APIs. It's free to join.



### Explore our APIs

Take a look at our [API products](#) and quickly find APIs to construct a fully featured application



### Create

Subscribe to a plan and create your application to make use of our APIs.

This tutorial takes you through the following steps:

1. [Create a view.](#)
2. [Customize the view.](#)
3. [Embed the view on the front page of your Developer Portal.](#)

## Create a view

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Structure > Views > Add view.
4. In the **View name** block, enter a name for your view, for example Recent Blog posts.
5. Under **VIEW SETTINGS**, show **Content** of type **Blog Post** sorted by **Newest first**.
6. Check **Create a block** in **BLOCK SETTINGS**, and choose the number of blog posts that you want to be displayed.

**VIEW BASIC INFORMATION**

**View name \***  
Recent Blog posts Machine name: recent\_blog\_posts [Edit]

Description

**VIEW SETTINGS**

Show: **Content** of type: **Blog post** sorted by: **Newest first**

**PAGE SETTINGS**

Create a page

**BLOCK SETTINGS**

Create a block

Block title  
Recent Blog posts

**BLOCK DISPLAY SETTINGS**

Display format: **Unformatted list** of: **titles (linked)**

Items per block  
3

Use a pager

[Save and edit](#) [Cancel](#)

7. Click Save and edit.

## Customize the view

You are now on the page where you can edit your view.

1. If you are not on the edit view page, navigate to Structure > Views, find the name of the view that you created, and click Edit.
2. Click Unformatted list in the **Format** section.
3. Select **Bootstrap List Group**, then click Apply.
4. Select Content: Title in the **Title field** drop-down, then click Apply.
5. Click Save.

## Embed the view on the front page of your Developer Portal

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Structure > Pages.
3. Click **Edit** on the welcome page.
4. Click **Panels**, then **Content**.
5. Click +Add new block.
6. Select your new block, which is under **LISTS (VIEWS)**.



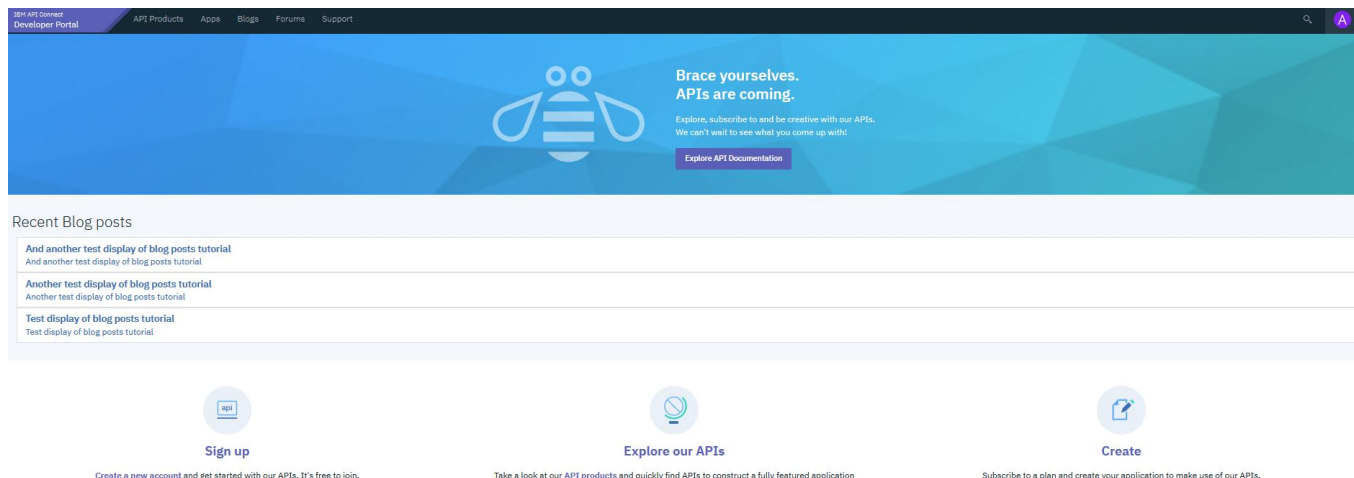
7. Select the number of **Items per block**, then click Add block.
8. You can now rearrange your view to be where you would like it on the front page.

| LABEL          | PLUGIN ID         |  |
|----------------|-------------------|--|
| <b>Content</b> |                   |  |
| +              | Welcome Banner    | block_content:2fcd632-d4b9-44d9-b195-359e751966a1  |
| +              | Featured Content  | featuredcontent                                    |
| +              | Recent Blog posts | views_block:recent_blog_posts-block_1              |
| +              | Getting Started   | block_content:f2b7762b-97f4-4d51-b7dc-d6de5be6d0ce |

9. Click Update and save.

## What you did in this tutorial

You can check that the view you created appears on the front page of your site page.



## What to do next

You can edit your view at any time by navigating to Structure > Views and selecting your view by its name. You can also modify or delete the view's layout on the front page by navigating back to the layout section of your page under Structure > Pages.



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Displaying tweets on the front page

You can customize the appearance of the front page of your Developer Portal to display recent tweets. You might use the tweets to provide your users with information, such as service availability or marketing campaigns.

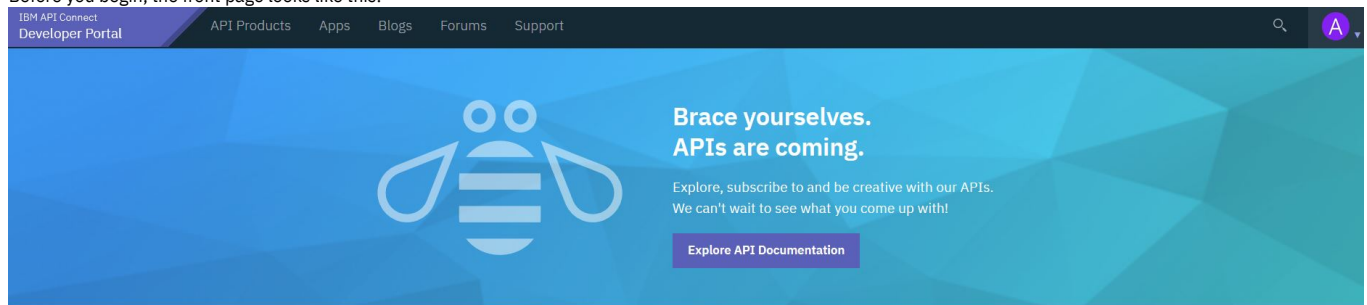
### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

In this task, you create a view, customize that view to show tweets, then embed that view on the front page of your Developer Portal site.

Before you begin, the front page looks like this:



#### Sign up

[Create a new account](#) and get started with our APIs.  
It's free to join.



#### Explore our APIs

Take a look at our [API products](#) and quickly find APIs to construct a fully featured application



#### Create

Subscribe to a plan and create your application to make use of our APIs.

This tutorial takes you through the following steps:

1. [Enable the Media entity twitter module.](#)
2. [Create an entity called Twitter to store the tweets.](#)
3. [Add media items.](#)
4. [Create a carousel view.](#)
5. [Customize the carousel view.](#)
6. [Embed the carousel view on the front page of your Developer Portal.](#)

## Enable the Media entity twitter module

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Click Extend, enter `media` in the search bar.
4. Check Media Entity Twitter, under **MEDIA**.

#### ▼ MEDIA

|                                     |                                |  |
|-------------------------------------|--------------------------------|--|
| <input checked="" type="checkbox"/> | <b>Enhanced Entity Browser</b> | ► Provides some behavior and style enhancements to Entity Browsers, specifically for multiselect and image/media browsers. |
| <input checked="" type="checkbox"/> | <b>Media Entity Twitter</b>    | ► Media Entity Twitter provider.   |

5. Click Enable.

## Create an entity called Twitter to store the tweets

1. Navigate to Structure > Media types > Add media type.
2. In the **Name** field, enter `Twitter`.



### Media source \*

Twitter

Media source that is responsible for additional logic related to this media type.

3. In the **Media source** dropdown, select **Twitter**.
4. Ensure that **Whether to use Twitter api to fetch tweets or not** value is set to No.
5. Ensure the values in the **FIELD MAPPING** section are set to - Skip field -.

**Name \***  
Twitter Machine name: twitter  
The human-readable name of this media type.

**Description**

Describe this media type. The text will be displayed on the *Add new media* page.

**Media source \***  
Twitter  
*The media source cannot be changed after the media type is created.*

**MEDIA SOURCE CONFIGURATION**  
*Tweet URL* field is used to store the essential information about the media item.

**Whether to use Twitter api to fetch tweets or not.**  
No  
In order to use Twitter's api you have to create a developer account and an application. For more information consult the readme file.

Generate thumbnails  
If checked, Drupal will automatically generate thumbnails for tweets that do not reference any external media. In certain circumstances

**FIELD MAPPING**  
Media sources can provide metadata fields such as title, caption, size information, credits, etc. Media can automatically save this information

**Tweet ID**  
- Skip field -

**Twitter user information**  
- Skip field -

6. Click Save.
7. Click Manage display from the Edit dropdown for the new `Twitter` media type.
8. Drag the **Author**, **Author**, and **Thumbnail** fields to the **Disabled** section, and set the **FORMAT** of the **Tweet URL** field to **Twitter embed**.

| FIELD                   | LABEL      | FORMAT        |
|-------------------------|------------|---------------|
| ✚ Tweet URL             | Above      | Twitter embed |
| <b>Disabled</b>         |            |               |
| ✚ Language              | Above      | Language      |
| ✚ Name                  | Above      | Smart trimmed |
| ✚ Search result excerpt |            |               |
| ✚ Authored on           | - Hidden - | Default       |
| ✚ Authored by           | - Hidden - | Author        |
| ✚ Thumbnail             | - Hidden - | Image         |

▶ CUSTOM DISPLAY SETTINGS

Save

9. Click Save.

## Add media items

1. Navigate to Content > Media > Add media > Twitter.
2. In the **Name** field, enter a name for the media, for example `Drupal Tweet`.

3. In the **Tweet URL** field, enter a URL, for example, <https://twitter.com/drupal/status/966443708160839684>.

**Name \***

**Tweet URL \***

|  |   |
|--|---|
| <b>Revision information</b><br>No revision                 | <b>Revision log message</b><br><br>Briefly describe the changes you |
| <b>URL alias</b><br>No alias                               |   |
| <b>Authoring information</b><br>By admin (1) on 2020-01-08 |   |

Published

**Save**

4. Click Save.
5. Repeat these steps to add more tweets, you need at least 2 to make a carousel view.

## Create a carousel view

1. Navigate to Structure > Views > Add view.
2. In the **view name** block, enter a name for your view, for example `Tweet view`.
3. Under **VIEW SETTINGS**, show **Media** of type **Twitter** sorted by **Unsorted**.
4. In the **BLOCK SETTINGS**, section check **Create a block** and ensure it is **Display format Slick Carousel** of **fields**.

**VIEW BASIC INFORMATION**

**View name \***

 Machine name: tweet\_view [Edit]

Description

**VIEW SETTINGS**

Show:  of type:  sorted by:

**PAGE SETTINGS**

Create a page

**BLOCK SETTINGS**

Create a block

Block title

**BLOCK DISPLAY SETTINGS**

Display format:  of:

5. Click Save and edit.

## Customize the carousel view

You are now on the page where you can edit your view.

1. If you are not the edit view page, navigate to Structure > Views, find the name of the view that you created, and click Edit.
2. Click **Tweet view** and change it to the value **<none>**. Then, click Apply.

**Block: The title of this view**

**Title**

This title will be displayed with the view, wherever titles are normally displayed; i.e., as the page title, block title, etc.

**Apply** **Cancel**

3. Click Add in the **FIELDS** section, then search for **tweet**, select the checkbox for **Tweet URL**. Then, click Add and configure fields.

|                                     |                  |       |                      |
|-------------------------------------|------------------|-------|----------------------|
| <input checked="" type="checkbox"/> | <b>Tweet URL</b> | Media | Appears in: twitter. |
|-------------------------------------|------------------|-------|----------------------|

Selected: Tweet URL

**Add and configure fields** **Cancel**

4. From the **Formatter** dropdown select **Twitter embed**, then click Apply.

**Configure field: Media: Tweet URL**

Appears in: twitter.

Create a label

Exclude from display  
Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

**Formatter**

Twitter embed ▼

- ▶ GLOBAL REPLACEMENT PATTERNS (FOR DESCRIPTION FIELD ONLY)
- ▶ STYLE SETTINGS
- ▶ REWRITE RESULTS
- ▶ NO RESULTS BEHAVIOR
- ▶ ADMINISTRATIVE TITLE

**Apply** **Cancel** [Remove](#)

5. In the **FIELDS** section, click **Media: Name**, then, click Remove.

### Configure field: Media: Name

Create a label

Exclude from display  
Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

**Formatter**

Plain text

Link to the Media

▶ GLOBAL REPLACEMENT PATTERNS (FOR DESCRIPTION FIELD ONLY)

▶ STYLE SETTINGS

▶ REWRITE RESULTS

▶ NO RESULTS BEHAVIOR

▶ ADMINISTRATIVE TITLE

[Apply](#) [Cancel](#) [Remove](#)

6. In the **FILTER CRITERIA** section, click the **Media: Published (=**

### Configure filter criterion: Media: Published

Expose this filter to visitors, to allow them to change it

**Operator**

Is equal to

Is not equal to

**Published**

True

False

▶ GLOBAL REPLACEMENT PATTERNS (FOR DESCRIPTION FIELD ONLY)

▶ ADMINISTRATIVE TITLE

[Apply](#) [Cancel](#) [Remove](#)

**True**), then click Remove.

7. In the **FORMAT** section, click **Settings** next to **Slick Carousel**.

8. Set **Skin Main** to **Default**.

9. In the **CAPTION FIELDS** section, check **Media: Tweet URL**, and **Override main optionset**.

10. In the **OVERRIDEABLE OPTIONS** section, select **Autoplay** and **Dots** then, click Apply.

| CAPTION FIELDS  |  |   |   |
|---|--|---|---|
| Media: Tweet URL <input checked="" type="checkbox"/>        |  |   |   |
| slick ID # <input type="text"/>                             |  | Cache <input type="text" value="&lt;No caching&gt;"/> |   |
| Override main optionset <input checked="" type="checkbox"/> |  |   |   |
| OVERRIDABLE OPTIONS   |  |   |   |
| Arrows <input type="checkbox"/>                             | Autoplay <input checked="" type="checkbox"/> | Dots <input checked="" type="checkbox"/>              | Draggable <input type="checkbox"/>      |
| Infinite <input type="checkbox"/>                           | Mousewheel <input type="checkbox"/>          | Randomize <input type="checkbox"/>                    | Variable width <input type="checkbox"/> |

**Block: Pager options**

**Items to display**  
  
 Enter 0 for no limit.

**Offset (number of items to skip)**  
  
 For example, set this to 3 and the first 3 items will not be displayed.

11. In the **PAGER** section, click **Items**, set the field to 0 then, click Apply.  
 12. Click Save.

## Embed the view on the front page of your Developer Portal

1. Navigate to Structure > Pages.
2. Click **Edit** on the welcome page.
3. Click **Panel**s, then **Content**.
4. Click +Add new block.

### LISTS (VIEWS)

- Active forum topics
- APIs
- Applications
- Blog
- New forum topics
- Product List
- Recent comments
- Recent content
- Tweet view
- Who's online
- Workbench: Current user: Overview block
- Workbench: Edits by user: Overview block
- Workbench: Recent content: Overview block

5. Select your new block, which is under **LISTS (VIEWS)**.
6. Deselect **Display title** then, click Add block.

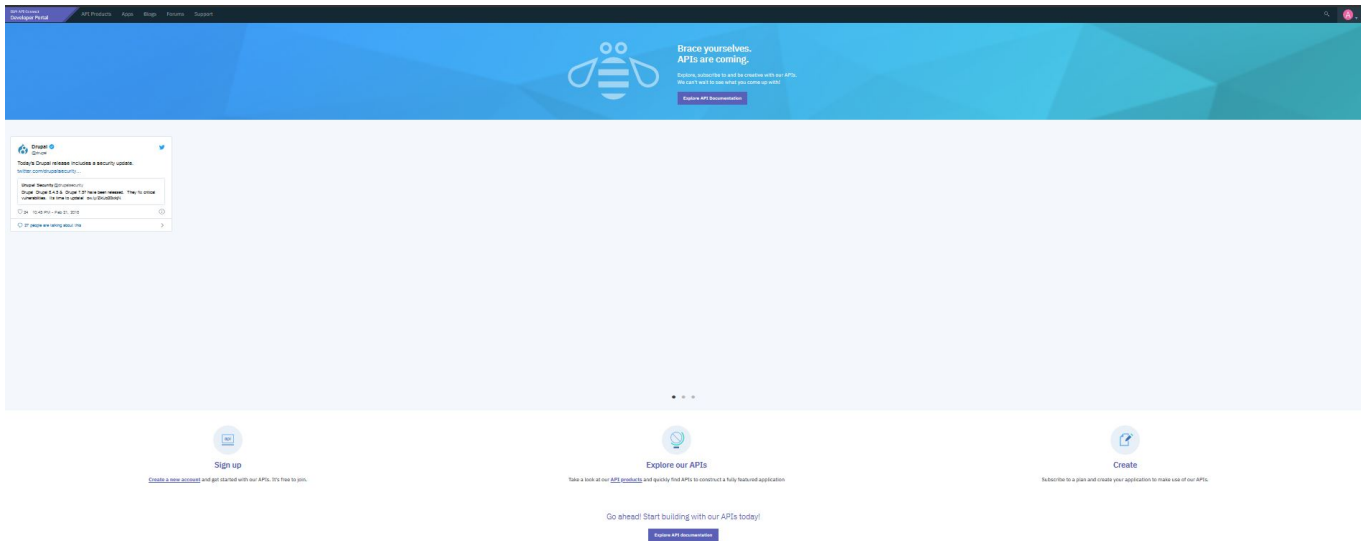
7. You can now rearrange your view to be where you would like it on the front page.

| LABEL            | PLUGIN ID  |
|------------------|--|
| <b>Content</b>   |  |
| Welcome Banner   | block_content:d0f69e7a-e8c3-4f56-8537-13cea477f68a |
| Featured Content | featuredcontent                                    |
| Tweet view...    | views_block:tweet_view-block_1                     |
| Getting Started  | block_content:f56a934e-01ac-4083-8d80-da7169dd0289 |
| Social Block     | social_block                                       |
| Go ahead         | block_content:42d66e02-3835-4300-abc0-b719dcadd4e6 |
| Get help         | block_content:777e45d7-90c4-499e-8727-e08215cae05a |

8. Click Update and save.

## What you did in this tutorial

You can check that the view you created appears on the front page of your site page.



## What to do next

You can edit your view at any time by navigating to Structure > Views and selecting your view by its name. You can also modify or delete the view's layout on the front page by navigating back to the layout section of your page under Structure > Pages.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Changing the front page of your Developer Portal

You can create and arrange content in your Developer Portal to provide a front page that aligns with your brand, and helps your users discover your APIs.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

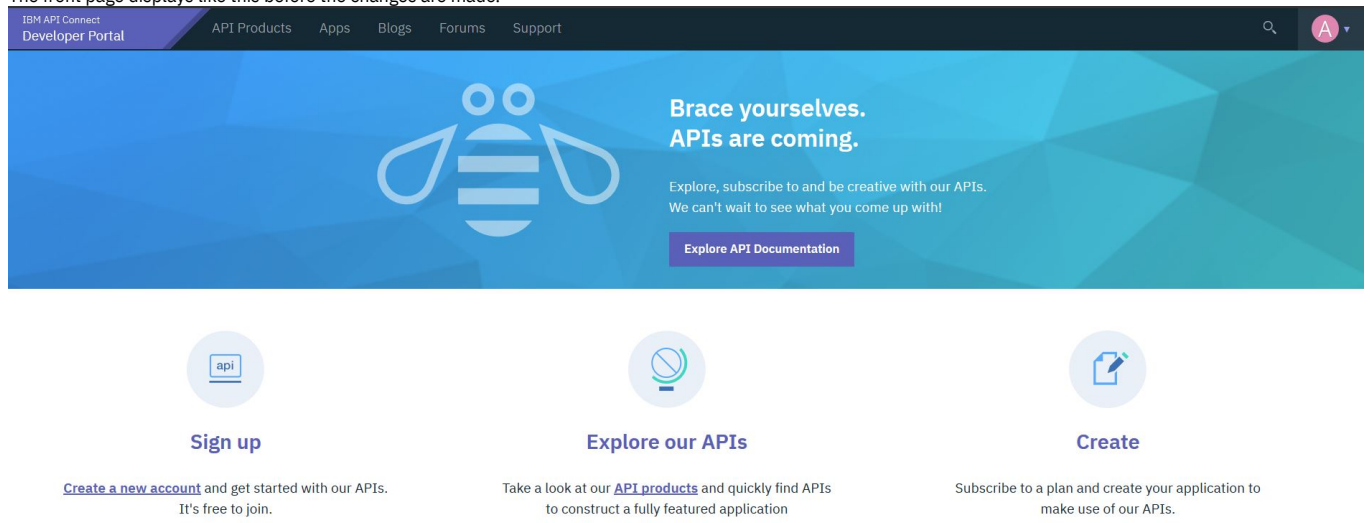
### About this tutorial

In this tutorial, you create some content for a page, and then change the front page to be your new page.

This tutorial takes you through the following steps:

1. [Creating some custom blocks.](#)
2. [Create and configure a page.](#)
3. [Adding the custom blocks to the page.](#)
4. [Changing the front page to be your new page.](#)

The front page displays like this before the changes are made.



## Creating some custom blocks

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Structure > Block layout > Add custom block
4. Enter `Block 1` in the **Block description**.
5. In the body area, click **Source**, then enter this HTML:

```
<div class="tutorial_block_1 tutorial_block_frontpage">
<h1><span>Create with our APIs</span></h1>

<p><span>Welcome to our Developer Portal where you will find APIs to create your awesome app.</span></p>
</div>
```

### Add custom block ☆

[Home](#)

#### Block description \*

Block 1

A brief description of your block.

#### Body

Rich text editor toolbar with options: Bold, Italic, Underline, Strikethrough, Bulleted list, Numbered list, Indent, Outdent, Link, Unlink, Image, Video, Table, Undo, Redo, Format, Source, and Refresh.

```
<div class="tutorial_block_1 tutorial_block_frontpage">
<h1><span>Create with our APIs</span></h1>

<p><span>Welcome to our Developer Portal where you will find APIs to create your awesome app.</span></p>
</div>
```

6. Check what the block looks like by clicking **Source** again.

## Add custom block ☆

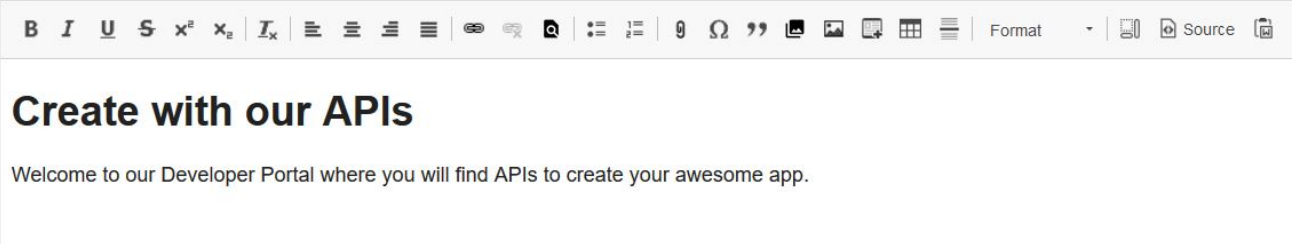
Home

### Block description \*

Block 1

A brief description of your block.

### Body



The screenshot shows a rich text editor interface. At the top, there is a toolbar with various icons for text formatting (bold, italic, underline, strikethrough, text color, background color), list creation, link, unlink, and other functions. Below the toolbar, the main content area contains the following text:

## Create with our APIs

Welcome to our Developer Portal where you will find APIs to create your awesome app.

Note: This block uses almost no inline-styling, instead it allows your theme to decide how it is rendered. The advantage of this set up is that you can easily change the rendering without modifying existing content in the database. Any styling that is required can be added to a custom theme. For more information, see [Tutorial: Creating a custom theme for the Developer Portal](#).

7. Click **Save**.

You have created **Block 1**, repeat steps 3 to 7 to create **Block 2**, **Block 3**, and **Block 4**.

8. Enter **Block 2** in the **Block description**. In the body area, click **Source**, then enter this HTML:

```
<div class="tutorial_block 2 tutorial_block frontpage">
<p class="text-align-center">Explore and subscribe to our APIs.<br />
See what you come up with!</p>

<p class="text-align-center">&nbsp;</p>

<div class="bannerButtonRow text-align-center"><a class="button cta-btns--white_btn"
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_product">Explore API
Documentation</a></div>
</div>
```

9. Enter **Block 3** in the **Block description**. In the body area, click **Source**, then enter this HTML:

```
<div class="tutorial_block 3">
<p class="text-align-center"><a
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_user_register">Create a new
account</a> and get started with our APIs. It's free to join.</p>
</div>
```

10. Enter **Block 4** in the **Block description**. In the body area, click **Source**, then enter this HTML:

```
<div class="block-get-help">
<div class="get_help">
<div class="column col1">
<h3>Getting Help</h3>

<div>Be sure to check out these extra help resources.</div>
</div>

<div class="column col2">
<h4>Get Started</h4>

<div>Get started with our APIs by creating an account and exploring the documentation to find what's right for you.</div>

<div><a href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_user_register"><svg
height="24"
id="_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_tutorial_portal_example_frontpage
_Layer_1" space="preserve" version="1.1" viewBox="0 0 32 32" width="24" x="0px" xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2000/svg" y="0px"> <g> <polygon points="15.293,10.707 19.586,15 8,15 8,17 19.586,17 15.293,21.293
16.707,22.707 23.414,16 16.707,9.293"></polygon> <path d="M16,2C8.269,2,2,8.269,2,16s6.269,14,14,14c7.731,0,14-6.269,14-
14S23.731,2,16,2z M16,28C9.383,28,4,22.617,4,16S9.383,4,16,4c6.617,0,12,5.383,12,12S22.617,28,16,28z"></path> </g>
</svg>Create an Account</a></div>
</div>

<div class="column col3">
<h4>Forum</h4>

<div>Ask a question in the forums or search to forum history to see if it's been asked before.</div>

<div><a href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_forum"><svg
height="24"
id="_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_tutorial_portal_example_frontpage
_Layer_1" space="preserve" version="1.1" viewBox="0 0 32 32" width="24" x="0px" xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2000/svg" y="0px"> <g> <polygon points="15.293,10.707 19.586,15 8,15 8,17 19.586,17 15.293,21.293
16.707,22.707 23.414,16 16.707,9.293"></polygon> <path d="M16,2C8.269,2,2,8.269,2,16s6.269,14,14,14c7.731,0,14-6.269,14-
14S23.731,2,16,2z M16,28C9.383,28,4,22.617,4,16S9.383,4,16,4c6.617,0,12,5.383,12,12S22.617,28,16,28z"></path> </g>
</svg>Join the discussion</a></div>
</div>

<div class="column col4">
<h4>Contact Us</h4>
```



<div>Can't find the answer to your question? Need more help? Have some feedback? Let us know!</div>

```
<div><a href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_contact"><svg
height="24"
id="_home_markdown_jenkins_workspace_Transform_in_SSMNED_2018_com.ibm.apic.devportal.doc_tutorial_portal_example_frontpage
_Layer_1" space="preserve" version="1.1" viewBox="0 0 32 32" width="24" x="0px" xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2000/svg" y="0px"> <g> <polygon points="15.293,10.707 19.586,15 8,15 8,17 19.586,17 15.293,21.293
16.707,22.707 23.414,16 16.707,9.293"></polygon> <path d="M16,2C8.269,2,2,8.269,2,16s6.269,14,14,14c7.731,0,14-6.269,14-
14S23.731,2,16,2z M16,28C9.383,28,4,22.617,4,16S9.383,4,16,4c6.617,0,12,5.383,12,12S22.617,28,16,28z"></path> </g>
</svg>Email us</a></div>
</div>
</div>
```

11. Navigate to Structure > Block layout > Custom block library to see the new blocks.

Custom block library ☆

Block layout Custom block library

Blocks Block types

Home » Administration » Structure » Block layout

Blocks in the block library belong to Custom block types, each with its own fields and display settings. After creating

+ Add custom block

Block description Block type

- Any -

Apply

BLOCK DESCRIPTION	BLOCK TYPE
Block 4	Basic block
Block 3	Basic block
Block 2	Basic block
Block 1	Basic block

## Create and configure a page

1. If the administrator dashboard isn't displayed, click Manage to display it.
2. Navigate to Structure > Pages, click + Add page.

+ Add page

LABEL	MACHINE NAME	PATH	OPERATIONS
Node view	node_view	/node/{node}	Edit ▾
welcome	welcome	/home	Edit ▾

3. Enter Portal home in the Administrative title field.
4. Enter newhome in the Path field.
5. Select Panels in the Variant type drop-down.
6. Leave other fields clear.
7. Click Next.
8. Accept the defaults of Panels for Label, and Standard for Builder, click Next.

## Configure variant ☆

[Home](#) » [Administration](#) » [Structure](#) » [Pages](#) » [Add new page](#)

Page information » **Configure variant** » [Layout](#) » [Content](#)

**Label \***

Panels

**Builder**

Standard

Previous

Next

9. Select **Two column bricks** from **Layout** drop-down. Click Next.

10. Keep all settings except for the following:

- **Top Left**, set Classes to **Small:9**
- **Top Right**, set Classes to **Small:3**
- **Bottom Left**, set Classes to **Small:4**
- **Bottom Right**, set Classes to **Small:8**

▼ **REGION: TOP LEFT**

**Wrapper**

Div

**Classes**

Small: 6  
Small: 7  
Small: 8  
Small: 9

Add region specific classes: `bs-region` and `bs-region--top_left`

**Additional attributes**

E.g. `id|custom-id,role|navigation,data-something|some value`

[Browse available tokens.](#)

11. Click Next.

## Adding the custom blocks to the page

1. Leaving **Page title** blank, click + **Add new block**.
2. Select **Block 1** from the Custom block list.



3. Clear **Display title**, select **Top Left** from the Region drop down, then click Add block.

You added **Block 1**, repeat steps 1 to 3 to add **Block 2**, **Block 3**, and **Block 4**.

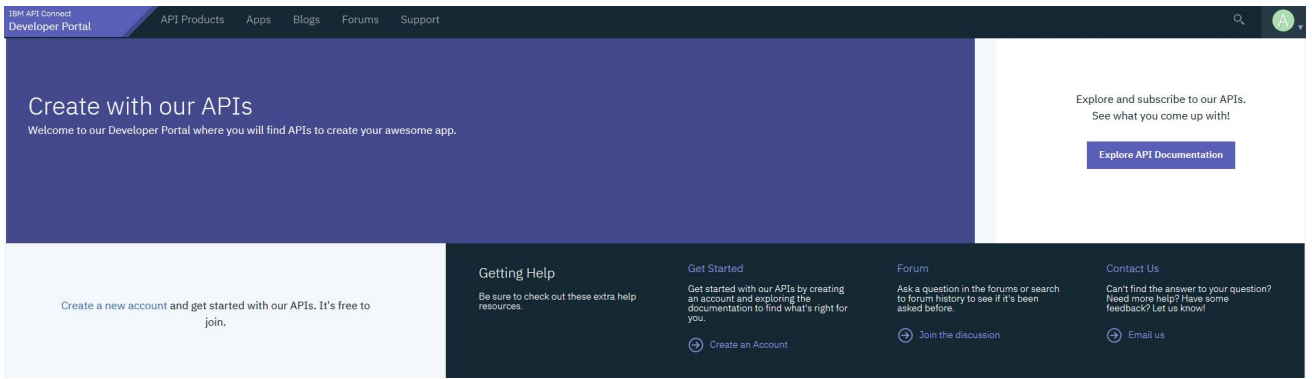
4. For **Block 2** select region **Top Right**.
5. For **Block 3** select region **Bottom left**.
6. For **Block 4** select region **Bottom Right**.
7. Click **Finish**.
8. Click Update and save.

## Changing the front page to be your new page

---

Note: In this tutorial, you create a new page and swap it with the default. If errors are made, the default page is still available as a backup.

1. Navigate to **Structure > Pages**.
2. Click the **newhome** URL link for your new **Portal home** page, check that your page looks as designed.



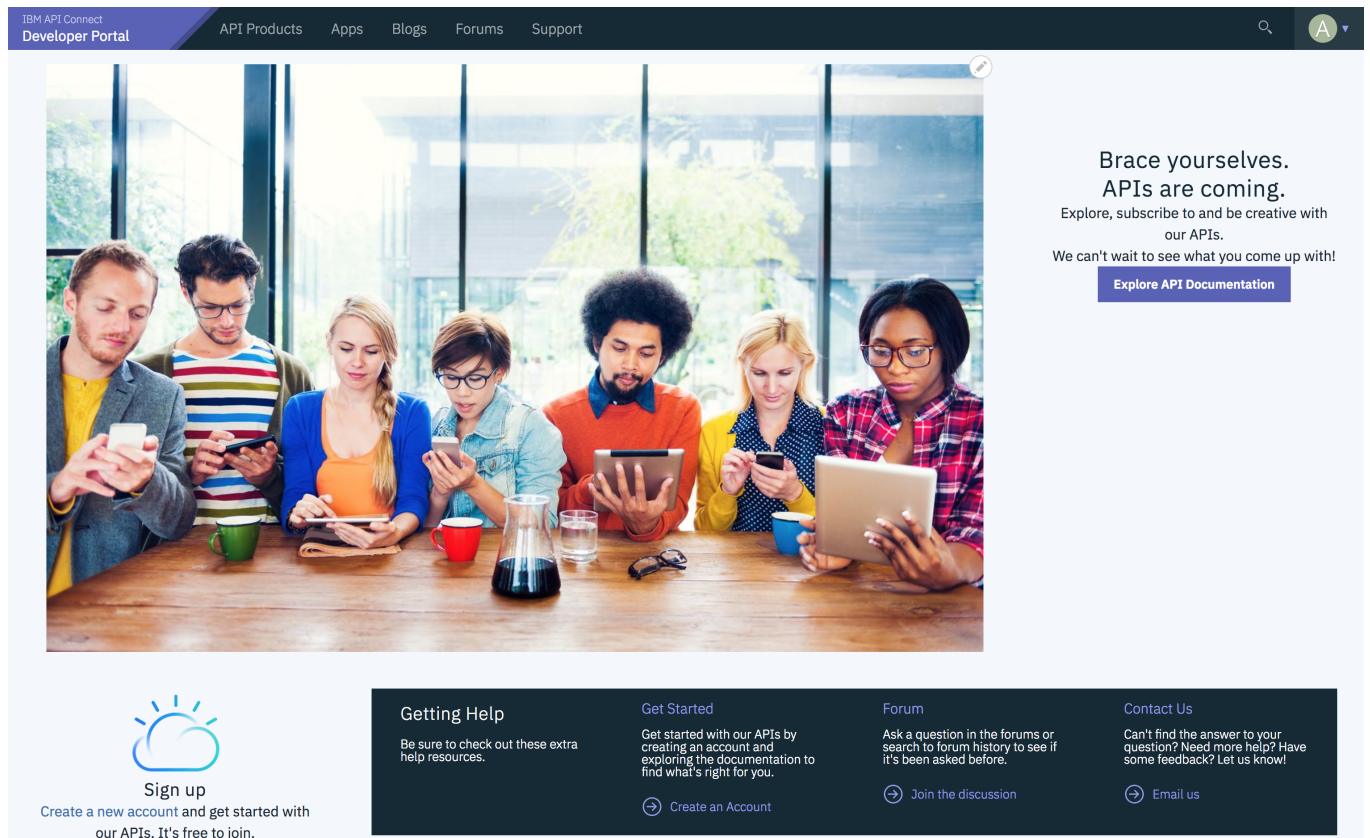
3. Navigate to Configuration > System > Basic site settings
4. In **Front Page** region set Default front page field to **newhome**.
5. Click Save configuration.

## What you did in this tutorial

The front page of your Developer Portal is now changed and laid out with the blocks of content that you specified.

## What to do next

You can add further branding by adding content to your custom blocks such as images, or try to use one of the provided custom blocks such as the Featured Content Block. By default this block is English, you might want to consider adding versions in other languages.



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Grouping products by category

You can configure the Developer Portal so that products are displayed in groups according to the category that they are tagged with.

You can use the **Views** module to fetch content from your Developer Portal site, such as lists of products, and present the content to users in different formats. You can combine multiple views on a new page to display whatever content you want.

## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

## About this tutorial

In this tutorial you create three views in the Developer Portal, each one is configured to display a list of products according to their category. You then create a new page to show the views, where you see products that are grouped by category.

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Create three new taxonomy tags.
  - a. Navigate to **Structure > Taxonomy > Tags**.
  - b. Click **Add term**.
  - c. Enter **Accounts** in the **Name** field and click **Save**.
  - d. Enter **Logistics** in the **Name** field and click **Save**.
  - e. Enter **Marketing** in the **Name** field and click **Save**.
  - f. Navigate to **Structure > Taxonomy > Tags** to show the list of the new tags.

**Tags** ☆

List Edit Manage fields Manage form display Manage display

Home » Administration » Structure » Taxonomy » Edit Tags

You can reorganize the terms in *Tags* using their drag-and-drop handles, and group terms under a parent term by sliding them under and to the right of the parent.

+ Add term

NAME	OPERATIONS
+ Accounts	Edit ▾
+ Logistics	Edit ▾
+ Marketing	Edit ▾

Save Reset to alphabetical

4. Create a view to display products with the **Accounts** category.
  - a. Navigate to **Structure > Views > Add new view**.
  - b. Enter **Accounts products** in the **View name** field.
  - c. In **View Settings**, select **Show: Content, of type: Product, sorted by: Newest first**, from the drop-down lists.
  - d. In **Page settings** leave the **Create a page** cleared.
  - e. In **Block settings** select **Create a block** and enter **Accounts products** in the **Block title**.
  - f. Ensure **Block Display Settings** has **Display format: Masonry, of: tiles (linked)**, and select **Use a pager**.

**PAGE SETTINGS**

Create a page

**BLOCK SETTINGS**

Create a block

Block title

Accounts products

---

**BLOCK DISPLAY SETTINGS**

Display format: **Masonry** of: **titles (linked)**

---

Items per block

5

Use a pager

**Save and edit** **Cancel**

- g. Click **Save and edit**.
- 5. Configure the view block settings.
  - a. Under the **Format** section click the **Fields** link next to the **Show** label.
  - b. In the dialog box, select the **Content** radio button then click **Apply**.
  - c. Select **Card** from the **View mode** drop-down list, then click **Apply**.
  - d. Click **Add** next to the **Filter Criteria** section.
  - e. In the dialog box, search for **tag** then select the option **Tags (apic\_tags)** from the list then click **Add and configure filter criteria**.
  - f. In the next dialog box, select the **Tags** radio button and the **Client-side hierarchical** then click **Apply and continue**.
  - g. Next set the **Operator** to **Is all of**, and from the **Select terms from vocabulary Tags** drop-down select **Accounts** and click **Apply**.
  - h. Under the **Advanced** options, click the **None** link next to the **CSS class** label.
  - i. If you are using the default API connect theme, then enter the CSS class name **product-view** and click **Apply**. If you have your own css class that you want to use, then enter its name in here instead.
  - j. Click **Save** to save all the changes.

Displays

Block\* + Add Edit view name/description

Display name: **Block** Duplicate Block

<p><b>TITLE</b></p> <p>Title: <b>Accounts products</b></p> <p><b>FORMAT</b></p> <p>Format: <b>Masonry</b>   Settings</p> <p>Show: <b>Content</b>   <b>Card</b></p> <p><b>FIELDS</b></p> <p>The selected style or row format does not use fields.</p> <p><b>FILTER CRITERIA</b> <span>Add</span></p> <p>Content: <b>Published (= Yes)</b></p> <p>Content: <b>Content type (= Product)</b></p> <p>Content: <b>Tags (= Accounts)</b>   Settings</p> <p><b>SORT CRITERIA</b> <span>Add</span></p> <p>Content: <b>Authored on (desc)</b></p>	<p><b>BLOCK SETTINGS</b></p> <p>Block name: <b>None</b></p> <p>Block category: <b>Lists (Views)</b></p> <p>Allow settings: <b>Items per page</b></p> <p>Access: <b>Permission</b>   <b>View published content</b></p> <p><b>HEADER</b> <span>Add</span></p> <p><b>FOOTER</b> <span>Add</span></p> <p><b>NO RESULTS BEHAVIOR</b> <span>Add</span></p> <p><b>PAGER</b></p> <p>Use pager: <b>Full</b>   <b>Paged, 5 items</b></p> <p>More link: <b>No</b></p> <p>Link display: <b>None</b></p> <p><b>LANGUAGE</b></p> <p>Rendering Language: <b>Content language of view row</b></p>	<p><b>ADVANCED</b></p> <p><b>CONTEXTUAL FILTERS</b> <span>Add</span></p> <p><b>RELATIONSHIPS</b> <span>Add</span></p> <p><b>EXPOSED FORM</b></p> <p>Exposed form style: <b>Basic</b>   Settings</p> <p><b>OTHER</b></p> <p>Machine Name: <b>block_1</b></p> <p>Administrative comment: <b>None</b></p> <p>Use AJAX: <b>No</b></p> <p>Hide attachments in summary: <b>No</b></p> <p>Contextual links: <b>Shown</b></p> <p>Use aggregation: <b>No</b></p> <p>Query settings: <b>Settings</b></p> <p>Caching: <b>Tag based</b></p> <p>CSS class: <b>product-view</b></p> <p>Hide block if the view output is empty: <b>No</b></p>
---	---	--

**Save** **Cancel**

- 6. Create a view for the Logistics products.
  - a. Select **Structure > Views**.
  - b. Navigate to the **Accounts products** view and select the **Duplicate** option.
  - c. Give the duplicated view the name **Logistics products** and click **Duplicate**.

- d. Click the Title **Accounts products** and change it to **Logistics products** and click **Apply**.
  - e. Under the **Filter Criteria** section click the **Content: Tags: (=Accounts)** link.
  - f. Change the value in the **Select terms from vocabulary Tags** drop-down to **Logistics** and click **Apply**.
  - g. Click **Save** to save all the changes.
7. Repeat step 6 for Marketing products.
  8. Next, create a page to display the views.
    - a. Select Structure, > Pages.
    - b. Click **Add page**.
    - c. Enter **Products by category** in the **Administrative title**, and **products-by-category** in the **Path**. Select the **Panels** option from the **Variant type** drop-down list then click **Next**.

Page information > Configure variant

**PAGE MANAGER**

**Administrative title \***

Products by category Machine name: products\_by\_category [Edit]

---

**Administrative description**

---

**Path \***

products-by-category

Path to your custom page. Beginning and Ending slashes are automatically removed.

Use admin theme

**Variant type**

Panels

---

**Optional features**

Page access

Variant contexts

Variant selection criteria

Check any optional features you need to be presented with forms for configuring them. If you do not check them here you will still be able to utilize these features once the new page is created. If you are not sure, leave these unchecked.

**Next**

- d. Change the **Label** value to **Products grouped by category** and set the **Builder** to **Standard** then click **Next**.
- e. From the **Layout** drop-down list select the value **Three column** then click **Next**.
- f. Leave all the defaults on the next page and click **Next**.
- g. Click **+Add new block**.
- h. Select **Accounts products** from the **Lists (views)** section.
- i. Ensure that **Display tile** is selected and **Region** is set to **Left**, click **Add block**.

**Add block** ×

---

**Block description**

Accounts products

Display title

**Items per block**

Override title

**Region \***

Left

**Add block**

- j. Click **Add new block**. Select **Logistics products** view from the list and set the **Region** to **Middle**, click **Add block**.
- k. Click **Add new block**. Select **Marketing products** view from the list and set the **Region** to **Right**, click **Add block**.
- l. Enter **Products grouped by category** in the **Page title** field, and click **Finish**.

**Page title**  
  
 Configure the page title that will be used for this display.

[+ Add new block](#)

LABEL	PLUGIN ID	REGION
<b>Left</b>		
Accounts products	views_block:accounts_products-block_1	Left
<b>Middle</b>		
Logistics products	views_block:logistics_products-block_1	Middle
<b>Right</b>		
Marketing products	views_block:marketing_products-block_1	Right

[Previous](#) [Finish](#)

- From your API Manager, create three new products, or update existing ones, to give each one a category value of either Accounts, Logistics, or Marketing. Publish each of the products to the catalog with the configured Developer Portal.
- Check that the **Products grouped by category** page shows your published products by category.
  - Select **Structure > Pages**.
  - Click the **/products-by-category** path link.
  - You are redirected to the **/products-by-category** page and can see the products that you assigned to these categories.

## What you did in this tutorial

You created three new views in the Developer Portal, and configured each one to display a list of products according to their category. You then created a new page to show the views.

### Accounts products

### Logistics products

### Marketing products

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Adding a field to the sign up form

You might want to collect and store additional information about your Developer Portal users. For example, you might want to collect a department name, employee ID, or similar data when a user registers for an account on the portal.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

In this tutorial, you will add a Department code field to the Sign up form. This field is required for all portal users.

This tutorial takes you through the following steps:

- [Adding a field to a user entity.](#)
- [Adding the new field to the Sign up form.](#)
- [Testing the new field.](#)

### Adding a field to a user entity

- Log in to your Developer Portal as an administrator.



2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Configuration** » **People** » **Account settings** » **Manage fields**

You are on the **Manage fields** tab.

1. Click **+ Add field**.
2. In the **Select a field type** dropdown, select **Text (plain)**.
3. In the **Label** field enter, **Department code**.
4. Click **Save and continue**.

You are on the **Field settings** tab.

1. Accept the defaults of **Maximum length** (255) and **Allowed number of values** (Limited:1).
2. Click **Save field settings**.

You are on the **Edit** tab.

1. Add some help text. This is information that is displayed to the user to assist them completing the field on a form. Examples or links to further information are useful here. We are going to add an example in this tutorial.
2. Add **Example department code: DEP123** into the **Help text** field.
3. Check the **Required field**.
4. Click **Save settings**.

The field has now been created. You are back on the **Manage fields** tab.

1. Check that **Department code** is a listed field.

## Adding the new field to the Sign up form

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Configuration** » **People** » **Account settings** » **Manage form display**

You are on the **Manage form display** tab. These tabs, one per form, show the fields that are displayed from the user entity, and in which order.

1. Switch to the **Register** tab. Notice that initially the new **Department code** field is below the **Disabled row** in the page. This means that it isn't shown on the form.
2. Click the **Department code** icon and drag it so that it is between **Last name** and **Consumer organization** in the list.

The screenshot shows the 'Manage form display' interface for the 'Register' form. At the top, there are tabs for 'Settings', 'Manage fields', 'Manage form display', 'Manage display', and 'Translate account settings'. The 'Manage form display' tab is active. Below the tabs, there are two sub-tabs: 'Default' and 'Register'. The 'Register' sub-tab is selected. The breadcrumb path is 'Home » Administration » Configuration » People » Account settings » Manage form display'. There is a '+ Add group' button. A 'Show row weights' link is visible on the right. The main area contains a table with two columns: 'FIELD' and 'WIDGET'. The table lists several fields, including 'Status', 'Username', 'Roles', 'Notify user about new account', 'Current password', 'E-mail address', 'First Name', 'Last Name', 'Department code', 'Consumer organization', 'Password', and 'Language code'. The 'Department code' field is currently in the 'Disabled' row. The 'Consumer organization' field is highlighted in blue. Each field has a widget dropdown menu and a 'Textfield size: 60' label. There are gear icons for settings next to the 'First Name', 'Last Name', 'Department code', and 'Consumer organization' fields.

FIELD	WIDGET	Textfield size: 60	⚙️
+	Status		
+	Username		
+	Roles		
+	Notify user about new account		
+	Current password		
+	E-mail address		
+	First Name	Textfield	⚙️
+	Last Name	Textfield	⚙️
+	Department code	Textfield	⚙️
+	Consumer organization	Textfield	⚙️
+	Password		
<b>Disabled</b>			
+	Language code	Language select	

3. Click **Save**.

## Testing the new field

1. Sign out as the Admin user.
2. Click **Create account**.
3. Observe that the **Department code** field is now part of the sign up form.

## What you did in this tutorial

You added a Department code field to the Sign up form, then tested the form to ensure that the `Department code` field is now part of the sign up form.

In reality, you would need to add this configuration in such a way that it is available for all sites. To achieve this you would include the configuration in your custom module. For more information, see [this Drupal documentation](#).

You might want to add extra validation to the Department code field on the Sign up form. For more information, see [Tutorial: Adding validation to a field on the sign-up form](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Adding validation to a field on the sign-up form

You might want to add extra validation to a field to help your Developer Portal users. For example, you might want to validate a department name, employee ID, or similar data when a user registers for an account on the portal.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. You must have completed the tutorial about [Adding a field to the sign up form](#), where you added a Department code field to the Sign-up form.

### About this tutorial

In this tutorial, you add some validation to the Department code field on the Sign-up form. This is a required field for all portal users and is a string of the format `DEPnnn`, where `DEP` is the string prefix and `nnn` is a three-digit number, for example, `DEP123`. The validation is done by using a custom module that is already available in [IBM APIC Portal - addons](#). However, for more information, see [Custom module development: creating a module skeleton](#).

This tutorial takes you through the following steps:

1. [Cloning and packaging the validation module](#).
2. [Installing and enabling the validation module](#).
3. [Testing the validation in the sign up form](#).

Look at the files we will use in this link [IBM APIC Portal - addons](#).

- `user_field_example.info.yml` - this is the metadata for the module, defining things like the name and version, and containing other important information that is required by Drupal to import and enable the module.

```
name: 'IBM APIC Portal - user field example'
type: module
description: 'IBM API Connect Developer Portal tutorial - Example of validating a custom user field'
package: 'Custom'
core_version_requirement: ^8 || ^9
version: 1.0.0
project: 'user_field_example'
dependencies:
  - ibm_apim
  - auth_apic
```

- `user_field_example.module` - this contains a validation function and adds this to the Sign-up form by using the `hook_form_alter` function.

```
<?php
use Drupal\Core\Form\FormStateInterface;

/**
 * Implementation of hook_form_alter() to alter the Sign up form
 */
function user_field_example_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if ($form_id === 'user_register_form') {
    $form['#validate'][] = 'user_field_example_validate_department_code';
  }
}

/**
 * Validate the Department code field on the Sign up form.
 *
 * Valid entry = DEPnnn
 * where n = single figure digit.
 */
function user_field_example_validate_department_code($form, &$form_state) {
  $dept_code = $form_state->getValue('field_department_code')[0]['value'];
  $valid = preg_match('/^DEP\d{3}$/', $dept_code);
  if (!$valid) {
    $form_state->setErrorByName('field_department_code', t('Invalid department code.'));
  }
}
```

### Cloning and packaging the validation module

Validation is added to the **Department code** field on the Sign-up form by using a custom module.

1. Clone the custom module. Here is an example command if you are using Mac and Linux:

```
git clone https://github.com/ibm-apiconnect/devportal-addons
```

2. Package up the custom module.

```
cd devportal-addons/apic_v10/modules
tar -czf user_field_example.tgz user_field_example/
```

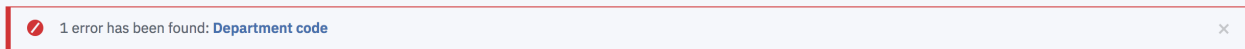
## Installing and enabling the validation module

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend**, **Install new module**.
4. Click **Choose file** from the **Upload a module or theme archive to install** section.
5. Browse to and select **user\_field\_example.tgz**, the packaged custom module that you created earlier. Click **Open**.
6. Click **Install**.
7. Click the **Enable newly added modules** link.
8. Enter **IBM APIC Portal - user field example** in the filter box.
9. Check the checkbox next to **IBM APIC Portal - user field example**.
10. Click **Enable**.

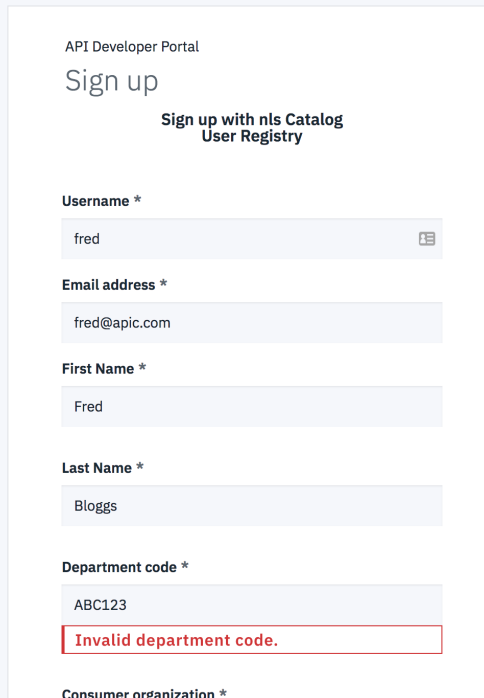
The module is now installed and enabled.

## Testing the validation in the sign-up form

1. Sign out as the Admin user.
2. Click **Create account**.
3. Complete all fields, set **Department code** to **ABC123**.
4. Click **Sign up**.
5. Observe the failure message and highlighting on the **Department code** field.



1 error has been found: **Department code**



API Developer Portal

### Sign up

Sign up with nls Catalog  
User Registry

**Username \***  
fred

**Email address \***  
fred@apic.com

**First Name \***  
Fred

**Last Name \***  
Bloggs

**Department code \***  
ABC123  
**Invalid department code.**

**Consumer organization \***

6. Change the **Department code** field to **DEP245**, reenter the password and click **Sign up** to see that a valid code is accepted.

## What you did in this tutorial

You added some validation to the Department code field on the Sign-up form, then tested the form to ensure that the **Department code** field accepts only codes in the format of **DEPnnn**.

In reality, you would need to add this configuration in such a way that it is available for all sites. To achieve this availability, you would include the configuration in your custom module. For more information, see [Drupal custom modules](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Tutorial: Adding a field group to group fields in a content type

You might want to change how your user account pages are displayed when you have lots of fields.

If you have many fields for one of your content types, for example, a user, then you might want to group some of these fields together when you display them to other users, or when these fields are being edited inside a user form.

Field groups allow you to create a parent field that contains other fields, set up as children, within that group. The **Fieldgroup** forms come with default HTML wrappers such as vertical tabs, horizontal tabs, accordions, fieldsets, or div wrappers.

---

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial.

---

### About this tutorial

You will create field groups for various groups of user information. These groups can then be seen under the users own section on the account view and edit pages.

This tutorial takes you through the following steps:

1. [Adding a field group called Consumer Organization to the user account display.](#)
2. [Adding a field group called Personal Details to the user form display.](#)

---

## Adding a field group called Consumer Organization to the user account display

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click Manage to display it.
3. Navigate to Configuration » People » Account settings » Manage display.
4. Click + **Add group**.
5. Select the group type **tab**.
6. Enter **Consumer Organization** as the label.
7. Click **Save and continue**.
8. Set the default state to **open**. This sets the tab to be expanded by default.
9. Click **Create group**.

[Home](#) » [Administration](#) » [Configuration](#) » [People](#) » [Account settings](#) » [Manage display](#)

Field group label

Default state

Description

ID

Extra CSS classes

Create group

When the group is created, you can assign fields to that group and rearrange the group on the page. After you create the group, you are on the **Manage Display** page.

1. Drag your new group from the disable section up into the upper section, just underneath **Picture**.
2. Drag **URL, Consumer Organization**, and **Consumer Organization URL** underneath the **Consumer Organization** group row, and slightly to the right. It fixates itself underneath and to the right of the parent, and this is now set as the child of that group.

3. Click **Save**.

FIELD	LABEL	FORMAT	
⊕ First Name	Above	Plain text	⚙
⊕ Last Name	Above	Plain text	⚙
⊕ Member for			
⊕ Picture	Above	Image	Original image ⚙
⊕ Consumer Organization		Tab	Tab: open <a href="#">delete</a> ⚙
⊕ URL	Above	Plain text	⚙
⊕ Consumer organization	Above	Plain text	⚙
⊕ Consumer organization URL	Above	Plain text	⚙
⊕ State	Above	Default	
⊕ Language code	Above	Language	⚙

You are now able to see the new group of fields on the account page of a user.

### Consumer Organization

**URL**  
 /consumer-api/user-registries/574dbbea-e045-4457-91f3-2e8581a72d6c/users/e29beb15-2fd6-4583-854c-fe2854389ee7

**Consumer organization**  
 Joe Smith

**Consumer organization URL**  
 /consumer-api/orgs/7d9808be-eae5-44dc-a45c-551e17c2b7d1

## Adding a field group called Personal Details to the user form display

1. Navigate to Configuration > People > Account settings > Manage form display.
2. Click **Add group**.
3. Select the group type **tab**.
4. Enter **Personal Details** as the label.

[Home](#) » [Administration](#) » [Configuration](#) » [People](#) » [Account settings](#) » [Manage form display](#)

Add a new group \*

Tab

Label \*

Personal Details Machine name: group\_personal\_details [Edit]

Save and continue

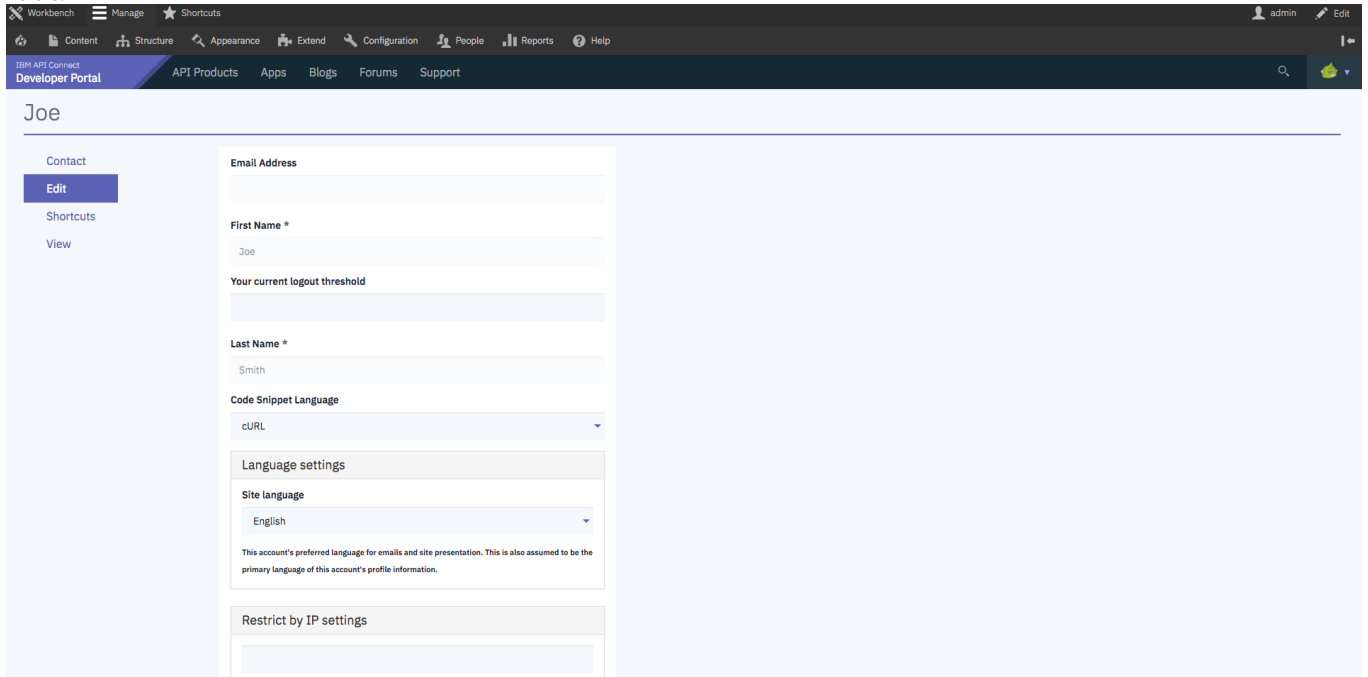
5. Click **Save and continue**.
6. Set the default state to **open**. This sets the tab to be expanded by default.
7. Click **Create group**.

Again, when the group is created, you can assign fields to that group and then rearrange the group on the page as you require. After you create the group, you are returned to the **Manage Display** page.

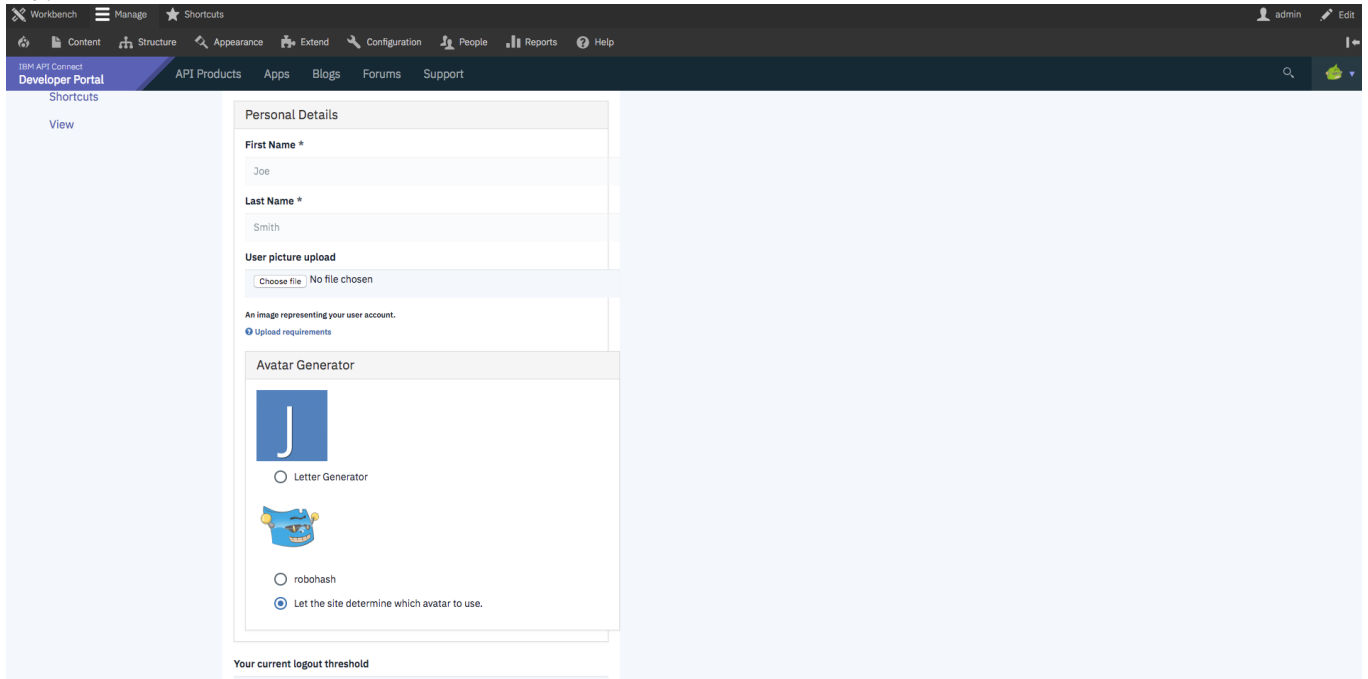
1. Drag your new group from the disabled section up into the upper section, onto the highest row.
2. Drag **First Name**, **Last Name**, **User picture upload**, and **Avatar Generator** to underneath the **Personal Details** group row, and slightly to the right. It fixates itself underneath and to the right of the parent, and this is now set as the child of that group.
3. Click **Save**.

You and your user can now see the new grouped section when you edit their account.

Before:



After:



## What you did in this tutorial

You have successfully created field groups for various groups of user information, and presented them to the user under its own section on the account view and edit pages.

You can check whether your field groups are applied, by viewing the profile of another user. Do this check by finding any other users published content and clicking the associated avatar. Or, from the administrator dashboard you can click **People** and then click a user. You can see the new Consumer Organization field group on their profile page. If you then click **Edit** for the user, you can see the Personal Details field group.

## What to do next

You can edit the user displays anytime by navigating back to **Configuration** > **People** > **Account settings** > **Manage display** or **Manage form display**.

You can delete the field group by clicking the **delete** button in the field group row on either of these pages. You can also modify the type of group it is, for example, vertical tabs, horizontal tabs, accordions, and field sets.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Adding a custom block to the front page

You can customize the appearance of the front page of your Developer Portal by adding a custom block. Blocks are boxes of content that are rendered into an area, or region, of a web page such as the Products page or Apps page that can be displayed in regions, such as footer or sidebar, on your page.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so. The tutorial [Adding a custom block to a page other than the front page](#) explains how to add a custom block to any other page.

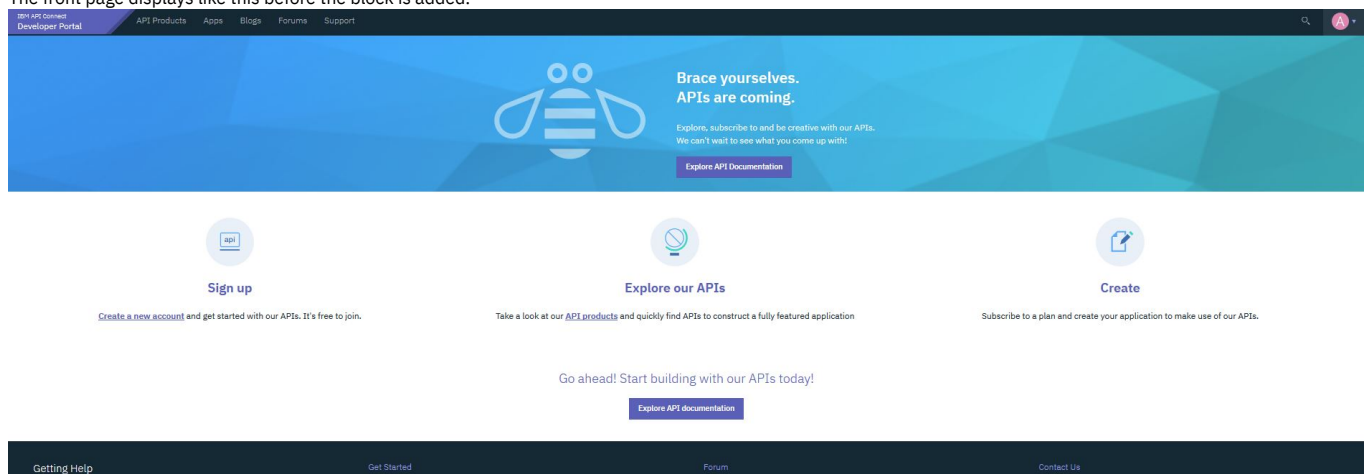
### About this tutorial

In this tutorial, you create a block that contains a marketing announcement and add it to the front page of your Developer Portal.

This tutorial takes you through the following steps:

1. [Creating a custom block.](#)
2. [Adding the custom block to the front page.](#)

The front page displays like this before the block is added.



### Creating a custom block

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Structure**, **Block layout**, **Add custom block**.
4. Enter **Marketing Info** in the **Block description**.
5. In the body area, click **Source**, then enter this HTML:

```
<div class="tutorial_block_1 tutorial_block_frontpage">
<h1><span>Create with our APIs</span></h1>
<p><span>Welcome to our Developer Portal where you will find APIs to create your awesome app.</span></p>
</div>
```

## Add custom block ☆

[Home](#)

### Block description \*

Marketing Info

A brief description of your block.

### Body

**B I U S** x<sup>2</sup> x<sub>2</sub> | *I*<sub>x</sub> | ≡ ≡ ≡ ≡ | 🔗 🔗 🔗 🔗 | ⋮ ⋮ ⋮ ⋮ | ☰ Ω ” ” | 🖼️ 🖼️ 🗨️ 🗨️ 🗨️ | Format ▾ | 📄 Source 📄

```
<div class="tutorial_block_1 tutorial_block_frontpage">
<h1><span>Create with our APIs</span></h1>

<p><span>Welcome to our Developer Portal where you will find APIs to create your awesome app.</span></p>
</div>
```

Text format

6. To see a preview of the block click **Source** again.

## Add custom block ☆

[Home](#)

### Block description \*

Marketing Info

A brief description of your block.

### Body

**B I U S** x<sup>2</sup> x<sub>2</sub> | *I*<sub>x</sub> | ≡ ≡ ≡ ≡ | 🔗 🔗 🔗 🔗 | ⋮ ⋮ ⋮ ⋮ | ☰ Ω ” ” | 🖼️ 🖼️ 🗨️ 🗨️ 🗨️ | Format ▾ | 📄 Source 📄

# Create with our APIs

Welcome to our Developer Portal where you will find APIs to create your awesome app.

Text format

Note: This block uses almost no inline-styling, instead it allows your theme to decide how it is rendered. The advantage of this set up is that you can easily change the rendering without modifying existing content in the database. Any styling that is required can be added to a custom theme. For more information, see [Tutorial: Creating a custom theme for the Developer Portal](#).

7. Click **Save**.

You have now created the custom block.

## Adding the custom block to the front page

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Structure > Pages**.
3. Click **Edit** on the **Welcome** page label.
4. Click **Panels** in the **Variants** section.
5. Click **Content** on the expanded menu.
6. Click + **Add new block**.
7. Click **Marketing Info** in the Custom list.
8. Clear **Display title**, keep **Content** in the drop-down list, and click **Add block**.



Add block
✕

**Block description**  
Marketing Info

**Title \***

Display title

**Region \***

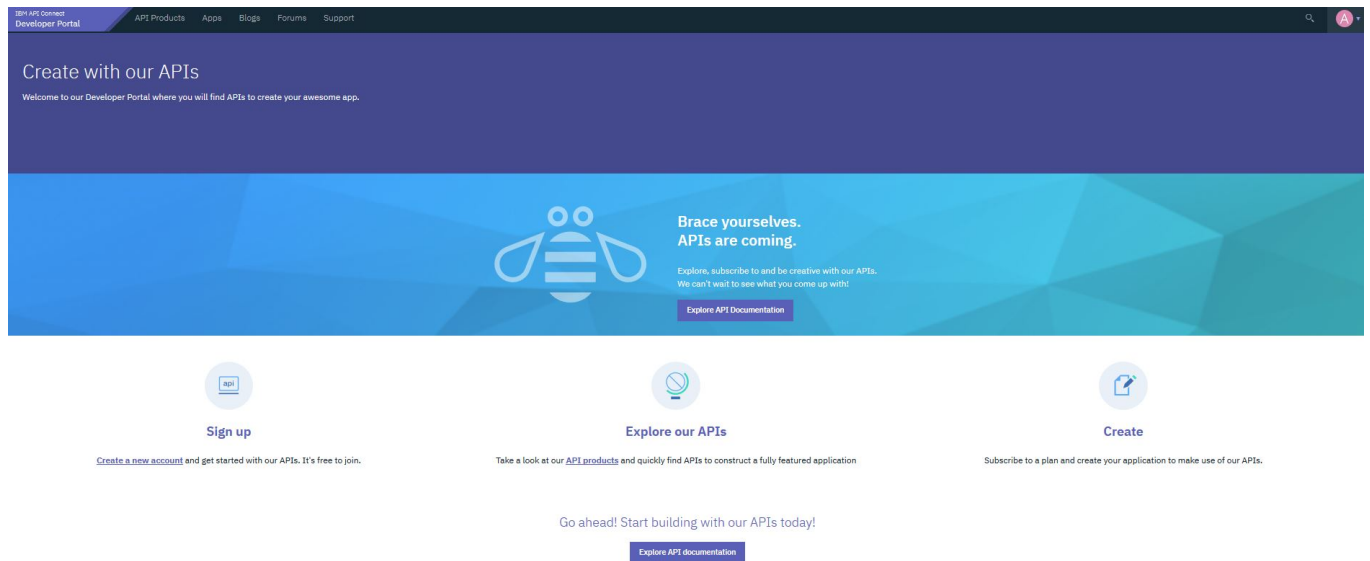
Content ▾

Add block

9. Drag **Marketing Info** to the order you want, in this example before the **Welcome banner** content.
10. Click **Update and save**.

## What you did in this tutorial

The front page now has the new block with a marketing message.



## What to do next

You can add further branding by adding content to your custom blocks such as images, or try using one of the provided custom blocks such as the Featured Content Block. Instead of using inline styling like that provided in the html, adopt best practice by adding class selectors and developing a custom theme. By default this block is English, you might want to consider adding translated versions.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Adding a custom block to a page other than the front page

You can customize the appearance of a page on your Developer Portal by adding a custom block. Blocks are boxes of content that are rendered into an area or region of a web page, such as the Products page or Apps page. Blocks can be displayed in regions, such as footer or sidebar, on your page.

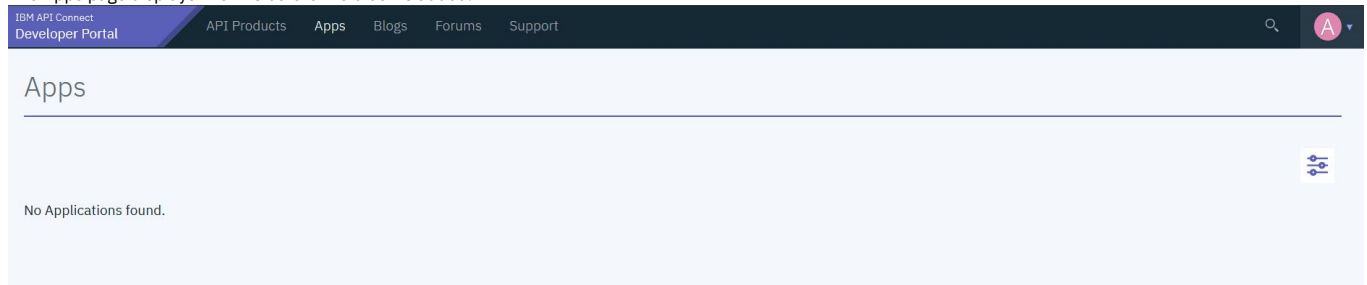
## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so. The tutorial [Adding a custom block to the front page](#) explains how to add a custom block to the front page.

## About this tutorial

In this tutorial, you create a custom block that contains extra guidance on application creation and registration, and add that custom block to the Apps page of your Developer Portal.

The Apps page displays like this before the block is added.



This tutorial takes you through creating a custom block, and configuring it to show on the Apps page.

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Structure > Block layout > Add custom block**.
4. Enter **App Info** in the **Block description**.
5. In the body area, click **Source**, then enter this HTML:

```
<div class="tutorial_block_2">
<p>Before you can use an API, you must create an application. When you create an application you are provided with a client ID and client secret for the application. You must supply the client ID when you call an API that requires you to identify your application by using a client ID, or a client ID and client Secret.</p>
</div>
```

**Block description \***

  
A brief description of your block.

**Body**

**Source**

```
<div class="tutorial_block_2">
<p>Before you can use an API, you must create an application. When you create an application you are provided with a client ID and client secret for the application. You must supply the client ID when you call an API that requires you to identify your application by using a client ID, or a client ID and client Secret.</p>
</div>
```

Text format: Full HTML

[About text formats](#)

6. To see a preview of the block click **Source** again.

**Block description \***

  
A brief description of your block.

**Body**

**Source**

Before you can use an API, you must create an application. When you create an application you are provided with a client ID and client secret for the application. You must supply the client ID when you call an API that requires you to identify your application by using a client ID, or a client ID and client Secret.

Text format: Full HTML

[About text formats](#)

Note: This block uses almost no inline-styling, instead it allows your theme to decide how it is rendered. The advantage of this set up is that you can easily change the rendering without modifying existing content in the database. Any styling that is required can be added to a custom theme. For more information, see [Tutorial: Creating a custom theme for the Developer Portal](#).

7. Click **Save**.

8. Enter `Creating an App` in the **Title** box, keep the **Display title** selected.

✓ Basic block *App Info* has been created.

**Block description:** App Info

**Title \***

Creating an App

Machine name: appinfo [Edit]

This field supports tokens. [Browse available tokens.](#)

Display title

9. Click **Pages** in the **Visibility** list.

10. Enter `/application` in the **Pages** text area, keep the **Show for the listed pages** selected.

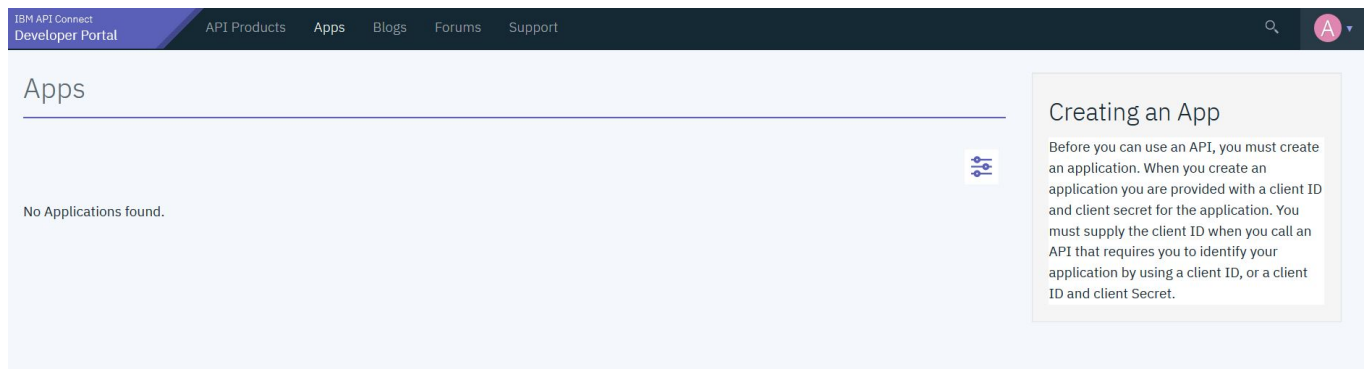
11. Select **Secondary** from the **Region** drop-down menu.

12. Click **Save block**.

13. Click **Save blocks**.

## What you did in this tutorial

The **Apps** page now has an additional block with a guidance message.



The screenshot shows the IBM API Connect Developer Portal interface. The top navigation bar includes 'API Products', 'Apps', 'Blogs', 'Forums', and 'Support'. The main content area is titled 'Apps' and displays 'No Applications found.' A sidebar block titled 'Creating an App' provides guidance: 'Before you can use an API, you must create an application. When you create an application you are provided with a client ID and client secret for the application. You must supply the client ID when you call an API that requires you to identify your application by using a client ID, or a client ID and client Secret.'

## What to do next

Instead of using inline styling like that provided in the html, adopt best practice by adding class selectors and developing a custom theme. By default this block is English, so you might want to consider adding translated versions.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Configuring the RobotsTxt file

You can control the access of a visiting Web robot. You can configure the `robots.txt` file that exists on your web server, usually at the root level, to control access. Web robots are programs that crawl through the web to obtain web content for all the sites that are visited, and provide indexing for better performance of search engines. You can also specify separate rules for different robots.

## Why would I want to edit Drupal's pre-existing robots.txt file?

Malicious robots might choose not to honor the `robots.txt` file, and by editing this file you are broadcasting which sites you do not want others to see. Therefore, you should not use this file to hide sensitive data. Instead, you might want to edit your `robots.txt` file to:

- Prevent duplicate information from being identified on your site
- Prevent internal pages from appearing in search engines
- Prevent private pages from appearing in search engines
- Prevent particular images, files, and so on, from being crawled
- Specify a `crawl-delay` attribute to prevent robots from overloading your server at load time
- Exclude a particular robot

## Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial.

## About this tutorial

---

You will edit the pre-existing `robots.txt` file and exclude access to a visiting robot called BadBot.

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Navigate to **Configuration** » **Search and Metadata** » **RobotsTxt**.

### RobotsTxt

[Home](#) » [Administration](#) » [Configuration](#) » [Search and metadata](#)

See <http://www.robotstxt.org/> for more information concerning how to write your `robots.txt` file.

#### Contents of robots.txt

```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used: http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html

User-agent: *
# CSS, JS, Images
Allow: /core/*.css$
Allow: /core/*.css?
Allow: /core/*.js$
Allow: /core/*.js?
```

**Save configuration**

4. Enter the policy to exclude access to a robot called BadBot.

```
User-agent: BadBot
Disallow: /
```

5. Click **Save Configuration** to save your changes.

## What you did in this tutorial

---

You have now successfully customized the `robots.txt` file. Robots now use this updated file to decide where they can crawl on your site. The BadBot robot is excluded access.

You can check whether your `robots.txt` file is successfully changed by navigating to your site and appending `/robots.txt`. You should see the content you entered into that file.

```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used: http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html

User-agent: BadBot
Disallow: /

User-agent: *
# CSS, JS, Images
Allow: /core/*.css$
Allow: /core/*.css?
Allow: /core/*.js$
Allow: /core/*.js?
```

For more information on how to edit your `robots.txt` file, see <https://www.robotstxt.org/>.

---

## What to do next

You can edit the `robots.txt` at any time by navigating back to the page within the configuration settings. You might choose to duplicate this file across all of your sites, or choose different policies for different sites.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorials for using custom modules in the Developer Portal

You can configure custom modules to complete specific actions in the Developer Portal.

Note: You must have administrator access to complete the tutorials.

You can configure rules for the following events that occur in the Developer Portal:

- Change the profile page to display **firstname last name**, instead of **username**.
- Add validation to a field on the sign up form.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## Tutorial: Changing the profile page to display firstname lastname, instead of username

You might want to change the profile page of your users on your Developer Portal to display their **firstname** and **lastname** rather than their **username**.

---

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial.

---

### About this tutorial

In this tutorial, you change the profile page of your users to display their **firstname** and **lastname** rather than their **username**. The change to the profile page is done by using the custom module `change_account_display` that is already available in [IBM APIC Portal - addons](#). However, for more information, see [Custom module development: creating a module skeleton](#).

This tutorial takes you through the following steps:

1. [Cloning and packaging the change\\_account\\_display module](#).
2. [Installing and enabling the change\\_account\\_display module](#).

## Cloning and packaging the `change_account_display` module

---

1. Clone the custom module. Here is an example command if you are using Mac and Linux:

```
git clone https://github.com/ibm-apiconnect/devportal-addons
```

2. Package up the custom module.

```
cd devportal-addons/Drupal8/modules
tar -czf change_account_display.tgz change_account_display/
```

## Installing and enabling the `change_account_display` module

---

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend** > **Install new module**.
4. Click **Browse** from the **Upload a module or theme archive to install** section.
5. Navigate to and select the packaged custom module that you created earlier. For example, `change_account_display.tgz`, then click **Open**.

# API Developer Portal

## Update manager

✓ Installation was completed successfully.

### change\_account\_display

- Installed *change\_account\_display* successfully

### Next steps

- [Install another module](#)
- [Enable newly added modules](#)
- [Administration pages](#)

6. Click **Install**.
7. Click the **Enable newly added modules** link.
8. Enter **Change Account Display** in the filter box.
9. Check the check box next to **Change Account Display**.
10. Click **Enable**.

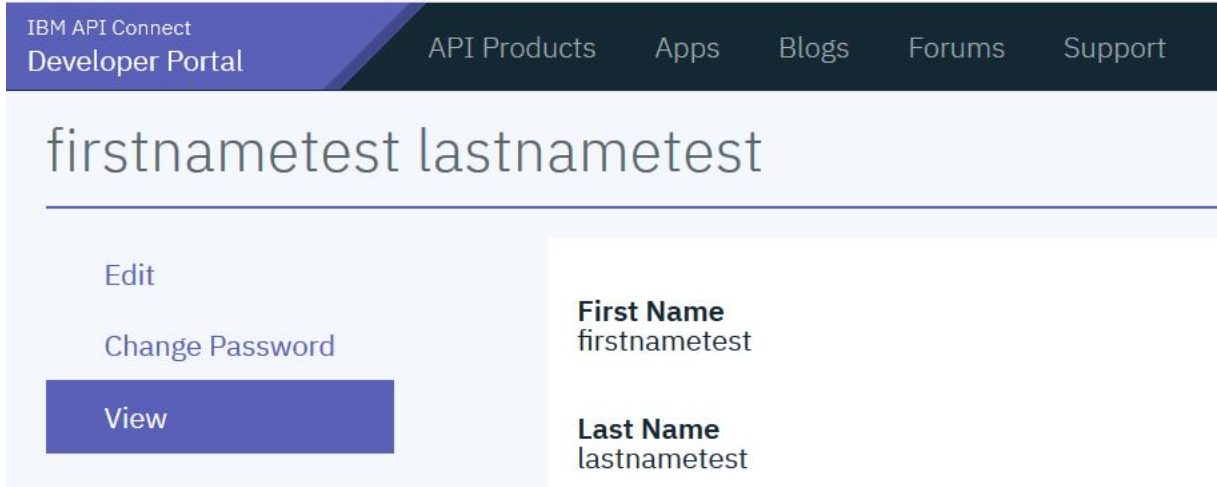
The module is now installed and enabled.

## Test the profile page change

---

1. Sign out as the Admin user.
2. Sign in as a user of your Developer Portal.
3. Click **My account** from the drop down next to your **username**.

4. Notice that your **firstname** and **lastname** are now displayed.



IBM API Connect  
Developer Portal

API Products Apps Blogs Forums Support

firstnametest lastnametest

Edit

Change Password

View

**First Name**  
firstnametest

**Last Name**  
lastnametest

## What you did in this tutorial

You used a custom module to change the profile page display for the users of your Developer Portal.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Adding validation to a field on the sign-up form

You might want to add extra validation to a field to help your Developer Portal users. For example, you might want to validate a department name, employee ID, or similar data when a user registers for an account on the portal.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. You must have completed the tutorial about [Adding a field to the sign up form](#), where you added a Department code field to the Sign-up form.

### About this tutorial

In this tutorial, you add some validation to the Department code field on the Sign-up form. This is a required field for all portal users and is a string of the format **DEPnnn**, where **DEP** is the string prefix and **nnn** is a three-digit number, for example, **DEP123**. The validation is done by using a custom module that is already available in [IBM APIC Portal - addons](#). However, for more information, see [Custom module development: creating a module skeleton](#).

This tutorial takes you through the following steps:

1. [Cloning and packaging the validation module](#).
2. [Installing and enabling the validation module](#).
3. [Testing the validation in the sign up form](#).

Look at the files we will use in this link [IBM APIC Portal - addons](#).

- `user_field_example.info.yml` - this is the metadata for the module, defining things like the name and version, and containing other important information that is required by Drupal to import and enable the module.

```
name: 'IBM APIC Portal - user field example'
type: module
description: 'IBM API Connect Developer Portal tutorial - Example of validating a custom user field'
package: 'Custom'
core_version_requirement: ^8 || ^9
version: 1.0.0
project: 'user_field_example'
dependencies:
  - ibm_apim
  - auth_apic
```

- `user_field_example.module` - this contains a validation function and adds this to the Sign-up form by using the `hook_form_alter` function.

```
<?php
use Drupal\Core\Form\FormStateInterface;

/**
 * Implementation of hook_form_alter() to alter the Sign up form
 */
function user_field_example_form_alter(&$form, FormStateInterface $form_state, $form_id) {
  if ($form_id === 'user_register_form') {
```

```

        $form['#validate'][] = 'user_field_example_validate_department_code';
    }
}

/**
 * Validate the Department code field on the Sign up form.
 *
 * Valid entry = DEPnnn
 * where n = single figure digit.
 */
function user_field_example_validate_department_code($form, &$form_state) {
    $dept_code = $form_state->getValue('field_department_code')[0]['value'];
    $valid = preg_match('/^DEP\d{3}$/', $dept_code);
    if (!$valid) {
        $form_state->setErrorByName('field_department_code', t('Invalid department code.'));
    }
}
}

```

## Cloning and packaging the validation module

---

Validation is added to the `Department code` field on the Sign-up form by using a custom module.

1. Clone the custom module. Here is an example command if you are using Mac and Linux:

```
git clone https://github.com/ibm-apiconnect/devportal-addons
```

2. Package up the custom module.

```
cd devportal-addons/apic_v10/modules
tar -czf user_field_example.tgz user_field_example/
```

## Installing and enabling the validation module

---

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Extend > Install new module**.
4. Click **Choose file** from the **Upload a module or theme archive to install** section.
5. Browse to and select `user_field_example.tgz`, the packaged custom module that you created earlier. Click **Open**.
6. Click **Install**.
7. Click the **Enable newly added modules** link.
8. Enter **IBM APIC Portal - user field example** in the filter box.
9. Check the checkbox next to **IBM APIC Portal - user field example**.
10. Click **Enable**.

The module is now installed and enabled.

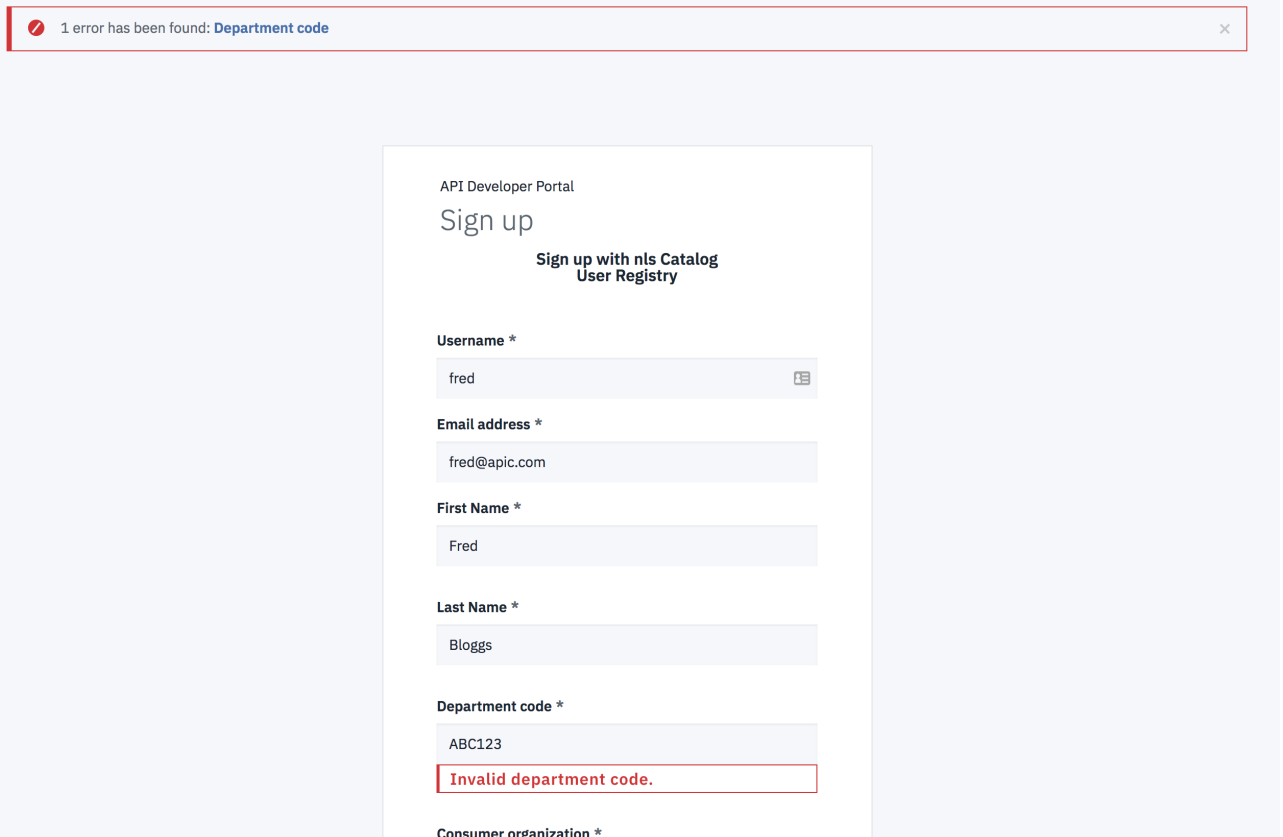
## Testing the validation in the sign-up form

---

1. Sign out as the Admin user.
2. Click **Create account**.
3. Complete all fields, set **Department code** to **ABC123**.
4. Click **Sign up**.



5. Observe the failure message and highlighting on the **Department code** field.



The screenshot shows a sign-up form titled "API Developer Portal Sign up with nls Catalog User Registry". The form includes fields for Username, Email address, First Name, Last Name, Department code, and Consumer organization. The Department code field contains "ABC123" and has a red border with the message "Invalid department code." below it. At the top of the page, a red error message states "1 error has been found: Department code".

6. Change the **Department code** field to **DEP245**, reenter the password and click **Sign up** to see that a valid code is accepted.

## What you did in this tutorial

You added some validation to the Department code field on the Sign-up form, then tested the form to ensure that the **Department code** field accepts only codes in the format of **DEPnnn**.

In reality, you would need to add this configuration in such a way that it is available for all sites. To achieve this availability, you would include the configuration in your custom module. For more information, see [Drupal custom modules](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details. For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Tutorial: Using a custom weighting sort order on the product list page

You can use a custom weighting to sort your products on the API products list page of your Developer Portal.

### Before you begin

You must have a Developer Portal enabled, and you must have administrator access to complete this tutorial. The tutorial [Creating the Portal](#) explains how to enable the portal if you have not already done so.

### About this tutorial

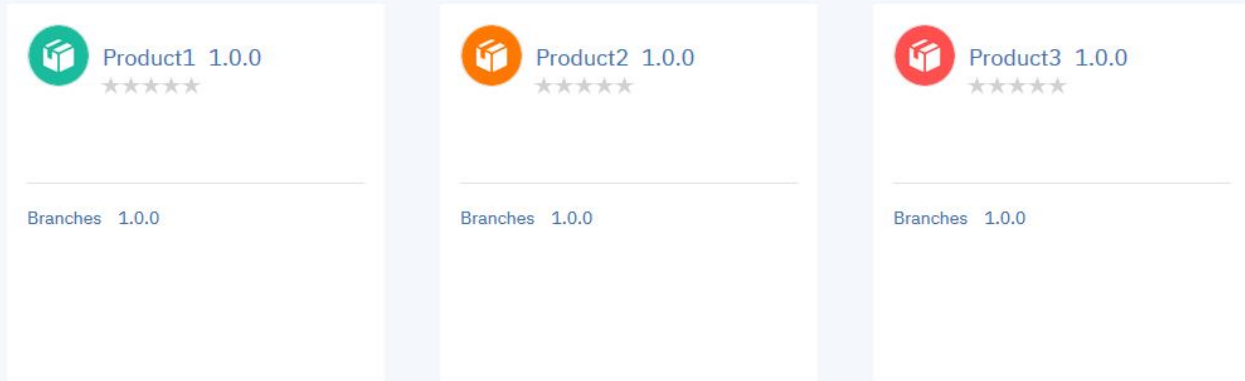
In this task, you set up a weighting sort criteria and use it to list your products, on the API Products page, in a particular order. You set up the weightings so the order reads, Product3, Product1, and Product2.

This tutorial takes you through the following steps:

1. [Add a custom integer field to the API content type.](#)
2. [Modify the Products view to change the sort order.](#)
3. [Add a weighting to each Product.](#)

The API Products page displays Product1, Product2, and Product3 in this order before the weightings are added.

## API Products



### Add a custom integer field to the API content type

1. Log in to your Developer Portal as an administrator.
2. If the administrator dashboard isn't displayed, click **Manage** to display it.
3. Click **Structure** > **Content types** > **Product** > **Manage fields**.
4. Click **+ Add field**.
5. In **Add a new field**, select **Number (Integer)**. Enter **Weighting** as the label title.

#### Add field ☆

[Home](#) > [Administration](#) > [Structure](#) > [Content types](#) > [Product](#) > [Manage fields](#)

#### Add a new field

Number (integer) ▼

or

#### Re-use an existing field

- Select an existing field - ▼

#### Label \*

Weighting

Machine name: field\_weighting [Edit]

**Save and continue**

6. Click **Save and continue**.
7. Leave the **Allowed number of values** set to 1 and click **Save field settings**.
8. Enter 0 as the default value, click **Save settings**.
9. Click **Save**.

The **Weighting** field now shows in the **Manage fields** list.

### Modify the Products view to change the sort order

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Navigate to **Structure** > **Views**, scroll down to the **Products** view and click **Edit**.
3. In the **SORT CRITERIA** section click **Add**.
4. Enter **weight** in the search box. Then, select **Weighting (field\_weighting)**, and click **Apply** (all displays).
5. Leave the **Order** as **Sort ascending**, click **Apply** (all displays).
6. In the **SORTCRITERIA** section, select **Rearrange** from the drop-down menu next to the **Add**.

Rearrange sort criteria ✕

**For**  
All displays (except overridden) ▾

[Show row weights](#)

	REMOVE
✚ Content: Title Exposed	<a href="#">Remove</a>
✚ Content: Version (apic_version) Exposed	<a href="#">Remove</a>
✚ Content: Changed Exposed	<a href="#">Remove</a>
✚ Content: Authored on Exposed	<a href="#">Remove</a>
✚ Content: Rating (apic_rating) Exposed	<a href="#">Remove</a>
✚ Content: Weighting (field_weighting) asc	<a href="#">Remove</a>

Apply (all displays)
Cancel

7. For all rows except **Weighting (field\_weighting)**, click **Remove**.
8. Click **Apply (all displays)**.

You modified the **Products** view to sort based on **Weighting (field\_weighting)**.

## Add a weighting to each Product

1. If the administrator dashboard isn't displayed, click **Manage** to display it.
2. Click **Content**, and select **Product** from the **Content type**, click **Filter**. A list of available Products on your site is displayed.

+ Add content

Title Content type Published status Language  
 Product ▾ - Any - ▾ - Any - ▾

Filter

**Action**  
Delete content ▾

Apply to selected items

<input type="checkbox"/>	TITLE	CONTENT TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
<input type="checkbox"/>	Product3	Product	admin	Published	06/18/2019 - 15:34	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px;">Edit</span> ▾
<input type="checkbox"/>	Product2	Product	admin	Published	06/18/2019 - 15:32	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px;">Edit</span> ▾
<input type="checkbox"/>	Product1	Product	admin	Published	06/18/2019 - 15:31	<span style="border: 1px solid #ccc; border-radius: 3px; padding: 2px 5px;">Edit</span> ▾


3. For Product1 click edit, in the **Weighting field**, select 2 as the value, click **Save**.
4. For Product2 click edit, in the **Weighting field**, select 3 as the value, click **Save**.
5. For Product3 click edit, in the **Weighting field**, select 1 as the value, click **Save**.

The API Products page displays Product3, Product1, and Product2 in this order after the weightings are added.

## What you did in this tutorial

You changed the order that the API Products appear on the products page.


## API Products



**Product3 1.0.0**  
★★★★★

---


Branches 1.0.0



**Product1 1.0.0**  
★★★★★

---

Branches 1.0.0



**Product2 1.0.0**  
★★★★★

---

Branches 1.0.0

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Reference information

Reference information for IBM® API Connect, including summary reference material for the developer toolkit command-line tool (CLI), a link to the API Connect Version 2018.4.1.x Whitepaper, and information about the API Connect platform REST APIs.

Reference item	Description
<a href="#">Command-line tool reference for the developer toolkit</a>	Summary reference material for each of the available commands for the developer toolkit command-line tool (CLI). (For more detailed information about how to use the command-line tool, see <a href="#">Using the developer toolkit command-line tool</a> .)
<a href="#">API Connect REST APIs</a>	The platform REST APIs that can be used to manage the API Connect cloud configuration, access the API provider capability, and use the consumer API microservice.
<a href="#">API Connect 2018.4.1.x Whitepaper</a>	A comprehensive technical guide to best practices, considerations, and deployment options for API Connect. The whitepaper covers the major components of API Connect, as well as considerations for configuring different clouds and environments, to help users with their API strategies. The target audience for the whitepaper are solution and integration architects.

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic

APIConnect toolkit 43590781acfc4fbbc64f341c89928aeb1a6bb86b (Built 2020-09-02T16:45:15Z)

## Synopsis

APIConnect toolkit 43590781acfc4fbbc64f341c89928aeb1a6bb86b (Built 2020-09-02T16:45:15Z)

**apic** [flags]

## Options

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>-h, --help</code>	Help for apic

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic activations

Activations operations

### Synopsis

Activations operations

```
apic activations [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for activations
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic activations:clear

Activations clear operations

### Synopsis

Activations clear operations

```
apic activations:clear [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--confirm string      Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for activations:clear
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic activations:delete

Activations delete operations

### Synopsis

---

Activations delete operations

```
apic activations:delete [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for activations:delete
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic activations:get

Activations get operations

### Synopsis

---

Activations get operations

```
apic activations:get [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for activations:get
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string       scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic activations:list

## Synopsis

---

Activations list operations

```
apic activations:list [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for activations:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic analytics

Analytics operations

## Synopsis

---

Analytics operations

```
apic analytics [flags]
```

## Options

---

<code>-h, --help</code>	Help for analytics
-------------------------	--------------------

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic analytics:create

Analytics create operations

## Synopsis

---

Analytics create operations

```
apic analytics:create [flags]
```

## Options

---

```

--analytics-service string  Analytics Service name or id (required)
-c, --catalog string       Catalog name or id (required)
--format string            Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                 Help for analytics:create
-o, --org string           Organization name or id (required)
--output string            Write file(s) to directory, instead of STDOUT (default "-")
--query string             Add query to request
--scope string             scope
-s, --server string        management server endpoint (required)
--space string            Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license          Accept the license for API Connect
--debug                  Enable debug output
--debug-output string    Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string        Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic analytics-services

Analytics Services operations

### Synopsis

Analytics Services operations

```
apic analytics-services [flags]
```

### Options

```

--availability-zone string  Availability Zone name or id (required)
-c, --catalog string       Catalog name or id (required)
--fields string            List of field names to return
--format string            Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                 Help for analytics-services
--limit int32              Maximum number of items to return
--offset int32             Offset item number from list to begin return
-o, --org string           Organization name or id (required)
--output string            Write file(s) to directory, instead of STDOUT (default "-")
--scope string             scope
-s, --server string        management server endpoint (required)
--space string            Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license          Accept the license for API Connect
--debug                  Enable debug output
--debug-output string    Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string        Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic analytics-services:clear

Clear the Analytics Service objects

### Synopsis

Clear the Analytics Service objects

```
apic analytics-services:clear [flags]
```

### Options

```

--availability-zone string  Availability Zone name or id (required)
--confirm string           Confirmation for critical updates (required)

```



```
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for analytics-services:clear
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic analytics-services:create

Create a Analytics Service object

### Synopsis

---

Create a Analytics Service object

```
apic analytics-services:create [flags]
```

### Options

---

```
--availability-zone string Availability Zone name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for analytics-services:create
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string      management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic analytics-services:delete

Delete the Analytics Service object by name or id

### Synopsis

---

Delete the Analytics Service object by name or id

```
apic analytics-services:delete [flags]
```

### Options

---

```
--availability-zone string Availability Zone name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for analytics-services:delete
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string      management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
```

```
--debug-output string  Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic analytics-services:get

Get the Analytics Service object by name or id

---

### Synopsis

Get the Analytics Service object by name or id

```
apic analytics-services:get [flags]
```

---

### Options

```
--availability-zone string  Availability Zone name or id (required)
-c, --catalog string       Catalog name or id (required)
--fields string           List of field names to return
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for analytics-services:get
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, use - for STDOUT. (default: cwd)
--query string            Add query to request
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
```

---

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic analytics-services:list

List the Analytics Service objects

---

### Synopsis

List the Analytics Service objects

```
apic analytics-services:list [flags]
```

---

### Options

```
--availability-zone string  Availability Zone name or id (required)
-c, --catalog string       Catalog name or id (required)
--fields string           List of field names to return
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for analytics-services:list
--limit int32             Maximum number of items to return
--offset int32            Offset item number from list to begin return
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
```

---

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic analytics-services:update

Update the Analytics Service object by name or id

### Synopsis

Update the Analytics Service object by name or id

```
apic analytics-services:update [flags]
```

### Options

```
--availability-zone string  Availability Zone name or id (required)
--format string             Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                 Help for analytics-services:update
-o, --org string           Organization name or id (required)
--output string            Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string        management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic api-keys

Api Keys operations

### Synopsis

Api Keys operations

```
apic api-keys [flags]
```

### Options

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for api-keys
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic api-keys:create

Create a API Key object

### Synopsis

---

Create a API Key object

```
apic api-keys:create [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for api-keys:create  
--output string  Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string  Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic api-keys:delete

Delete the API Key object by name or id

### Synopsis

---

Delete the API Key object by name or id

```
apic api-keys:delete [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for api-keys:delete  
--output string  Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string  Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic api-keys:get

Get the API Key object by name or id

### Synopsis

---

Get the API Key object by name or id

```
apic api-keys:get [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for api-keys:get
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic api-keys:list

List the API Key objects

## Synopsis

---

List the API Key objects

```
apic api-keys:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for api-keys:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic apis

Apis operations

## Synopsis

---

Apis operations

```
apic apis [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for apis
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
```

```
--scope string      scope
-s, --server string  management server endpoint (required)
--space string      Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:clone

Apis clone operations

## Synopsis

---

Apis clone operations

```
apic apis:clone [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for apis:clone
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:document

Apis document operations

## Synopsis

---

Apis document operations

```
apic apis:document [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for apis:document
--id                 id
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:get

Apis get operations

### Synopsis

---

Apis get operations

```
apic apis:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return (default "add(wsd1,api)")
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apis:get
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:list

Apis list operations

### Synopsis

---

Apis list operations

```
apic apis:list [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apis:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:list-all

Apis list-all operations

### Synopsis

Apis list-all operations

```
apic apis:list-all [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for apis:list-all
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:update

Apis update operations

### Synopsis

Apis update operations

```
apic apis:update [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for apis:update
--id string          id
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
```



```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apis:wsdl

Apis wsdl operations

### Synopsis

Apis wsdl operations

```
apic apis:wsdl [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for apis:wsdl
--id                id
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string      scope
-s, --server string  management server endpoint (required)
--space string      Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apps

Apps operations

### Synopsis

Apps operations

```
apic apps [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string    Consumer Organization name or id (required)
--expand string          List of transient field to expand
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help              Help for apps
--limit int32           Maximum number of items to return
--offset int32          Offset item number from list to begin return
-o, --org string        Organization name or id (required)
--output string         Write file(s) to directory, instead of STDOUT (default "-")
--scope string          scope
-s, --server string     management server endpoint (required)
--space string          Space name or id (required)
--space-initiated       space-initiated
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apps:clear

Clear the Application objects

### Synopsis

Clear the Application objects

```
apic apps:clear [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--confirm string          Confirmation for critical updates (required)
--consumer-org string     Consumer Organization name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for apps:clear
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated        space-initiated
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apps:create

Create a Application object

### Synopsis

Create a Application object

```
apic apps:create [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string     Consumer Organization name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for apps:create
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated        space-initiated
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apps:delete

Delete the Application object by name or id

### Synopsis

---

Delete the Application object by name or id

```
apic apps:delete [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apps:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic apps:get

Get the Application object by name or id

### Synopsis

---

Get the Application object by name or id

```
apic apps:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apps:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic apps:list

List the Application objects

### Synopsis

---

List the Application objects

```
apic apps:list [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--expand string</code>	List of transient field to expand
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apps:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic apps:update

Update the Application object by name or id

### Synopsis

---

Update the Application object by name or id

```
apic apps:update [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for apps:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic associates

Associates operations

### Synopsis

---

Associates operations

```
apic associates [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for associates
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic associates:get

Associates get operations

### Synopsis

---

Associates get operations

```
apic associates:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for associates:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic associates:list

Associates list operations

### Synopsis

---

Associates list operations

```
apic associates:list [flags]
```

### Options

---

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string    Consumer Organization name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for associates:list
--limit int32            Maximum number of items to return
--offset int32           Offset item number from list to begin return
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
--space-initiated       space-initiated
```

### Options inherited from parent commands

---

```
--accept-license        Accept the license for API Connect
--debug                 Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones

Availability Zones operations

### Synopsis

---

Availability Zones operations

```
apic availability-zones [flags]
```

### Options

---

```
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for availability-zones
--limit int32          Maximum number of items to return
--offset int32         Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license        Accept the license for API Connect
--debug                 Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:clear

Clear the Availability Zone objects

### Synopsis

---

Clear the Availability Zone objects

```
apic availability-zones:clear [flags]
```

### Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for availability-zones:clear
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:create

Create a Availability Zone object

### Synopsis

---

Create a Availability Zone object

```
apic availability-zones:create [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for availability-zones:create
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:delete

Delete the Availability Zone object by name or id

### Synopsis

---

Delete the Availability Zone object by name or id

```
apic availability-zones:delete [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for availability-zones:delete
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:get

Get the Availability Zone object by name or id

## Synopsis

---

Get the Availability Zone object by name or id

```
apic availability-zones:get [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for availability-zones:get
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:list

List the Availability Zone objects

## Synopsis

---

List the Availability Zone objects

```
apic availability-zones:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for availability-zones:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
```



```
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic availability-zones:update

Update the Availability Zone object by name or id

### Synopsis

---

Update the Availability Zone object by name or id

```
apic availability-zones:update [flags]
```

### Options

---

```
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for availability-zones:update
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic billings

Billings operations

### Synopsis

---

Billings operations

```
apic billings [flags]
```

### Options

---

```
--fields string List of field names to return
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for billings
--limit int32 Maximum number of items to return
--offset int32 Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic billings:clear

Clear the Billing objects

### Synopsis

Clear the Billing objects

```
apic billings:clear [flags]
```

### Options

```
--confirm string Confirmation for critical updates (required)
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:clear
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic billings:create

Create a Billing object

### Synopsis

Create a Billing object

```
apic billings:create [flags]
```

### Options

```
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:create
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic billings:delete

Delete the Billing object by name or id

### Synopsis

---

Delete the Billing object by name or id

```
apic billings:delete [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:delete
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic billings:get

Get the Billing object by name or id

### Synopsis

---

Get the Billing object by name or id

```
apic billings:get [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:get
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic billings:list

List the Billing objects

### Synopsis

---

List the Billing objects

```
apic billings:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic billings:update

Update the Billing object by name or id

## Synopsis

---

Update the Billing object by name or id

```
apic billings:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for billings:update
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalog-settings

Catalog Settings operations

## Synopsis

---

Catalog Settings operations

```
apic catalog-settings [flags]
```

## Options

---

```
-h, --help  Help for catalog-settings
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic catalog-settings:get

Get the Catalog Setting object

### Synopsis

Get the Catalog Setting object

```
apic catalog-settings:get [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for catalog-settings:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic catalog-settings:update

Update the Catalog Setting object

### Synopsis

Update the Catalog Setting object

```
apic catalog-settings:update [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for catalog-settings:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic catalogs

Catalogs operations

### Synopsis

---

Catalogs operations

```
apic catalogs [flags]
```

### Options

---

```
--fields string    List of field names to return
--format string    Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for catalogs
--limit int32      Maximum number of items to return
--my              my
--offset int32     Offset item number from list to begin return
-o, --org string   Organization name or id (required)
--output string    Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:clear

Clear the Catalog objects

### Synopsis

---

Clear the Catalog objects

```
apic catalogs:clear [flags]
```

### Options

---

```
--confirm string    Confirmation for critical updates (required)
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for catalogs:clear
-o, --org string    Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:create

Create a Catalog object

## Synopsis

---

Create a Catalog object

```
apic catalogs:create [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for catalogs:create  
-o, --org string Organization name or id (required)  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:delete

Delete the Catalog object by name or id

## Synopsis

---

Delete the Catalog object by name or id

```
apic catalogs:delete [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for catalogs:delete  
-o, --org string Organization name or id (required)  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:get

Get the Catalog object by name or id

## Synopsis

---

Get the Catalog object by name or id

```
apic catalogs:get [flags]
```

## Options

---

```
--fields string    List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for catalogs:get
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:list

List the Catalog objects

## Synopsis

---

List the Catalog objects

```
apic catalogs:list [flags]
```

## Options

---

```
--fields string    List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for catalogs:list
--limit int32     Maximum number of items to return
--my              my
--offset int32    Offset item number from list to begin return
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:transfer-owner

Transfer owner to an associate

## Synopsis

---

Transfer owner to an associate

```
apic catalogs:transfer-owner [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for catalogs:transfer-owner
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---



```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic catalogs:update

Update the Catalog object by name or id

### Synopsis

---

Update the Catalog object by name or id

```
apic catalogs:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for catalogs:update
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic client-creds

Manage the client configuration parameters. Client ID and Client Secret.

### Synopsis

---

Manage the client configuration parameters. Client ID and Client Secret.

```
apic client-creds [flags]
```

### Options

---

```
-g, --global  list the global configuration variables
-h, --help    Help for client-creds
-l, --local   list the local application configuration variables
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic client-creds:clear

clear the set client credentials.

### Synopsis

---

clear the set client credentials.

```
apic client-creds:clear [flags]
```

### Examples

---

```
$ apic client-creds:clear  
Deleted client credentials.
```

### Options

---

```
-g, --global    list the global configuration variables  
-h, --help      Help for client-creds:clear  
-l, --local     list the local application configuration variables
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic client-creds:list

List the set clientID/secret parameters.

### Synopsis

---

List the set clientID/secret parameters.

```
apic client-creds:list [flags]
```

### Examples

---

```
$ apic client-creds:list
```

### Options

---

```
-g, --global    list the global configuration variables  
-h, --help      Help for client-creds:list  
-l, --local     list the local application configuration variables
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic client-creds:set

Set the client configuration parameters. ClientID and Client Secret.

## Synopsis

---

Set the client configuration parameters. ClientID and Client Secret.

```
apic client-creds:set /path/to/creds/json ... [flags]
```

## Examples

---

```
$ apic client-creds:set /Users/local_user/credential.json
```

## Options

---

```
-g, --global list the global configuration variables
-h, --help   Help for client-creds:set
-l, --local  list the local application configuration variables
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic cloud-settings

Cloud Settings operations

## Synopsis

---

Cloud Settings operations

```
apic cloud-settings [flags]
```

## Options

---

```
-h, --help Help for cloud-settings
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic cloud-settings:about

Return information about the cloud

## Synopsis

---

Return public information about the cloud

```
apic cloud-settings:about [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for cloud-settings:about
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:audit-endpoint-test-connection

Test an audit endpoint connection

### Synopsis

---

Test an audit endpoint connection

```
apic cloud-settings:audit-endpoint-test-connection [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for cloud-settings:audit-endpoint-test-connection
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:designer-credentials-list

List credential for designer

### Synopsis

---

List credential for designer

```
apic cloud-settings:designer-credentials-list [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for cloud-settings:designer-credentials-list
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:get

Get the Cloud Setting object

### Synopsis

---

Get the Cloud Setting object

```
apic cloud-settings:get [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for cloud-settings:get
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:info

Return public information about the cloud deployment

### Synopsis

---

Return public information about the cloud deployment

```
apic cloud-settings:info [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for cloud-settings:info
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:mail-server-configured

Return true or false based on if mail server is configured or not

## Synopsis

---

Return true or false based on if mail server is configured or not

```
apic cloud-settings:mail-server-configured [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for cloud-settings:mail-server-configured  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:oauth2-certs

Support JWKS\_URI endpoint, conform to OIDC specification

## Synopsis

---

Return JWKS that is used to issue token. You MUST supply the --format option when using this command from the API Connect toolkit.

```
apic cloud-settings:oauth2-certs [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for cloud-settings:oauth2-certs  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:toolkit-credentials-list

List credential for toolkit and consumer toolkit

## Synopsis

---

List credential for toolkit and consumer toolkit

```
apic cloud-settings:toolkit-credentials-list [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for cloud-settings:toolkit-credentials-list  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:topology

Return the topology of the cloud for governance

### Synopsis

---

Return the topology of the cloud for governance

`apic cloud-settings:topology [flags]`

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for cloud-settings:topology
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic cloud-settings:update

Update the Cloud Setting object

### Synopsis

---

Update the Cloud Setting object

`apic cloud-settings:update [flags]`

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for cloud-settings:update
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic config

Manage configuration variables

### Synopsis

---

Manage configuration variables

```
apic config [flags]
```

### Options

---

```
-g, --global  list the global configuration variables
-h, --help    Help for config
-l, --local   list the local application configuration variables
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic config:clear

Delete all configuration variables

### Synopsis

---

Delete all configuration variables

```
apic config:clear [flags]
```

### Examples

---

```
$ apic config:clear
Deleted config catalog
Deleted config org
Deleted config space

$ apic config:clear --global
Deleted config catalog
```

### Options

---

```
-g, --global  list the global configuration variables
-h, --help    Help for config:clear
-l, --local   list the local application configuration variables
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic config:delete

Delete a configuration variable



## Synopsis

---

Configuration variables:

**catalog** The catalog configuration variable should be set to the URI of an API Connect catalog. The value of the catalog provides the default identity of a catalog for all the commands that are used to manage aspects of a catalog. The default values defined by the catalog's URI can be overridden using the `--server`, `--organization`, and `--catalog` command line options. The catalog URI has the form: `https://MANAGEMENT_SERVER/api/catalogs/ORGANIZATION_NAME/CATALOG_NAME`

**org** The default value of `org` defined by the app's or catalog's URI can be set using the `org` URI. The `org` URI has the form: `https://MANAGEMENT_SERVER/api/orgs/ORGANIZATION_NAME`

**space** The space configuration variable should be set to the URI of an API Connect space. The value of the space provides the default identity of a space for all the commands that are used to manage aspects of a space. The default values defined by the spaces's URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--space` command line options. The space URI has the form: `https://MANAGEMENT_SERVER/api/spaces/ORGANIZATION_NAME/CATALOG_NAME/SPACE_NAME`

**consumer** The consumer configuration variable should be set to the URI of an API Connect consumer. The value of the `consumer-org` provides the default identity of a consumer. The default values defined by the consumer URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--consumer` command line options. The consumer URI has the form: `https://MANAGEMENT_SERVER/api/consumer-orgs/ORGANIZATION_NAME/CATALOG_NAME/CONSUMER_ORG_NAME`

**cloud** The cloud configuration variable should be set to the management server URI. The value of the cloud variable provides default server URI for cloud admin commands. It can be overridden using the `--server` and `--mode` command line options. The cloud URI has the form: `https://MANAGEMENT_SERVER/api/`

**mode** The value of mode configuration variable defines the toolkit operation mode. It can be overridden using the `--mode` command line option. The value can be set to `apim` or `consumer`

**template-path** List of places to look for handlebar templates

**template-default-api** Defines the default api template

**template-default-product** Defines the default product template

**apic config:delete** `NAME...` [`flags`]

## Examples

---

```
$ apic config:delete catalog
Deleted config catalog
```

```
$ apic config:delete org
Deleted config org
```

```
$ apic config:delete template-path
Deleted config template-path
```

## Options

---

```
-g, --global list the global configuration variables
-h, --help   Help for config:delete
-l, --local  list the local application configuration variables
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic config:get

Get a configuration variable

## Synopsis

---

Configuration variables:

**catalog** The catalog configuration variable should be set to the URI of an API Connect catalog. The value of the catalog provides the default identity of a catalog for all the commands that are used to manage aspects of a catalog. The default values defined by the catalog's URI can be overridden using the `--server`, `--organization`, and `--catalog` command line options. The catalog URI has the form: `https://MANAGEMENT_SERVER/api/catalogs/ORGANIZATION_NAME/CATALOG_NAME`

**org** The default value of `org` defined by the app's or catalog's URI can be set using the `org` URI. The `org` URI has the form: `https://MANAGEMENT_SERVER/api/orgs/ORGANIZATION_NAME`

space The space configuration variable should be set to the URI of an API Connect space. The value of the space provides the default identity of a space for all the commands that are used to manage aspects of a space. The default values defined by the spaces's URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--space` command line options. The space URI has the form: `https://MANAGEMENT_SERVER/api/spaces/ORGANIZATION_NAME/CATALOG_NAME/SPACE_NAME`

consumer The consumer configuration variable should be set to the URI of an API Connect consumer. The value of the consumer-org provides the default identity of a consumer. The default values defined by the consumer URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--consumer` command line options. The consumer URI has the form: `https://MANAGEMENT_SERVER/api/consumer-orgs/ORGANIZATION_NAME/CATALOG_NAME/CONSUMER_ORG_NAME`

cloud The cloud configuration variable should be set to the management server URI. The value of the cloud variable provides default server URI for cloud admin commands. It can be overridden using the `--server` and `--mode` command line options. The cloud URI has the form: `https://MANAGEMENT_SERVER/api/`

mode The value of mode configuration variable defines the toolkit operation mode. It can be overridden using the `--mode` command line option. The value can be set to `apim` or `consumer`

template-path List of places to look for handlebar templates

template-default-api Defines the default api template

template-default-product Defines the default product template

```
apic config:get NAME [flags]
```

## Examples

```
$ apic config:get catalog
catalog: https://mgmthost.com/api/catalogs/climbon/sb
```

```
$ apic config:get org
org: https://mgmthost.com/api/orgs/climbon
```

```
$ apic config:get template-path
template-path: /etc/templates
```

## Options

```
-g, --global list the global configuration variables
-h, --help   Help for config:get
-l, --local  list the local application configuration variables
```

## Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic config:list

List the application and global configuration variables

## Synopsis

Configuration variables:

catalog The catalog configuration variable should be set to the URI of an API Connect catalog. The value of the catalog provides the default identity of a catalog for all the commands that are used to manage aspects of a catalog. The default values defined by the catalog's URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--catalog` command line options. The catalog URI has the form: `https://MANAGEMENT_SERVER/api/catalogs/ORGANIZATION_NAME/CATALOG_NAME`

org The default value of org defined by the app's or catalog's URI can be set using the org URI. The org URI has the form: `https://MANAGEMENT_SERVER/api/orgs/ORGANIZATION_NAME`

space The space configuration variable should be set to the URI of an API Connect space. The value of the space provides the default identity of a space for all the commands that are used to manage aspects of a space. The default values defined by the spaces's URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--space` command line options. The space URI has the form: `https://MANAGEMENT_SERVER/api/spaces/ORGANIZATION_NAME/CATALOG_NAME/SPACE_NAME`

consumer The consumer configuration variable should be set to the URI of an API Connect consumer. The value of the consumer-org provides the default identity of a consumer. The default values defined by the consumer URI can be overridden using the `--server`, `--organization`, `--catalog`, and `--consumer` command line options. The consumer URI has the form: `https://MANAGEMENT_SERVER/api/consumer-orgs/ORGANIZATION_NAME/CATALOG_NAME/CONSUMER_ORG_NAME`

cloud The cloud configuration variable should be set to the management server URI. The value of the cloud variable provides default server URI for cloud admin commands. It can be overridden using the `--server` and `--mode` command line options. The cloud URI has the form: `https://MANAGEMENT_SERVER/api/`

mode The value of mode configuration variable defines the toolkit operation mode. It can be overridden using the `--mode` command line option. The value can be set to `apim` or `consumer`

template-path List of places to look for handlebar templates

template-default-api Defines the default api template

template-default-product Defines the default product template

```
apic config:list [flags]
```

## Examples

---

```
$ apic config
catalog: https://mgmthost.com/api/catalogs/climbon/sb
org: https://mgmthost.com/api/orgs/climbon

$ apic config --global
catalog: https://mgmthost.com/api/catlogs/climbon/sb
```

## Options

---

```
-g, --global list the global configuration variables
-h, --help Help for config:list
-l, --local list the local application configuration variables
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic config:set

Set or update configuration variables

## Synopsis

---

Configuration variables:

catalog The catalog configuration variable should be set to the URI of an API Connect catalog. The value of the catalog provides the default identity of a catalog for all the commands that are used to manage aspects of a catalog. The default values defined by the catalog's URI can be overridden using the --server, --organization, and --catalog command line options. The catalog URI has the form: https://MANAGEMENT\_SERVER/api/catalogs/ORGANIZATION\_NAME/CATALOG\_NAME

org The default value of org defined by the app's or catalog's URI can be set using the org URI. The org URI has the form: https://MANAGEMENT\_SERVER/api/orgs/ORGANIZATION\_NAME

space The space configuration variable should be set to the URI of an API Connect space. The value of the space provides the default identity of a space for all the commands that are used to manage aspects of a space. The default values defined by the spaces's URI can be overridden using the --server, --organization, --catalog, and --space command line options. The space URI has the form: https://MANAGEMENT\_SERVER/api/spaces/ORGANIZATION\_NAME/CATALOG\_NAME/SPACE\_NAME

consumer The consumer configuration variable should be set to the URI of an API Connect consumer. The value of the consumer-org provides the default identity of a consumer. The default values defined by the consumer URI can be overridden using the --server, --organization, --catalog, and --consumer command line options. The consumer URI has the form: https://MANAGEMENT\_SERVER/api/consumer-orgs/ORGANIZATION\_NAME/CATALOG\_NAME/CONSUMER\_ORG\_NAME

cloud The cloud configuration variable should be set to the management server URI. The value of the cloud variable provides default server URI for cloud admin commands. It can be overridden using the --server and --mode command line options. The cloud URI has the form: https://MANAGEMENT\_SERVER/api/

mode The value of mode configuration variable defines the toolkit operation mode. It can be overridden using the --mode command line option. The value can be set to apim or consumer

template-path List of places to look for handlebar templates

template-default-api Defines the default api template

template-default-product Defines the default product template

```
apic config:set NAME=VALUE ... [flags]
```

## Examples

---

```
$ apic config:set catalog=https://mgmthost.com/api/catalogs/climbon/sb
catalog: https://mgmthost.com/api/catalogs/climbon/sb

$ apic config:set org=https://mgmthost2.com/api/orgs/hikeon
org: https://mgmthost2.com/api/orgs/hikeon
```

```
$ apic config:set --global template-path="/etc/templates"
template-path: /etc/templates
```

## Options

---

```
-g, --global    list the global configuration variables
-h, --help      Help for config:set
-l, --local     list the local application configuration variables
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-api-user-registries

Configured Api User Registries operations

## Synopsis

---

Configured Api User Registries operations

```
apic configured-api-user-registries [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-api-user-registries
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-api-user-registries:clear

Configured Api User Registries clear operations

## Synopsis

---

Configured Api User Registries clear operations

```
apic configured-api-user-registries:clear [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string      Confirmation for critical updates (required)
```

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-api-user-registries:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-api-user-registries:create

Configured Api User Registries create operations

### Synopsis

Configured Api User Registries create operations

```
apic configured-api-user-registries:create [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-api-user-registries:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-api-user-registries:delete

Configured Api User Registries delete operations

### Synopsis

Configured Api User Registries delete operations

```
apic configured-api-user-registries:delete [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-api-user-registries:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-api-user-registries:get

Configured Api User Registries get operations

### Synopsis

---

Configured Api User Registries get operations

```
apic configured-api-user-registries:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-api-user-registries:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-api-user-registries:list

Configured Api User Registries list operations

### Synopsis

---

Configured Api User Registries list operations

```
apic configured-api-user-registries:list [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-api-user-registries:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-billings

Configured Billings operations

### Synopsis

Configured Billings operations

```
apic configured-billings [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for configured-billings
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-billings:clear

Clear the Configured Billing objects

### Synopsis

Clear the Configured Billing objects

```
apic configured-billings:clear [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--confirm string      Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for configured-billings:clear
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-billings:create

Create a Configured Billing object

### Synopsis

---

Create a Configured Billing object

```
apic configured-billings:create [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-billings:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-billings:delete

Delete the Configured Billing object by name or id

### Synopsis

---

Delete the Configured Billing object by name or id

```
apic configured-billings:delete [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-billings:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-billings:get



Get the Configured Billing object by name or id

## Synopsis

---

Get the Configured Billing object by name or id

```
apic configured-billings:get [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for configured-billings:get
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-billings:list

List the Configured Billing objects

## Synopsis

---

List the Configured Billing objects

```
apic configured-billings:list [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for configured-billings:list
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-catalog-user-registries

Configured Catalog User Registries operations

## Synopsis

---

Configured Catalog User Registries operations

apic configured-catalog-user-registries [flags]

## Options

---

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-catalog-user-registries
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-catalog-user-registries:create

Create a Configured Catalog User Registry object

## Synopsis

---

Create a Configured Catalog User Registry object

apic configured-catalog-user-registries:create [flags]

## Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-catalog-user-registries:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-catalog-user-registries:delete

Delete the Configured Catalog User Registry object by name or id

## Synopsis

---

Delete the Configured Catalog User Registry object by name or id

apic configured-catalog-user-registries:delete [flags]

## Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-catalog-user-registries:delete

```
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-catalog-user-registries:get

Get the Configured Catalog User Registry object by name or id

### Synopsis

---

Get the Configured Catalog User Registry object by name or id

```
apic configured-catalog-user-registries:get [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-catalog-user-registries:get
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-catalog-user-registries:list

List the Configured Catalog User Registry objects

### Synopsis

---

List the Configured Catalog User Registry objects

```
apic configured-catalog-user-registries:list [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-catalog-user-registries:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-catalog-user-registries:search

Search for users in the catalog user registry

### Synopsis

Search for users in the catalog user registry

```
apic configured-catalog-user-registries:search [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-catalog-user-registries:search
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-gateway-services

Configured Gateway Services operations

### Synopsis

Configured Gateway Services operations

```
apic configured-gateway-services [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-gateway-services
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-gateway-services:clear

Configured Gateway Services clear operations

### Synopsis

Configured Gateway Services clear operations

```
apic configured-gateway-services:clear [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-gateway-services:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-gateway-services:create

Configured Gateway Services create operations

### Synopsis

Configured Gateway Services create operations

```
apic configured-gateway-services:create [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-gateway-services:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-gateway-services:delete

Configured Gateway Services delete operations

### Synopsis

---

Configured Gateway Services delete operations

```
apic configured-gateway-services:delete [flags]
```

### Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-gateway-services:delete
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-gateway-services:get

Configured Gateway Services get operations

### Synopsis

---

Configured Gateway Services get operations

```
apic configured-gateway-services:get [flags]
```

### Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-gateway-services:get
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-gateway-services:list

Configured Gateway Services list operations

## Synopsis

---

Configured Gateway Services list operations

```
apic configured-gateway-services:list [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-gateway-services:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers

Configured OAuth Providers operations

## Synopsis

---

Configured OAuth Providers operations

```
apic configured-oauth-providers [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-oauth-providers
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers:clear

Configured OAuth Providers clear operations

## Synopsis

---

Configured OAuth Providers clear operations

```
apic configured-oauth-providers:clear [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--confirm string       Confirmation for critical updates (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-oauth-providers:clear
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers:create

Configured OAuth Providers create operations

## Synopsis

---

Configured OAuth Providers create operations

```
apic configured-oauth-providers:create [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-oauth-providers:create
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers:delete

Configured OAuth Providers delete operations

## Synopsis

---

Configured OAuth Providers delete operations

```
apic configured-oauth-providers:delete [flags]
```



## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-oauth-providers:delete
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers:get

Configured OAuth Providers get operations

## Synopsis

---

Configured OAuth Providers get operations

```
apic configured-oauth-providers:get [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for configured-oauth-providers:get
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic configured-oauth-providers:list

Configured OAuth Providers list operations

## Synopsis

---

Configured OAuth Providers list operations

```
apic configured-oauth-providers:list [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
```

-h, --help	Help for configured-oauth-providers:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-tls-client-profiles

Configured Tls Client Profiles operations

### Synopsis

Configured Tls Client Profiles operations

```
apic configured-tls-client-profiles [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-tls-client-profiles
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic configured-tls-client-profiles:clear

Configured Tls Client Profiles clear operations

### Synopsis

Configured Tls Client Profiles clear operations

```
apic configured-tls-client-profiles:clear [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for configured-tls-client-profiles:clear
-o, --org string	Organization name or id (required)

```
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:clear-all

Configured Tls Client Profiles clear-all operations

### Synopsis

---

Configured Tls Client Profiles clear-all operations

```
apic configured-tls-client-profiles:clear-all [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string      Confirmation for critical updates (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-tls-client-profiles:clear-all
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:create

Configured Tls Client Profiles create operations

### Synopsis

---

Configured Tls Client Profiles create operations

```
apic configured-tls-client-profiles:create [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-tls-client-profiles:create
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:delete

Configured Tls Client Profiles delete operations

### Synopsis

---

Configured Tls Client Profiles delete operations

```
apic configured-tls-client-profiles:delete [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-tls-client-profiles:delete
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:get

Configured Tls Client Profiles get operations

### Synopsis

---

Configured Tls Client Profiles get operations

```
apic configured-tls-client-profiles:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-tls-client-profiles:get
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:list

Configured Tls Client Profiles list operations

### Synopsis

Configured Tls Client Profiles list operations

```
apic configured-tls-client-profiles:list [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-tls-client-profiles:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-tls-client-profiles:list-all

Configured Tls Client Profiles list-all operations

### Synopsis

Configured Tls Client Profiles list-all operations

```
apic configured-tls-client-profiles:list-all [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for configured-tls-client-profiles:list-all
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
```

```
--debug-output string  Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-org-settings

Consumer Org Settings operations

### Synopsis

---

Consumer Org Settings operations

```
apic consumer-org-settings [flags]
```

### Options

---

```
-h, --help  Help for consumer-org-settings
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-org-settings:delete

Delete the Consumer Organization Setting object

### Synopsis

---

Delete the Consumer Organization Setting object

```
apic consumer-org-settings:delete [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for consumer-org-settings:delete
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated    space-initiated
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-org-settings:get

Get the Consumer Organization Setting object

### Synopsis

---

Get the Consumer Organization Setting object

```
apic consumer-org-settings:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for consumer-org-settings:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-org-settings:update

Update the Consumer Organization Setting object

### Synopsis

---

Update the Consumer Organization Setting object

```
apic consumer-org-settings:update [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for consumer-org-settings:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-orgs

## Synopsis

---

Consumer Orgs operations

```
apic consumer-orgs [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help              Help for consumer-orgs
--limit int32          Maximum number of items to return
--offset int32         Offset item number from list to begin return
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic consumer-orgs:clear

Clear the Consumer Organization objects

## Synopsis

---

Clear the Consumer Organization objects

```
apic consumer-orgs:clear [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--confirm string        Confirmation for critical updates (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help              Help for consumer-orgs:clear
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic consumer-orgs:create

Create a Consumer Organization object



## Synopsis

---

Create a Consumer Organization object

```
apic consumer-orgs:create [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for consumer-orgs:create
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-orgs:delete

Delete the Consumer Organization object by name or id

## Synopsis

---

Delete the Consumer Organization object by name or id

```
apic consumer-orgs:delete [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for consumer-orgs:delete
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-orgs:get

Get the Consumer Organization object by name or id

## Synopsis

---

Get the Consumer Organization object by name or id

```
apic consumer-orgs:get [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for consumer-orgs:get
-o, --org string          Organization name or id (required)
--output string           Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated         space-initiated
```

## Options inherited from parent commands

---

```
--accept-license         Accept the license for API Connect
--debug                  Enable debug output
--debug-output string     Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string         Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-orgs:list

List the Consumer Organization objects

## Synopsis

---

List the Consumer Organization objects

```
apic consumer-orgs:list [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for consumer-orgs:list
--limit int32            Maximum number of items to return
--offset int32           Offset item number from list to begin return
-o, --org string          Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated         space-initiated
```

## Options inherited from parent commands

---

```
--accept-license         Accept the license for API Connect
--debug                  Enable debug output
--debug-output string     Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string         Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic consumer-orgs:transfer-owner

Transfer owner to an associate

## Synopsis

---

Transfer owner to an associate

```
apic consumer-orgs:transfer-owner [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for consumer-orgs:transfer-owner
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic consumer-orgs:update

Update the Consumer Organization object by name or id

### Synopsis

Update the Consumer Organization object by name or id

```
apic consumer-orgs:update [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for consumer-orgs:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic create

Create an API or product definition

### Synopsis

Create an API or product definition

```
apic create [flags]
```

### Options

<code>-h, --help</code>	Help for create
<code>-i, --interactive</code>	use interactive mode

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic create:api

Create an OpenAPI (Swagger) definition

### Synopsis

Create an OpenAPI (Swagger) definition

```
apic create:api [flags]
```

### Examples

Create an API

```
$ apic create:api --title Routes
Created routes.yaml API definition [routes:1.0.0]
```

Create an API and generate a product referencing the API

```
$ apic create:api --title Routes --product "Climb On"
Created routes.yaml API definition [routes:1.0.0]
Created climb-on.yaml product definition [climb-on:1.0.0]
```

Create an API interactively

```
$ apic create:api
Title: Routes
Name (routes): routes
File (routes.yaml): routes.yaml
Template: ()
Basepath (/routes): /routes
Hostname ($(catalog.host)): $(catalog.host)
Schemes:
Target url: ()
Create product [true]: true
Product title (Routes Product): Climb On
Product name (climb-on): climb-on
Product file (climb-on.yaml): climb-on.yaml
Created routes.yaml API definition [routes:1.0.0]
Created climb-on.yaml product definition [climb-on:1.0.0]
```

Create an API from a WSDL document

```
$ apic create:api --wsdl globalweather.wsdl
Created globalweather.yaml API definition [globalweather.yaml:1.0.0]
```

Create an API using APIC's default OAuth 2 provider template

```
$ apic create:api --title "OAuth2 Provider" --template oauth2
Created oauth2-provider.yaml API definition [oauth2-provider:1.0.0]
```

Create an API using one of your templates

```
$ apic config:set --global template-path="/etc/templates"
$ ls /etc/templates
proxy.hbs staging.hbs
$ apic create:api --title "Proxy Provider" --template proxy
Created proxy-provider.yaml API definition [proxy-provider:1.0.0]
```

Create an API using your default template

```
$ apic config:set --global template-path="/etc/templates"
$ ls /etc/templates
proxy.hbs staging.hbs
$ apic config:set --global template-default-api=staging
$ apic create:api --title "Staging Provider"
Created staging-provider.yaml API definition [staging-provider:1.0.0]
```

### Options

<code>--api_type string</code>	The type of api (rest, wsdl-to-rest, or wsdl) (default "wsdl")
<code>--basepath string</code>	basepath value (default derived from name)
<code>--disable_ws_security</code>	Disable generation of WS-Security definitions in api
<code>--filename string</code>	filename (default derived from name)
<code>--gateway-type string</code>	The type of the gateway (datapower-gateway, datapower-api-gateway) (default "datapower-gateway")
<code>-h, --help</code>	Help for create:api
<code>--hostname string</code>	host value (default \$(catalog.host))
<code>-i, --interactive</code>	use interactive mode
<code>--name string</code>	x-ibm-name value (default derived from title)
<code>--product string</code>	generate a product definition referencing the API
<code>--schemes string</code>	list of schemes (valid options are http, https, ws and wss)
<code>--services string</code>	service names separated by space

<code>--target-url string</code>	target url
<code>--template string</code>	use a provider template (if empty defaults to apic template)
<code>--title string</code>	title value (required)
<code>-v, --version string</code>	version value (default "1.0.0")
<code>--wsdl string</code>	wsdl file to use as the source (required authentication via apic login)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic create:product

Create a product definition

### Synopsis

Create a product definition

```
apic create:product [flags]
```

### Examples

```
Create a product
$ apic create:product --title "Climb On"
Created climb-on.yaml product definition [climb-on:1.0.0]
Create a product interactively
$ apic create:product
? Title: Climb On
? Name: climb-on
? File: climb-on.yaml
? Template:
? API Files:
Created climb-on.yaml product definition [climb-on:1.0.0]
Create a product referencing existing APIs
$ apic create:product --title "Climb On" --apis "routes.yaml ascents.yaml"
Created climb-on.yaml product definition [climb-on:1.0.0]
Create an product using one of your templates
$ apic config:set --global template-path="/etc/templates"
$ ls /etc/templates
proxy-product.hbs staging-product.hbs
$ apic create:product --title "Proxy Product" --template proxy
Created proxy-product.yaml product definition [proxy-product:1.0.0]
Create a product using your default template
$ apic config:set --global template-path="/etc/templates"
$ ls /etc/templates
proxy-product.hbs staging-product.hbs
$ apic config:set --global template-default-product=staging
$ apic create:product --title "Staging Product"
Created staging-product.yaml product definition [staging-product:1.0.0]
```

### Options

<code>--apis string</code>	api file names separated by a space
<code>--filename string</code>	filename (default derived from name)
<code>--gateway-type string</code>	The type of the gateway (datapower-gateway, datapower-api-gateway) (default "datapower-gateway")
<code>-h, --help</code>	Help for create:product
<code>-i, --interactive</code>	use interactive mode
<code>--name string</code>	x-ibm-name value (default derived from title)
<code>--template string</code>	use a provider template (if empty defaults to apic template)
<code>--title string</code>	title value (required)
<code>-v, --version string</code>	version value (default "1.0.0")

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic credentials

Credentials operations

### Synopsis

---

Credentials operations

```
apic credentials [flags]
```

### Options

---

<code>-a, --app string</code>	Application name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for credentials
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:clear

Clear the Application Credential objects

### Synopsis

---

Clear the Application Credential objects

```
apic credentials:clear [flags]
```

### Options

---

<code>-a, --app string</code>	Application name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for credentials:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:create

Create a Application Credential object

### Synopsis

---

Create a Application Credential object

```
apic credentials:create [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:delete

Delete the Application Credential object by name or id

### Synopsis

---

Delete the Application Credential object by name or id

```
apic credentials:delete [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:delete
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:get

Get the Application Credential object by name or id

### Synopsis

---

Get the Application Credential object by name or id

```
apic credentials:get [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:get
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:list

List the Application Credential objects

### Synopsis

---

List the Application Credential objects

```
apic credentials:list [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic credentials:reset-client-secret

Reset the client secret

### Synopsis

---

Reset the client secret

```
apic credentials:reset-client-secret [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:reset-client-secret
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:reset

Reset the client id and client secret

### Synopsis

---

Reset the client id and client secret

```
apic credentials:reset [flags]
```

### Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:reset
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:update

Update the Application Credential object by name or id

## Synopsis

---

Update the Application Credential object by name or id

```
apic credentials:update [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:update
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic credentials:verify-client-secret

Verify the client secret

## Synopsis

---

Verify the client secret

```
apic credentials:verify-client-secret [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for credentials:verify-client-secret
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis

Draft Apis operations

## Synopsis

---

Draft Apis operations

```
apic draft-apis [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:clear-all

Clear all Draft API objects in all collections

## Synopsis

---

Clear all Draft API objects in all collections

```
apic draft-apis:clear-all [flags]
```

## Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for draft-apis:clear-all
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:clear

Clear the Draft API objects

## Synopsis

---

Clear the Draft API objects

```
apic draft-apis:clear [flags]
```

## Options

---

```

--confirm string      Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for draft-apis:clear
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)

```

## Options inherited from parent commands

```

--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic draft-apis:clone

Clone the draft-api objects

### Synopsis

Clone the draft-api objects

```
apic draft-apis:clone [flags]
```

### Options

```

--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for draft-apis:clone
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string   management server endpoint (required)

```

## Options inherited from parent commands

```

--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic draft-apis:create

Create a Draft API object

### Synopsis

Create a Draft API object

```
apic draft-apis:create [flags]
```

### Options

```

--api_type string     The type of api (rest, wsdl_to_rest, or wsdl)
--assembly_type string The type of the assembly to generate (rest_to_proxy)
--disable_ws_security Disable generation of WS-Security definitions in api
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway_type string The type of the gateway (datapower-gateway, datapower-api-gateway)
-h, --help           Help for draft-apis:create
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
--wsdl_service string Name of WSDL service to create the OpenAPI definition from

```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-apis:delete

Delete a Draft API

### Synopsis

---

Delete a Draft API

```
apic draft-apis:delete [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-apis:delete
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-apis:document

Get the Draft API document by name and version

### Synopsis

---

Get the Draft API document by name and version

```
apic draft-apis:document [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-apis:document
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:get

Get the Draft API object by name and version

### Synopsis

Get the Draft API object by name and version

```
apic draft-apis:get [flags]
```

### Options

```
--fields string  List of field names to return (default "add(wsd1,draft_api)")
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis:get
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:list

List the Draft API objects

### Synopsis

List the Draft API objects

```
apic draft-apis:list [flags]
```

### Options

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:list-all

List all Draft API objects in all collections

## Synopsis

---

List all Draft API objects in all collections

```
apic draft-apis:list-all [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis:list-all
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:update

Update the Draft API object by name and version

## Synopsis

---

Update the Draft API object by name and version

```
apic draft-apis:update [flags]
```

## Options

---

```
--api_type string  The type of api (rest, wsdl_to_rest, or wsdl)
--disable_ws_security Disable generation of WS-Security definitions in api
--format string    Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for draft-apis:update
--id              id
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
--wsdl_service string Name of WSDL service to create the OpenAPI definition from
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-apis:validate

Validate the draft api

## Synopsis

---

Validate the draft api

```
apic draft-apis:validate [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis:validate
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-apis:wsdl

Get the Draft API wsdl document by name and version

## Synopsis

---

Get the Draft API wsdl document by name and version

```
apic draft-apis:wsdl [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-apis:wsdl
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products

Draft Products operations

## Synopsis

---

Draft Products operations

```
apic draft-products [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-products
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
```



```
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:clear-all

Clear all Draft Product objects in all collections

## Synopsis

---

Clear all Draft Product objects in all collections

```
apic draft-products:clear-all [flags]
```

## Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for draft-products:clear-all
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:clear

Clear the Draft Product objects

## Synopsis

---

Clear the Draft Product objects

```
apic draft-products:clear [flags]
```

## Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for draft-products:clear
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products:clone

Clone the draft-product objects

### Synopsis

---

Clone the draft-product objects

```
apic draft-products:clone [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-products:clone
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products:create

Create a Draft Product object

### Synopsis

---

Create a Draft Product object

```
apic draft-products:create [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-products:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--product-only</code>	Upload only the product document
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:delete

Delete a Draft Product

### Synopsis

---

Delete a Draft Product

```
apic draft-products:delete [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-products:delete
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:document

Get the Draft Product document by name and version

### Synopsis

---

Get the Draft Product document by name and version

```
apic draft-products:document [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-products:document
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:get

Get the Draft Product object by name and version

### Synopsis

---

Get the Draft Product object by name and version

```
apic draft-products:get [flags]
```

## Options

---

```
--fields string  List of field names to return (default "add(draft_product,url)")
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-products:get
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:list-all

List all Draft Product objects in all collections

## Synopsis

---

List all Draft Product objects in all collections

```
apic draft-products:list-all [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-products:list-all
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic draft-products:list

List the Draft Product objects

## Synopsis

---

List the Draft Product objects

```
apic draft-products:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-products:list
```

```

--limit int32      Maximum number of items to return
--offset int32     Offset item number from list to begin return
-o, --org string   Organization name or id (required)
--output string    Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)

```

## Options inherited from parent commands

---

```

--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products:publish

Publish a draft product

### Synopsis

---

Publish a draft product

```
apic draft-products:publish [flags]
```

### Options

---

```

-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway_services string The list of gateway service names to support partial publishing
-h, --help            Help for draft-products:publish
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string    management server endpoint (required)
--space string        Space name or id (required)
--stage              stage

```

## Options inherited from parent commands

---

```

--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products:update

Update the Draft Product object by name and version

### Synopsis

---

Update the Draft Product object by name and version

```
apic draft-products:update [flags]
```

### Options

---

```

--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for draft-products:update
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)

```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic draft-products:validate

Validate the draft product

### Synopsis

---

Validate the draft product

```
apic draft-products:validate [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for draft-products:validate
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic drafts

Drafts operations

### Synopsis

---

Drafts operations

```
apic drafts [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for drafts
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic drafts:clear

Clear the Draft objects

### Synopsis

---

Clear the Draft objects

```
apic drafts:clear [flags]
```

### Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for drafts:clear
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic drafts:list

List the Draft objects

### Synopsis

---

List the Draft objects

```
apic drafts:list [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for drafts:list
--limit int32     Maximum number of items to return
--offset int32    Offset item number from list to begin return
-o, --org string  Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic entries

Entries operations

### Synopsis

---

Entries operations

```
apic entries [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for entries
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--truststore string  Truststore name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic entries:clear

Entries clear operations

### Synopsis

---

Entries clear operations

```
apic entries:clear [flags]
```

### Options

---

```
--confirm string     Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for entries:clear
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--truststore string  Truststore name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic entries:create



Entries create operations

## Synopsis

---

Entries create operations

```
apic entries:create [flags]
```

## Options

---

<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for entries:create
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope</code> string	scope
<code>-s, --server</code> string	management server endpoint (required)
<code>--truststore</code> string	Truststore name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic entries:delete

Entries delete operations

## Synopsis

---

Entries delete operations

```
apic entries:delete [flags]
```

## Options

---

<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for entries:delete
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope</code> string	scope
<code>-s, --server</code> string	management server endpoint (required)
<code>--truststore</code> string	Truststore name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic entries:get

Entries get operations

## Synopsis

---

Entries get operations

```
apic entries:get [flags]
```

## Options

---

```
--fields string      List of field names to return
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for entries:get
-o, --org string    Organization name or id (required)
--output string     Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string      scope
-s, --server string management server endpoint (required)
--truststore string Truststore name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic entries:list

Entries list operations

## Synopsis

---

Entries list operations

```
apic entries:list [flags]
```

## Options

---

```
--fields string      List of field names to return
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for entries:list
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
-o, --org string    Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--scope string      scope
-s, --server string management server endpoint (required)
--truststore string Truststore name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic entries:update

Entries update operations

## Synopsis

---

Entries update operations

```
apic entries:update [flags]
```

## Options

---

```
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for entries:update
```

-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--truststore string	Truststore name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions

Extensions operations

### Synopsis

Extensions operations

**apic extensions** [flags]

### Options

--availability-zone string	Availability Zone name or id (required)
-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string	Gateway Service name or id (required)
-h, --help	Help for extensions
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:clear-all

Extensions clear-all operations

### Synopsis

Extensions clear-all operations

**apic extensions:clear-all** [flags]

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to

```

yaml.
-h, --help                Help for extensions:clear-all
-o, --org string          Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string       management server endpoint (required)
--space string           Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:clear

Extensions clear operations

### Synopsis

Extensions clear operations

```
apic extensions:clear [flags]
```

### Options

```

-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--confirm string          Confirmation for critical updates (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                Help for extensions:clear
-o, --org string          Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string       management server endpoint (required)
--space string           Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:clone

Extensions clone operations

### Synopsis

Extensions clone operations

```
apic extensions:clone [flags]
```

### Options

```

--availability-zone string  Availability Zone name or id (required)
-c, --catalog string        Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string            List of field names to return
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.

```

```

--gateway-service string      Gateway Service name or id (required)
-h, --help                   Help for extensions:clone
--limit int32                Maximum number of items to return
--offset int32               Offset item number from list to begin return
-o, --org string              Organization name or id (required)
--output string               Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string                scope
-s, --server string           management server endpoint (required)
--space string                Space name or id (required)

```

## Options inherited from parent commands

---

```

--accept-license             Accept the license for API Connect
--debug                      Enable debug output
--debug-output string        Write debug output to file
--live-help                  Enable or disable tracking of limited usage information
-m, --mode string            Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic extensions:create

Extensions create operations

### Synopsis

---

Extensions create operations

```
apic extensions:create [flags]
```

### Options

---

```

-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string               Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                    Help for extensions:create
-o, --org string               Organization name or id (required)
--output string                Write file(s) to directory, instead of STDOUT (default "-")
--scope string                  scope
-s, --server string            management server endpoint (required)
--space string                  Space name or id (required)

```

## Options inherited from parent commands

---

```

--accept-license             Accept the license for API Connect
--debug                      Enable debug output
--debug-output string        Write debug output to file
--live-help                  Enable or disable tracking of limited usage information
-m, --mode string            Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic extensions:delete

Extensions delete operations

### Synopsis

---

Extensions delete operations

```
apic extensions:delete [flags]
```

### Options

---

```

-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string               Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                    Help for extensions:delete

```

--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:document

Extensions document operations

### Synopsis

Extensions document operations

```
apic extensions:document [flags]
```

### Options

--availability-zone string	Availability Zone name or id (required)
-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string	Gateway Service name or id (required)
-h, --help	Help for extensions:document
--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:get

Extensions get operations

### Synopsis

Extensions get operations

```
apic extensions:get [flags]
```

### Options

--availability-zone string	Availability Zone name or id (required)
-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to

```

yaml.
--gateway-service string      Gateway Service name or id (required)
-h, --help                    Help for extensions:get
--id                           id
-o, --org string               Organization name or id (required)
--output string                Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string                 scope
-s, --server string            management server endpoint (required)
--space string                 Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license              Accept the license for API Connect
--debug                       Enable debug output
--debug-output string         Write debug output to file
--live-help                   Enable or disable tracking of limited usage information
-m, --mode string             Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:list

Extensions list operations

### Synopsis

Extensions list operations

```
apic extensions:list [flags]
```

### Options

```

--availability-zone string    Availability Zone name or id (required)
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string               List of field names to return
--format string                Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
--gateway-service string      Gateway Service name or id (required)
-h, --help                    Help for extensions:list
--limit int32                 Maximum number of items to return
--offset int32                Offset item number from list to begin return
-o, --org string               Organization name or id (required)
--output string                Write file(s) to directory, instead of STDOUT (default "-")
--scope string                 scope
-s, --server string            management server endpoint (required)
--space string                 Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license              Accept the license for API Connect
--debug                       Enable debug output
--debug-output string         Write debug output to file
--live-help                   Enable or disable tracking of limited usage information
-m, --mode string             Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:list-all

Extensions list-all operations

### Synopsis

Extensions list-all operations

```
apic extensions:list-all [flags]
```

### Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service string</code>	Gateway Service name or id (required)
<code>-h, --help</code>	Help for extensions:list-all
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic extensions:update

Extensions update operations

### Synopsis

Extensions update operations

```
apic extensions:update [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for extensions:update
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic gateway-extensions

Gateway Extensions operations

### Synopsis

Gateway Extensions operations

```
apic gateway-extensions [flags]
```



## Options

---

```
-h, --help    Help for gateway-extensions
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-extensions:create

---

Gateway Extensions create operations

## Synopsis

---

Gateway Extensions create operations

```
apic gateway-extensions:create [flags]
```

## Options

---

```
--availability-zone string Availability Zone name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string  Gateway Service name or id (required)
-h, --help               Help for gateway-extensions:create
-o, --org string          Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-extensions:delete

---

Gateway Extensions delete operations

## Synopsis

---

Gateway Extensions delete operations

```
apic gateway-extensions:delete [flags]
```

## Options

---

```
--availability-zone string Availability Zone name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string  Gateway Service name or id (required)
-h, --help               Help for gateway-extensions:delete
-o, --org string          Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-extensions:get

Gateway Extensions get operations

### Synopsis

Gateway Extensions get operations

```
apic gateway-extensions:get [flags]
```

### Options

```
--availability-zone string  Availability Zone name or id (required)
--fields string             List of field names to return
--format string             Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string    Gateway Service name or id (required)
-h, --help                  Help for gateway-extensions:get
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string              scope
-s, --server string         management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-extensions:implementation

Get the Gateway Extension implementation document

### Synopsis

Get the Gateway Extension implementation document

```
apic gateway-extensions:implementation [flags]
```

### Options

```
--availability-zone string  Availability Zone name or id (required)
--format string             Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string    Gateway Service name or id (required)
-h, --help                  Help for gateway-extensions:implementation
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string         management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-extensions:update

Gateway Extensions update operations

### Synopsis

Gateway Extensions update operations

```
apic gateway-extensions:update [flags]
```

### Options

<code>--availability-zone</code> string	Availability Zone name or id (required)
<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service</code> string	Gateway Service name or id (required)
<code>-h, --help</code>	Help for gateway-extensions:update
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope</code> string	scope
<code>-s, --server</code> string	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-services

Gateway Services operations

### Synopsis

Gateway Services operations

```
apic gateway-services [flags]
```

### Options

<code>--availability-zone</code> string	Availability Zone name or id (required)
<code>--fields</code> string	List of field names to return
<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services
<code>--limit</code> int32	Maximum number of items to return
<code>--offset</code> int32	Offset item number from list to begin return
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope</code> string	scope
<code>-s, --server</code> string	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic gateway-services:clear

Clear the Gateway Service objects

### Synopsis

---

Clear the Gateway Service objects

```
apic gateway-services:clear [flags]
```

### Options

---

```
--availability-zone string  Availability Zone name or id (required)
--confirm string           Confirmation for critical updates (required)
--format string            Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                 Help for gateway-services:clear
-o, --org string           Organization name or id (required)
--output string            Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string        management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license          Accept the license for API Connect
--debug                   Enable debug output
--debug-output string     Write debug output to file
--live-help               Enable or disable tracking of limited usage information
-m, --mode string         Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic gateway-services:create

Create a Gateway Service object

### Synopsis

---

Create a Gateway Service object

```
apic gateway-services:create [flags]
```

### Options

---

```
--availability-zone string  Availability Zone name or id (required)
--format string            Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                 Help for gateway-services:create
-o, --org string           Organization name or id (required)
--output string            Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string        management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license          Accept the license for API Connect
--debug                   Enable debug output
--debug-output string     Write debug output to file
--live-help               Enable or disable tracking of limited usage information
-m, --mode string         Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic gateway-services:delete

Delete the Gateway Service object by name or id

### Synopsis

---

Delete the Gateway Service object by name or id

```
apic gateway-services:delete [flags]
```

## Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic gateway-services:get

Get the Gateway Service object by name or id

## Synopsis

Get the Gateway Service object by name or id

```
apic gateway-services:get [flags]
```

## Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic gateway-services:list

List the Gateway Service objects

## Synopsis

List the Gateway Service objects

```
apic gateway-services:list [flags]
```

## Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--fields string</code>	List of field names to return

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic gateway-services:reset-oauth-secret

Reset the oauth shared crypto material

### Synopsis

Reset the oauth shared crypto material

```
apic gateway-services:reset-oauth-secret [flags]
```

### Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services:reset-oauth-secret
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic gateway-services:update

Update the Gateway Service object by name or id

### Synopsis

Update the Gateway Service object by name or id

```
apic gateway-services:update [flags]
```

### Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for gateway-services:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policies

Global Policies operations

### Synopsis

Global Policies operations

```
apic global-policies [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for global-policies
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policies:clear

Global Policies clear operations

### Synopsis

Global Policies clear operations

```
apic global-policies:clear [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for global-policies:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:clear-all

Global Policies clear-all operations

### Synopsis

Global Policies clear-all operations

```
apic global-policies:clear-all [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for global-policies:clear-all
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:create

Global Policies create operations

### Synopsis

Global Policies create operations

```
apic global-policies:create [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for global-policies:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file



--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policies:delete

Global Policies delete operations

### Synopsis

Global Policies delete operations

```
apic global-policies:delete [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for global-policies:delete
--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policies:document

Global Policies document operations

### Synopsis

Global Policies document operations

```
apic global-policies:document [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for global-policies:document
--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:get

Global Policies get operations

### Synopsis

Global Policies get operations

```
apic global-policies:get [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for global-policies:get
--id string              id
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:list

Global Policies list operations

### Synopsis

Global Policies list operations

```
apic global-policies:list [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for global-policies:list
--limit int32            Maximum number of items to return
--offset int32           Offset item number from list to begin return
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
```

```
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:list-all

Global Policies list-all operations

### Synopsis

Global Policies list-all operations

```
apic global-policies:list-all [flags]
```

### Options

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string              List of field names to return
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policies:list-all
--limit int32               Maximum number of items to return
--offset int32              Offset item number from list to begin return
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policies:update

Global Policies update operations

### Synopsis

Global Policies update operations

```
apic global-policies:update [flags]
```

### Options

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policies:update
--id string                  id
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policy-posthooks

Global Policy Posthooks operations

### Synopsis

Global Policy Posthooks operations

```
apic global-policy-posthooks [flags]
```

### Options

```
-h, --help  Help for global-policy-posthooks
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policy-posthooks:create

Global Policy Posthooks create operations

### Synopsis

Global Policy Posthooks create operations

```
apic global-policy-posthooks:create [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for global-policy-posthooks:create
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic global-policy-posthooks:delete

Global Policy Posthooks delete operations

### Synopsis

---

Global Policy Posthooks delete operations

```
apic global-policy-posthooks:delete [flags]
```

### Options

---

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policy-posthooks:delete
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic global-policy-posthooks:get

Global Policy Posthooks get operations

### Synopsis

---

Global Policy Posthooks get operations

```
apic global-policy-posthooks:get [flags]
```

### Options

---

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string              List of field names to return
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policy-posthooks:get
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic global-policy-posthooks:update

## Synopsis

---

Global Policy Posthooks update operations

```
apic global-policy-posthooks:update [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help              Help for global-policy-posthooks:update
-o, --org string        Organization name or id (required)
--output string         Write file(s) to directory, instead of STDOUT (default "-")
--scope string          scope
-s, --server string     management server endpoint (required)
--space string          Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic global-policy-prehooks

Global Policy Prehooks operations

## Synopsis

---

Global Policy Prehooks operations

```
apic global-policy-prehooks [flags]
```

## Options

---

```
-h, --help  Help for global-policy-prehooks
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic global-policy-prehooks:create

Global Policy Prehooks create operations

## Synopsis

---

Global Policy Prehooks create operations

```
apic global-policy-prehooks:create [flags]
```

## Options

---

```

-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policy-prehooks:create
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license          Accept the license for API Connect
--debug                  Enable debug output
--debug-output string    Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string        Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policy-prehooks:delete

Global Policy Prehooks delete operations

### Synopsis

Global Policy Prehooks delete operations

```
apic global-policy-prehooks:delete [flags]
```

### Options

```

-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                  Help for global-policy-prehooks:delete
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
--space string              Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license          Accept the license for API Connect
--debug                  Enable debug output
--debug-output string    Write debug output to file
--live-help              Enable or disable tracking of limited usage information
-m, --mode string        Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policy-prehooks:get

Global Policy Prehooks get operations

### Synopsis

Global Policy Prehooks get operations

```
apic global-policy-prehooks:get [flags]
```

### Options

```

-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string              List of field names to return
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to

```

```

yaml.
-h, --help                Help for global-policy-prehooks:get
-o, --org string          Organization name or id (required)
--output string          Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string           scope
-s, --server string       management server endpoint (required)
--space string           Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic global-policy-prehooks:update

Global Policy Prehooks update operations

### Synopsis

Global Policy Prehooks update operations

```
apic global-policy-prehooks:update [flags]
```

### Options

```

-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                Help for global-policy-prehooks:update
-o, --org string          Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string       management server endpoint (required)
--space string           Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic groups

Groups operations

### Synopsis

Groups operations

```
apic groups [flags]
```

### Options

```

-c, --catalog string      Catalog name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for groups
--limit int32           Maximum number of items to return
--offset int32          Offset item number from list to begin return
-o, --org string          Organization name or id (required)

```



```
--output string Write file(s) to directory, instead of STDOUT (default "-")
--scope string scope
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic groups:clear

Groups clear operations

## Synopsis

---

Groups clear operations

```
apic groups:clear [flags]
```

## Options

---

```
-c, --catalog string Catalog name or id (required)
--confirm string Confirmation for critical updates (required)
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for groups:clear
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
--scope string scope
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic groups:create

Groups create operations

## Synopsis

---

Groups create operations

```
apic groups:create [flags]
```

## Options

---

```
-c, --catalog string Catalog name or id (required)
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for groups:create
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
--scope string scope
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic groups:delete

Groups delete operations

### Synopsis

Groups delete operations

```
apic groups:delete [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for groups:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic groups:get

Groups get operations

### Synopsis

Groups get operations

```
apic groups:get [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for groups:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic groups:list

Groups list operations

### Synopsis

---

Groups list operations

```
apic groups:list [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for groups:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic groups:update

Groups update operations

### Synopsis

---

Groups update operations

```
apic groups:update [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for groups:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic identity-providers

Identity Providers operations

### Synopsis

---

Identity Providers operations

```
apic identity-providers [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for identity-providers
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
--scope string   scope
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic identity-providers:list

Identity Providers list operations

### Synopsis

---

Identity Providers list operations

```
apic identity-providers:list [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for identity-providers:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
--scope string   scope
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic integrations

Integrations operations

## Synopsis

---

Integrations operations

```
apic integrations [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for integrations
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic integrations:clear

Integrations clear operations

## Synopsis

---

Integrations clear operations

```
apic integrations:clear [flags]
```

## Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string    Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for integrations:clear
--output string    Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic integrations:create

Integrations create operations

## Synopsis

---

Integrations create operations

```
apic integrations:create [flags]
```

## Options

---

```
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for integrations:create
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic integrations:delete

Integrations delete operations

### Synopsis

---

Integrations delete operations

```
apic integrations:delete [flags]
```

### Options

---

```
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for integrations:delete
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic integrations:get

Integrations get operations

### Synopsis

---

Integrations get operations

```
apic integrations:get [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for integrations:get
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string   management server endpoint (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic integrations:list

Integrations list operations

### Synopsis

Integrations list operations

```
apic integrations:list [flags]
```

### Options

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for integrations:list
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
--subcollection string subcollection
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic integrations:list-all

List all Integration objects in all collections

### Synopsis

List all Integration objects in all collections

```
apic integrations:list-all [flags]
```

### Options

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for integrations:list-all
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic integrations:update

Integrations update operations

### Synopsis

---

Integrations update operations

```
apic integrations:update [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for integrations:update
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--subcollection string</code>	subcollection

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic invitations

Invitations operations

### Synopsis

---

Invitations operations

```
apic invitations [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for invitations
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic invitations:clear

Invitations clear operations



## Synopsis

---

Invitations clear operations

```
apic invitations:clear [flags]
```

## Options

---

-c, --catalog string	Catalog name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for invitations:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic invitations:create

Invitations create operations

## Synopsis

---

Invitations create operations

```
apic invitations:create [flags]
```

## Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for invitations:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic invitations:delete

Invitations delete operations

## Synopsis

---

Invitations delete operations

```
apic invitations:delete [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for invitations:delete
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic invitations:get

Invitations get operations

## Synopsis

---

Invitations get operations

```
apic invitations:get [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for invitations:get
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic invitations:list

Invitations list operations

## Synopsis

---

Invitations list operations

```
apic invitations:list [flags]
```

## Options

---

```

-c, --catalog string      Catalog name or id (required)
--fields string          List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for invitations:list
--limit int32            Maximum number of items to return
--offset int32           Offset item number from list to begin return
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
--space-initiated       space-initiated

```

## Options inherited from parent commands

---

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic invitations:update

Invitations update operations

### Synopsis

---

Invitations update operations

```
apic invitations:update [flags]
```

### Options

---

```

-c, --catalog string      Catalog name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for invitations:update
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
--space-initiated       space-initiated

```

## Options inherited from parent commands

---

```

--accept-license        Accept the license for API Connect
--debug                Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic jobs

Jobs operations

### Synopsis

---

Jobs operations

```
apic jobs [flags]
```

### Options

---

```
-h, --help Help for jobs
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic configured-billings:clear

Clear the Configured Billing objects

### Synopsis

---

Clear the Configured Billing objects

```
apic configured-billings:clear [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for configured-billings:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic jobs:delete

Delete the Job object by name or id

### Synopsis

---

Delete the Job object by name or id

```
apic jobs:delete [flags]
```

### Options

---

<code>--billing string</code>	Billing name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for jobs:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic jobs:get

Get the Job object by name or id

### Synopsis

---

Get the Job object by name or id

```
apic jobs:get [flags]
```

### Options

---

```
--billing string  Billing name or id (required)
--fields string   List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for jobs:get
-o, --org string   Organization name or id (required)
--output string   Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic jobs:list

List the Job objects

### Synopsis

---

List the Job objects

```
apic jobs:list [flags]
```

### Options

---

```
--billing string  Billing name or id (required)
--fields string   List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help        Help for jobs:list
--limit int32     Maximum number of items to return
--offset int32    Offset item number from list to begin return
-o, --org string   Organization name or id (required)
--output string   Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic jobs:retry

Re-attempt blocked or failed job

### Synopsis

---

Re-attempt blocked or failed job

```
apic jobs:retry [flags]
```

### Options

---

```
--billing string      Billing name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for jobs:retry
--operation string    Operation to perform on the user registry (get_base_dn_list, validate_password) (required)
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic keystores

Keystores operations

### Synopsis

---

Keystores operations

```
apic keystores [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for keystores
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic keystores:clear

Clear the Keystore objects

## Synopsis

---

Clear the Keystore objects

```
apic keystores:clear [flags]
```

## Options

---

<code>--confirm</code>	string	Confirmation for critical updates (required)
<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for keystores:clear
<code>-o, --org</code>	string	Organization name or id (required)
<code>--output</code>	string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code>	string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>		Accept the license for API Connect
<code>--debug</code>		Enable debug output
<code>--debug-output</code>	string	Write debug output to file
<code>--live-help</code>		Enable or disable tracking of limited usage information
<code>-m, --mode</code>	string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic keystores:create

Create a Keystore object

## Synopsis

---

Create a Keystore object

```
apic keystores:create [flags]
```

## Options

---

<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for keystores:create
<code>-o, --org</code>	string	Organization name or id (required)
<code>--output</code>	string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code>	string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>		Accept the license for API Connect
<code>--debug</code>		Enable debug output
<code>--debug-output</code>	string	Write debug output to file
<code>--live-help</code>		Enable or disable tracking of limited usage information
<code>-m, --mode</code>	string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic keystores:delete

Delete the Keystore object by name or id

## Synopsis

---

Delete the Keystore object by name or id

```
apic keystores:delete [flags]
```

## Options

---

<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for keystores:delete

```
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic keystores:get

Get the Keystore object by name or id

### Synopsis

---

Get the Keystore object by name or id

```
apic keystores:get [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for keystores:get
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic keystores:list

List the Keystore objects

### Synopsis

---

List the Keystore objects

```
apic keystores:list [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for keystores:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
```



```
--debug-output string  Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string      Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic keystores:update

Update the Keystore object by name or id

### Synopsis

---

Update the Keystore object by name or id

```
apic keystores:update [flags]
```

### Options

---

```
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for keystores:update
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic lb4

create and manage LoopBack applications

### Synopsis

---

create and manage LoopBack applications

```
apic lb4 [flags]
```

### Options

---

```
-h, --help  Help for lb4
```

### Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic licenses

Review the license for API Connect

## Synopsis

---

Review the license for API Connect

```
apic licenses [flags]
```

## Options

---

```
-h, --help           Help for licenses
--non-ibm-license    Display the non IBM license files.
--notices            Display the notices file
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic log-spec

Log Spec operations

## Synopsis

---

Log Spec operations

```
apic log-spec [flags]
```

## Options

---

```
-h, --help   Help for log-spec
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic log-spec:get

Get the Log Spec object

## Synopsis

---

Get the Log Spec object

```
apic log-spec:get [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for log-spec:get
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic log-spec:update

Update the Log Spec object

### Synopsis

---

Update the Log Spec object

```
apic log-spec:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for log-spec:update
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic login

Log in to an IBM API Connect cloud

### Synopsis

---

Log in to an IBM API Connect cloud

```
apic login [flags]
```

### Examples

---

```
Interactive login
$ apic login
Enter your API Connect credentials
? Server: mgmthost.com
? Realm: company/realm
? Username: tommy
? Password: password
Logged into mgmthost.com successfully
```

```
Non-interactive login
$ apic login --username tommy --password password --server mgmthost.com --realm company/realm
Logged into mgmthost.com successfully
```

### Options

---

```
-h, --help      Help for login
-p, --password string password
-r, --realm string realm
-s, --server string management server endpoint
-u, --username string user name
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic logout

Log out of an IBM API Connect cloud

## Synopsis

---

Log out of an IBM API Connect cloud

```
apic logout [flags]
```

## Examples

---

```
Clear the local authentication credentials for mgmthost.com
$ apic logout --server mgmthost.com
Logged out of server mgmthost.com
```

## Options

---

<code>-h, --help</code>	Help for logout
<code>-s, --server string</code>	management server endpoint

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic mail-servers

Mail Servers operations

## Synopsis

---

Mail Servers operations

```
apic mail-servers [flags]
```

## Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for mail-servers
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic mail-servers:clear

Clear the Mail Server objects

### Synopsis

Clear the Mail Server objects

```
apic mail-servers:clear [flags]
```

### Options

```
--confirm string Confirmation for critical updates (required)
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for mail-servers:clear
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic mail-servers:create

Create a Mail Server object

### Synopsis

Create a Mail Server object

```
apic mail-servers:create [flags]
```

### Options

```
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for mail-servers:create
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic mail-servers:delete

Delete the Mail Server object by name or id

### Synopsis

---

Delete the Mail Server object by name or id

```
apic mail-servers:delete [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for mail-servers:delete  
-o, --org string Organization name or id (required)  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic mail-servers:get

Get the Mail Server object by name or id

### Synopsis

---

Get the Mail Server object by name or id

```
apic mail-servers:get [flags]
```

### Options

---

```
--fields string  List of field names to return  
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for mail-servers:get  
-o, --org string Organization name or id (required)  
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)  
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic mail-servers:list

List the Mail Server objects

### Synopsis

---

List the Mail Server objects

```
apic mail-servers:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for mail-servers:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic mail-servers:test-connection

Test a Mail Server connection

## Synopsis

---

Test a Mail Server connection

```
apic mail-servers:test-connection [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for mail-servers:test-connection
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
--test-config-only test-config-only
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic mail-servers:update

Update the Mail Server object by name or id

## Synopsis

---

Update the Mail Server object by name or id

```
apic mail-servers:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for mail-servers:update
-o, --org string Organization name or id (required)
```

```
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic me

Me operations

## Synopsis

---

Me operations

```
apic me [flags]
```

## Options

---

```
-h, --help Help for me
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic me:change-password

Change my password

## Synopsis

---

Change my password

```
apic me:change-password [flags]
```

## Options

---

```
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for me:change-password
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic me:delete

Delete the Me object

### Synopsis

---

Delete the Me object

```
apic me:delete [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for me:delete
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic me:get

Get the Me object

### Synopsis

---

Get the Me object

```
apic me:get [flags]
```

### Options

---

```
--context string  user's login context admin/manager
--fields string   List of field names to return
--format string   Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for me:get
--output string   Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic me:update

Update the Me object

### Synopsis

---

Update the Me object

```
apic me:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for me:update
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic member-invitations

Member Invitations operations

## Synopsis

---

Member Invitations operations

```
apic member-invitations [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for member-invitations
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated     space-initiated
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic member-invitations:clear

Member Invitations clear operations

## Synopsis

---

Member Invitations clear operations

```
apic member-invitations:clear [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string     Confirmation for critical updates (required)
--consumer-org string Consumer Organization name or id (required)
```

--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for member-invitations:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic member-invitations:create

Member Invitations create operations

### Synopsis

Member Invitations create operations

```
apic member-invitations:create [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for member-invitations:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic member-invitations:delete

Member Invitations delete operations

### Synopsis

Member Invitations delete operations

```
apic member-invitations:delete [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for member-invitations:delete
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")

--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic member-invitations:get

Member Invitations get operations

### Synopsis

Member Invitations get operations

```
apic member-invitations:get [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for member-invitations:get
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic member-invitations:list

Member Invitations list operations

### Synopsis

Member Invitations list operations

```
apic member-invitations:list [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for member-invitations:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")

```
--scope string      scope
-s, --server string management server endpoint (required)
--space string      Space name or id (required)
--space-initiated   space-initiated
```

## Options inherited from parent commands

---

```
--accept-license   Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic member-invitations:update

Member Invitations update operations

### Synopsis

---

Member Invitations update operations

```
apic member-invitations:update [flags]
```

### Options

---

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string     Consumer Organization name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for member-invitations:update
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated         space-initiated
```

## Options inherited from parent commands

---

```
--accept-license   Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic members

Members operations

### Synopsis

---

Members operations

```
apic members [flags]
```

### Options

---

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string     Consumer Organization name or id (required)
--fields string           List of field names to return
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for members
--limit int32             Maximum number of items to return
--offset int32            Offset item number from list to begin return
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
```

```
-s, --server string      management server endpoint (required)
--space string          Space name or id (required)
--space-initiated      space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic members:clear

Members clear operations

## Synopsis

---

Members clear operations

```
apic members:clear [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--confirm string          Confirmation for critical updates (required)
--consumer-org string     Consumer Organization name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for members:clear
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated        space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic members:create

Members create operations

## Synopsis

---

Members create operations

```
apic members:create [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string     Consumer Organization name or id (required)
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for members:create
-o, --org string           Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
--space-initiated        space-initiated
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic members:delete

Members delete operations

### Synopsis

---

Members delete operations

```
apic members:delete [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for members:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic members:get

Members get operations

### Synopsis

---

Members get operations

```
apic members:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for members:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic members:list

Members list operations

### Synopsis

Members list operations

```
apic members:list [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for members:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated     space-initiated
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic members:update

Members update operations

### Synopsis

Members update operations

```
apic members:update [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for members:update
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated     space-initiated
```

### Options inherited from parent commands



```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic notification-templates

Notification Templates operations

### Synopsis

Notification Templates operations

```
apic notification-templates [flags]
```

### Options

```
-c, --catalog string   Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for notification-templates
--limit int32          Maximum number of items to return
--offset int32         Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic notification-templates:get

Notification Templates get operations

### Synopsis

Notification Templates get operations

```
apic notification-templates:get [flags]
```

### Options

```
-c, --catalog string   Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for notification-templates:get
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--subcollection string subcollection
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
```

<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic notification-templates:list

Notification Templates list operations

### Synopsis

Notification Templates list operations

```
apic notification-templates:list [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for notification-templates:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--subcollection string</code>	subcollection

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic notification-templates:list-all

Notification Templates list-all operations

### Synopsis

Notification Templates list-all operations

```
apic notification-templates:list-all [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for notification-templates:list-all
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic notification-templates:update

Notification Templates update operations

### Synopsis

Notification Templates update operations

```
apic notification-templates:update [flags]
```

### Options

```
-c, --catalog string      Catalog name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for notification-templates:update
-o, --org string          Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string       management server endpoint (required)
--space string           Space name or id (required)
--subcollection string   subcollection
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic oauth-providers

Oauth Providers operations

### Synopsis

Oauth Providers operations

```
apic oauth-providers [flags]
```

### Options

```
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for oauth-providers
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic oauth-providers:clear

Clear the OAuth Provider objects

### Synopsis

---

Clear the OAuth Provider objects

```
apic oauth-providers:clear [flags]
```

### Options

---

<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for oauth-providers:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic oauth-providers:create

Create a OAuth Provider object

### Synopsis

---

Create a OAuth Provider object

```
apic oauth-providers:create [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for oauth-providers:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic oauth-providers:delete

Delete the OAuth Provider object by name or id

## Synopsis

---

Delete the Oauth Provider object by name or id

```
apic oauth-providers:delete [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for oauth-providers:delete  
-o, --org string Organization name or id (required)  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic oauth-providers:get

Get the Oauth Provider object by name or id

## Synopsis

---

Get the Oauth Provider object by name or id

```
apic oauth-providers:get [flags]
```

## Options

---

```
--fields string  List of field names to return  
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for oauth-providers:get  
-o, --org string Organization name or id (required)  
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug           Enable debug output  
--debug-output string Write debug output to file  
--live-help       Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic oauth-providers:list

List the Oauth Provider objects

## Synopsis

---

List the Oauth Provider objects

```
apic oauth-providers:list [flags]
```

## Options

---

```
--fields string  List of field names to return  
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
```

```
-h, --help           Help for oauth-providers:list
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic oauth-providers:update

Update the OAuth Provider object by name or id

### Synopsis

---

Update the OAuth Provider object by name or id

```
apic oauth-providers:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for oauth-providers:update
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic org-settings

Org Settings operations

### Synopsis

---

Org Settings operations

```
apic org-settings [flags]
```

### Options

---

```
-h, --help  Help for org-settings
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic org-settings:get

Get the Organization Setting object

### Synopsis

---

Get the Organization Setting object

```
apic org-settings:get [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for org-settings:get
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic org-settings:update

Update the Organization Setting object

### Synopsis

---

Update the Organization Setting object

```
apic org-settings:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for org-settings:update
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic orgs

## Synopsis

---

Orgs operations

```
apic orgs [flags]
```

## Options

---

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for orgs
--limit int32        Maximum number of items to return
--my                 my
--offset int32       Offset item number from list to begin return
--org_type string    Type of orgs to return
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic orgs:clear

Clear the Organization objects

## Synopsis

---

Clear the Organization objects

```
apic orgs:clear [flags]
```

## Options

---

```
--confirm string      Confirmation for critical updates (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for orgs:clear
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license     Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic orgs:create

Create an Organization object

## Synopsis

---

Create an Organization object

```
apic orgs:create [flags]
```



## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for orgs:create
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic orgs:delete

Delete the Organization object by name or id

## Synopsis

---

Delete the Organization object by name or id

```
apic orgs:delete [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for orgs:delete
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic orgs:get

Get the Organization object by name or id

## Synopsis

---

Get the Organization object by name or id

```
apic orgs:get [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for orgs:get
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic orgs:list

List the Organization objects

### Synopsis

List the Organization objects

```
apic orgs:list [flags]
```

### Options

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for orgs:list
--limit int32    Maximum number of items to return
--my            my
--offset int32   Offset item number from list to begin return
--org_type string Type of orgs to return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic orgs:transfer-owner

Transfer owner to an associate

### Synopsis

Transfer owner to an associate

```
apic orgs:transfer-owner [flags]
```

### Options

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for orgs:transfer-owner
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic orgs:update

Update the Organization object by name or id

### Synopsis

---

Update the Organization object by name or id

```
apic orgs:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for orgs:update  
--output string  Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic payment-methods

Payment Methods operations

### Synopsis

---

Payment Methods operations

```
apic payment-methods [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)  
--consumer-org string Consumer Organization name or id (required)  
--fields string      List of field names to return  
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help          Help for payment-methods  
--limit int32        Maximum number of items to return  
--offset int32       Offset item number from list to begin return  
-o, --org string     Organization name or id (required)  
--output string      Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string  management server endpoint (required)  
--space string       Space name or id (required)  
--space-initiated    space-initiated
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic payment-methods:create

Create a Payment Method object

## Synopsis

---

Create a Payment Method object

```
apic payment-methods:create [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for payment-methods:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic payment-methods:delete

Delete the Payment Method object by name or id

## Synopsis

---

Delete the Payment Method object by name or id

```
apic payment-methods:delete [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for payment-methods:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic payment-methods:get

Get the Payment Method object by name or id

## Synopsis

---

Get the Payment Method object by name or id

```
apic payment-methods:get [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for payment-methods:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic payment-methods:list

List the Payment Method objects

## Synopsis

---

List the Payment Method objects

```
apic payment-methods:list [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for payment-methods:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic payment-methods:update

Update the Payment Method object by name or id

## Synopsis

---

Update the Payment Method object by name or id

```
apic payment-methods:update [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--consumer-org string    Consumer Organization name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help               Help for payment-methods:update
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
--space-initiated        space-initiated
```

## Options inherited from parent commands

---

```
--accept-license        Accept the license for API Connect
--debug                 Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic permissions

Permissions operations

## Synopsis

---

Permissions operations

```
apic permissions [flags]
```

## Options

---

```
--fields string         List of field names to return
--format string         Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help              Help for permissions
--limit int32           Maximum number of items to return
--offset int32          Offset item number from list to begin return
--output string         Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string     management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license        Accept the license for API Connect
--debug                 Enable debug output
--debug-output string   Write debug output to file
--live-help             Enable or disable tracking of limited usage information
-m, --mode string       Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic permissions:get

Permissions get operations

## Synopsis

---

Permissions get operations

```
apic permissions:get [flags]
```

## Options

---

```
--fields string         List of field names to return
--format string         Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help              Help for permissions:get
--output string         Write file(s) to directory, use - for STDOUT. (default: cwd)
```

```
-s, --server string      management server endpoint (required)
--subcollection string  subcollection
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic permissions:list

Permissions list operations

### Synopsis

---

Permissions list operations

```
apic permissions:list [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for permissions:list
--limit int32        Maximum number of items to return
--my                 my
--offset int32       Offset item number from list to begin return
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic permissions:list-all

List all Permission objects in all collections

### Synopsis

---

List all Permission objects in all collections

```
apic permissions:list-all [flags]
```

### Options

---

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for permissions:list-all
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic policies

Policies operations

### Synopsis

Policies operations

`apic policies` [flags]

### Options

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service string</code>	Gateway Service name or id (required)
<code>-h, --help</code>	Help for policies
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic policies:clear

Policies clear operations

### Synopsis

Policies clear operations

`apic policies:clear` [flags]

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for policies:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)



## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:clear-all

Policies clear-all operations

### Synopsis

---

Policies clear-all operations

```
apic policies:clear-all [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for policies:clear-all
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:clone

Policies clone operations

### Synopsis

---

Policies clone operations

```
apic policies:clone [flags]
```

### Options

---

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service string</code>	Gateway Service name or id (required)
<code>-h, --help</code>	Help for policies:clone
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope

```
-s, --server string      management server endpoint (required)
--space string          Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:create

Policies create operations

## Synopsis

---

Policies create operations

```
apic policies:create [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for policies:create
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--policy-file string     Policy document override
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:delete

Policies delete operations

## Synopsis

---

Policies delete operations

```
apic policies:delete [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for policies:delete
--id string              id
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, instead of STDOUT (default "-")
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:document

Policies document operations

### Synopsis

---

Policies document operations

```
apic policies:document [flags]
```

### Options

---

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service string</code>	Gateway Service name or id (required)
<code>-h, --help</code>	Help for policies:document
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic policies:get

Policies get operations

### Synopsis

---

Policies get operations

```
apic policies:get [flags]
```

### Options

---

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return (default "add(policy,implementation)")
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>--gateway-service string</code>	Gateway Service name or id (required)
<code>-h, --help</code>	Help for policies:get
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope

```
-s, --server string      management server endpoint (required)
--space string          Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic policies:implementation

Policies implementation operations

### Synopsis

---

Policies implementation operations

```
apic policies:implementation [flags]
```

### Options

---

```
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help               Help for policies:implementation
--id                     id
-o, --org string         Organization name or id (required)
--output string          Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string           scope
-s, --server string      management server endpoint (required)
--space string           Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic policies:list

Policies list operations

### Synopsis

---

Policies list operations

```
apic policies:list [flags]
```

### Options

---

```
--availability-zone string Availability Zone name or id (required)
-c, --catalog string      Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--fields string           List of field names to return
--format string          Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
--gateway-service string Gateway Service name or id (required)
-h, --help               Help for policies:list
--limit int32            Maximum number of items to return
--offset int32           Offset item number from list to begin return
-o, --org string         Organization name or id (required)
```

--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic policies:list-all

Policies list-all operations

### Synopsis

Policies list-all operations

```
apic policies:list-all [flags]
```

### Options

--availability-zone string	Availability Zone name or id (required)
-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway-service string	Gateway Service name or id (required)
-h, --help	Help for policies:list-all
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic policies:update

Policies update operations

### Synopsis

Policies update operations

```
apic policies:update [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for policies:update

--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic portal-services

Portal Services operations

### Synopsis

Portal Services operations

```
apic portal-services [flags]
```

### Options

--availability-zone string	Availability Zone name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for portal-services
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic portal-services:clear

Clear the Portal Service objects

### Synopsis

Clear the Portal Service objects

```
apic portal-services:clear [flags]
```

### Options

--availability-zone string	Availability Zone name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for portal-services:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic portal-services:create

Create a Portal Service object

### Synopsis

---

Create a Portal Service object

```
apic portal-services:create [flags]
```

### Options

---

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for portal-services:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic portal-services:delete

Delete the Portal Service object by name or id

### Synopsis

---

Delete the Portal Service object by name or id

```
apic portal-services:delete [flags]
```

### Options

---

<code>--availability-zone string</code>	Availability Zone name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for portal-services:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic portal-services:get

Get the Portal Service object by name or id

### Synopsis

Get the Portal Service object by name or id

```
apic portal-services:get [flags]
```

### Options

```
--availability-zone string  Availability Zone name or id (required)
--fields string             List of field names to return
--format string             Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                  Help for portal-services:get
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string              scope
-s, --server string         management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license           Accept the license for API Connect
--debug                    Enable debug output
--debug-output string      Write debug output to file
--live-help                Enable or disable tracking of limited usage information
-m, --mode string          Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic portal-services:list

List the Portal Service objects

### Synopsis

List the Portal Service objects

```
apic portal-services:list [flags]
```

### Options

```
--availability-zone string  Availability Zone name or id (required)
--fields string             List of field names to return
--format string             Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                  Help for portal-services:list
--limit int32               Maximum number of items to return
--offset int32              Offset item number from list to begin return
-o, --org string            Organization name or id (required)
--output string             Write file(s) to directory, instead of STDOUT (default "-")
--scope string              scope
-s, --server string         management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license           Accept the license for API Connect
--debug                    Enable debug output
--debug-output string      Write debug output to file
--live-help                Enable or disable tracking of limited usage information
-m, --mode string          Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic portal-services:update

Update the Portal Service object by name or id

### Synopsis

---

Update the Portal Service object by name or id

```
apic portal-services:update [flags]
```

### Options

---

<code>--availability-zone</code> string	Availability Zone name or id (required)
<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for portal-services:update
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code> string	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic portal-services:update-credentials

Update the Portal Service configuration

### Synopsis

---

Update the Portal Service configuration

```
apic portal-services:update-credentials [flags]
```

### Options

---

<code>--availability-zone</code> string	Availability Zone name or id (required)
<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for portal-services:update-credentials
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code> string	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic primary-events

Primary Events operations

### Synopsis

---

Primary Events operations

apic primary-events [flags]

## Options

```
--availability-zone string      Availability Zone name or id (required)
-c, --catalog string           Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string                List of field names to return
--format string                 Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
--gateway-service string       Gateway Service name or id (required)
-h, --help                     Help for primary-events
--limit int32                  Maximum number of items to return
--offset int32                 Offset item number from list to begin return
-o, --org string                Organization name or id (required)
--output string                 Write file(s) to directory, instead of STDOUT (default "-")
--portal-service string         Portal Service name or id (required)
--scope string                  scope
-s, --server string             management server endpoint (required)
--state string                  State for a webhook event in subscriber queue
```

## Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug                Enable debug output
--debug-output string  Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string      Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic primary-events:get

Primary Events get operations

## Synopsis

Primary Events get operations

apic primary-events:get [flags]

## Options

```
--availability-zone string      Availability Zone name or id (required)
-c, --catalog string           Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string                List of field names to return
--format string                 Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
--gateway-service string       Gateway Service name or id (required)
-h, --help                     Help for primary-events:get
-o, --org string                Organization name or id (required)
--output string                 Write file(s) to directory, use - for STDOUT. (default: cwd)
--portal-service string         Portal Service name or id (required)
--scope string                  scope
-s, --server string             management server endpoint (required)
```

## Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug                Enable debug output
--debug-output string  Write debug output to file
--live-help            Enable or disable tracking of limited usage information
-m, --mode string      Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic primary-events:list

Primary Events list operations

## Synopsis

---

Primary Events list operations

```
apic primary-events:list [flags]
```

## Options

---

```
--availability-zone string      Availability Zone name or id (required)
-c, --catalog string           Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string                List of field names to return
--format string                Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
--gateway-service string       Gateway Service name or id (required)
-h, --help                     Help for primary-events:list
--limit int32                  Maximum number of items to return
--offset int32                 Offset item number from list to begin return
-o, --org string               Organization name or id (required)
--output string                Write file(s) to directory, instead of STDOUT (default "-")
--portal-service string        Portal Service name or id (required)
--scope string                 scope
-s, --server string            management server endpoint (required)
--state string                 State for a webhook event in subscriber queue
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products

Products operations

## Synopsis

---

Products operations

```
apic products [flags]
```

## Options

---

```
-c, --catalog string      Catalog name or id (required)
--fields string          List of field names to return
--format string           Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help                Help for products
--limit int32             Maximum number of items to return
--offset int32            Offset item number from list to begin return
-o, --org string          Organization name or id (required)
--output string           Write file(s) to directory, instead of STDOUT (default "-")
--scope string            scope
-s, --server string       management server endpoint (required)
--space string            Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products:clear

## Synopsis

---

Products clear operations

```
apic products:clear [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string     Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:clear
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products:clear-all

Products clear-all operations

## Synopsis

---

Products clear-all operations

```
apic products:clear-all [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string     Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:clear-all
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products:clone

Products clone operations

## Synopsis

---

Products clone operations

```
apic products:clone [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:clone
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:delete

Products delete operations

## Synopsis

---

Products delete operations

```
apic products:delete [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:delete
--id string          id
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:document

Products document operations

## Synopsis

---

Products document operations

```
apic products:document [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for products:document
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:execute-migration-target

Products execute-migration-target operations

### Synopsis

---

Products execute-migration-target operations

```
apic products:execute-migration-target [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for products:execute-migration-target
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:get

Products get operations

### Synopsis

---

Products get operations

```
apic products:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return (default "add(product)")

--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for products:get
--id	id
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic products:list

Products list operations

### Synopsis

Products list operations

```
apic products:list [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for products:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic products:list-all

Products list-all operations

### Synopsis

Products list-all operations

```
apic products:list-all [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for products:list-all
--limit int32	Maximum number of items to return

```

--offset int32      Offset item number from list to begin return
-o, --org string    Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--scope string      scope
-s, --server string management server endpoint (required)
--space string      Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic products:migrate-subscriptions

Products migrate-subscriptions operations

### Synopsis

Products migrate-subscriptions operations

```
apic products:migrate-subscriptions [flags]
```

### Options

```

-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for products:migrate-subscriptions
--id                  id
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)

```

## Options inherited from parent commands

```

--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic products:publish

Publish the product

### Synopsis

Publish the product

```
apic products:publish [flags]
```

### Options

```

-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
--gateway_services string The list of gateway service names to support partial publishing
-h, --help            Help for products:publish
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)

```



```
--space string      Space name or id (required)
--stage             stage
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:replace

Products replace operations

### Synopsis

---

Products replace operations

```
apic products:replace [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for products:replace
--id                  id
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic products:set-migration-target

Products set-migration-target operations

### Synopsis

---

Products set-migration-target operations

```
apic products:set-migration-target [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for products:set-migration-target
--id                  id
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products:supersede

Products supersede operations

### Synopsis

Products supersede operations

```
apic products:supersede [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:supersede
--id                id
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--scope string      scope
-s, --server string  management server endpoint (required)
--space string      Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic products:update

Products update operations

### Synopsis

Products update operations

```
apic products:update [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for products:update
--id                id
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--scope string      scope
-s, --server string  management server endpoint (required)
--space string      Space name or id (required)
```

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic properties

Properties operations

### Synopsis

---

Properties operations

```
apic properties [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for properties
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic properties:clear

Properties clear operations

### Synopsis

---

Properties clear operations

```
apic properties:clear [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--confirm string      Confirmation for critical updates (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for properties:clear
-o, --org string      Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic properties:create

Properties create operations

### Synopsis

---

Properties create operations

```
apic properties:create [flags]
```

### Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for properties:create
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic properties:delete

Properties delete operations

### Synopsis

---

Properties delete operations

```
apic properties:delete [flags]
```

### Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for properties:delete
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic properties:get

Properties get operations

### Synopsis

---

Properties get operations

```
apic properties:get [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for properties:get
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string       scope
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic properties:list

Properties list operations

## Synopsis

---

Properties list operations

```
apic properties:list [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for properties:list
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic properties:update

Properties update operations

## Synopsis

---

Properties update operations

```
apic properties:update [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for properties:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic registrations

Registrations operations

## Synopsis

---

Registrations operations

```
apic registrations [flags]
```

## Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for registrations
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic registrations:clear

Clear the Registration objects

## Synopsis

---

Clear the Registration objects

```
apic registrations:clear [flags]
```

## Options

---

<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for registrations:clear
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic registrations:create

Create a Registration object

### Synopsis

---

Create a Registration object

```
apic registrations:create [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for registrations:create
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic registrations:delete

Delete the Registration object by name or id

### Synopsis

---

Delete the Registration object by name or id

```
apic registrations:delete [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for registrations:delete
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic registrations:get

Get the Registration object by name or id

### Synopsis

---

Get the Registration object by name or id

```
apic registrations:get [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for registrations:get
--output string  Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic registrations:list

List the Registration objects

### Synopsis

---

List the Registration objects

```
apic registrations:list [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for registrations:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic registrations:update

Update the Registration object by name or id

### Synopsis

---

Update the Registration object by name or id



```
apic registrations:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help      Help for registrations:update  
--output string Write file(s) to directory, instead of STDOUT (default "-")  
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults

Role Defaults operations

## Synopsis

---

Role Defaults operations

```
apic role-defaults [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)  
--fields string      List of field names to return  
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.  
-h, --help          Help for role-defaults  
--limit int32       Maximum number of items to return  
--offset int32      Offset item number from list to begin return  
-o, --org string     Organization name or id (required)  
--output string     Write file(s) to directory, instead of STDOUT (default "-")  
--scope string      scope  
-s, --server string  management server endpoint (required)  
--space string      Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect  
--debug          Enable debug output  
--debug-output string Write debug output to file  
--live-help      Enable or disable tracking of limited usage information  
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:clear

Role Defaults clear operations

## Synopsis

---

Role Defaults clear operations

```
apic role-defaults:clear [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)  
--confirm string     Confirmation for critical updates (required)  
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
```

-h, --help	Help for role-defaults:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--subcollection string	subcollection

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:create

Role Defaults create operations

### Synopsis

---

Role Defaults create operations

```
apic role-defaults:create [flags]
```

### Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for role-defaults:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--subcollection string	subcollection

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:delete

Role Defaults delete operations

### Synopsis

---

Role Defaults delete operations

```
apic role-defaults:delete [flags]
```

### Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for role-defaults:delete
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)

```
--space string      Space name or id (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:get

Role Defaults get operations

### Synopsis

---

Role Defaults get operations

```
apic role-defaults:get [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for role-defaults:get
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:list

Role Defaults list operations

### Synopsis

---

Role Defaults list operations

```
apic role-defaults:list [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for role-defaults:list
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--space string       Space name or id (required)
--subcollection string subcollection
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:list-all

Role Defaults list-all operations

### Synopsis

---

Role Defaults list-all operations

```
apic role-defaults:list-all [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for role-defaults:list-all
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic role-defaults:update

Role Defaults update operations

### Synopsis

---

Role Defaults update operations

```
apic role-defaults:update [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for role-defaults:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--subcollection string</code>	subcollection

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic roles

Roles operations

---

### Synopsis

Roles operations

```
apic roles [flags]
```

---

### Options

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for roles
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated    space-initiated
```

---

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic roles:clear

Roles clear operations

---

### Synopsis

Roles clear operations

```
apic roles:clear [flags]
```

---

### Options

```
-c, --catalog string  Catalog name or id (required)
--confirm string     Confirmation for critical updates (required)
--consumer-org string Consumer Organization name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for roles:clear
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
--space-initiated    space-initiated
```

---

### Options inherited from parent commands

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic roles:create

Roles create operations

### Synopsis

---

Roles create operations

```
apic roles:create [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for roles:create
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string    management server endpoint (required)
--space string        Space name or id (required)
--space-initiated     space-initiated
```

### Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic roles:delete

Roles delete operations

### Synopsis

---

Roles delete operations

```
apic roles:delete [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--consumer-org string Consumer Organization name or id (required)
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for roles:delete
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string    management server endpoint (required)
--space string        Space name or id (required)
--space-initiated     space-initiated
```

### Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
```

<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic roles:get

Roles get operations

### Synopsis

Roles get operations

```
apic roles:get [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for roles:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic roles:list

Roles list operations

### Synopsis

Roles list operations

```
apic roles:list [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--consumer-org string</code>	Consumer Organization name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for roles:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>--scope string</code>	scope
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file

```
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic roles:update

Roles update operations

### Synopsis

Roles update operations

```
apic roles:update [flags]
```

### Options

```
-c, --catalog string    Catalog name or id (required)
--consumer-org string  Consumer Organization name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for roles:update
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated      space-initiated
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services

Services operations

### Synopsis

Services operations

```
apic services [flags]
```

### Options

```
-c, --catalog string    Catalog name or id (required)
--configured-gateway-service string Configured Gateway Service name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help             Help for services
--limit int32          Maximum number of items to return
--offset int32         Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
--space-initiated      space-initiated
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
```



--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:clear

Services clear operations

### Synopsis

Services clear operations

```
apic services:clear [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for services:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:clear-all

Services clear-all operations

### Synopsis

Services clear-all operations

```
apic services:clear-all [flags]
```

### Options

-c, --catalog string	Catalog name or id (required)
--configured-gateway-service string	Configured Gateway Service name or id (required)
--confirm string	Confirmation for critical updates (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for services:clear-all
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

### Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:create

Services create operations

### Synopsis

Services create operations

```
apic services:create [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for services:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:delete

Services delete operations

### Synopsis

Services delete operations

```
apic services:delete [flags]
```

### Options

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for services:delete
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:get

Services get operations

### Synopsis

---

Services get operations

```
apic services:get [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to

yaml.

<code>-h, --help</code>	Help for services:get
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic services:list

Services list operations

### Synopsis

---

Services list operations

```
apic services:list [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--configured-gateway-service string</code>	Configured Gateway Service name or id (required)
<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to

yaml.

<code>-h, --help</code>	Help for services:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--space string</code>	Space name or id (required)
<code>--space-initiated</code>	space-initiated

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:list-all

Services list-all operations

### Synopsis

Services list-all operations

```
apic services:list-all [flags]
```

### Options

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--fields string              List of field names to return
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                   Help for services:list-all
--limit int32                Maximum number of items to return
--offset int32               Offset item number from list to begin return
-o, --org string             Organization name or id (required)
--output string              Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string          management server endpoint (required)
--space string               Space name or id (required)
--space-initiated            space-initiated
```

### Options inherited from parent commands

```
--accept-license            Accept the license for API Connect
--debug                     Enable debug output
--debug-output string       Write debug output to file
--live-help                  Enable or disable tracking of limited usage information
-m, --mode string           Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic services:update

Services update operations

### Synopsis

Services update operations

```
apic services:update [flags]
```

### Options

```
-c, --catalog string          Catalog name or id (required)
--configured-gateway-service string  Configured Gateway Service name or id (required)
--format string              Output format. One of [json yaml go-template=... go-template-file=...], defaults to
yaml.
-h, --help                   Help for services:update
--id                          id
-o, --org string             Organization name or id (required)
--output string              Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string          management server endpoint (required)
--space string               Space name or id (required)
--space-initiated            space-initiated
```

### Options inherited from parent commands

```
--accept-license            Accept the license for API Connect
--debug                     Enable debug output
--debug-output string       Write debug output to file
--live-help                  Enable or disable tracking of limited usage information
-m, --mode string           Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic space-settings

Space Settings operations

### Synopsis

---

Space Settings operations

```
apic space-settings [flags]
```

### Options

---

```
-h, --help Help for space-settings
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic space-settings:get

Get the Space Setting object

### Synopsis

---

Get the Space Setting object

```
apic space-settings:get [flags]
```

### Options

---

```
-c, --catalog string Catalog name or id (required)
--fields string List of field names to return
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for space-settings:get
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
--space string Space name or id (required)
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic space-settings:update

Update the Space Setting object

### Synopsis

---

Update the Space Setting object

```
apic space-settings:update [flags]
```

## Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for space-settings:update
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces

Spaces operations

## Synopsis

---

Spaces operations

```
apic spaces [flags]
```

## Options

---

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for spaces
--limit int32	Maximum number of items to return
--my	my
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces:clear

Clear the Space objects

## Synopsis

---

Clear the Space objects

```
apic spaces:clear [flags]
```

## Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--confirm string</code>	Confirmation for critical updates (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for spaces:clear
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic spaces:create

Create a Space object

### Synopsis

---

Create a Space object

```
apic spaces:create [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for spaces:create
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic spaces:delete

Delete the Space object by name or id

### Synopsis

---

Delete the Space object by name or id

```
apic spaces:delete [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for spaces:delete
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces:get

Get the Space object by name or id

### Synopsis

---

Get the Space object by name or id

```
apic spaces:get [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for spaces:get
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string    management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces:list

List the Space objects

### Synopsis

---

List the Space objects

```
apic spaces:list [flags]
```

### Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help            Help for spaces:list
--limit int32         Maximum number of items to return
--my                  my
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string    management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces:transfer-owner

Transfer owner to an associate

### Synopsis

---

Transfer owner to an associate

```
apic spaces:transfer-owner [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for spaces:transfer-owner
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic spaces:update

Update the Space object by name or id

### Synopsis

---

Update the Space object by name or id

```
apic spaces:update [flags]
```

### Options

---

<code>-c, --catalog string</code>	Catalog name or id (required)
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for spaces:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic subscriber-events

## Synopsis

---

Subscriber Events operations

```
apic subscriber-events [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for subscriber-events
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
--scope string       scope
-s, --server string  management server endpoint (required)
--state string       State for a webhook event in subscriber queue
--webhook string     Webhook name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic subscriber-events:get

Subscriber Events get operations

## Synopsis

---

Subscriber Events get operations

```
apic subscriber-events:get [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for subscriber-events:get
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string       scope
-s, --server string  management server endpoint (required)
--webhook string     Webhook name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic subscriber-events:list

Subscriber Events list operations

## Synopsis

---

Subscriber Events list operations

```
apic subscriber-events:list [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for subscriber-events:list
--limit int32         Maximum number of items to return
--offset int32        Offset item number from list to begin return
-o, --org string       Organization name or id (required)
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string    management server endpoint (required)
--state string        State for a webhook event in subscriber queue
--webhook string      Webhook name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic subscriptions

Subscriptions operations

## Synopsis

---

Subscriptions operations

```
apic subscriptions [flags]
```

## Options

---

```
-a, --app string        Application name or id (required)
-c, --catalog string    Catalog name or id (required)
--consumer-org string   Consumer Organization name or id (required)
--fields string         List of field names to return
--format string         Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for subscriptions
--limit int32          Maximum number of items to return
--offset int32         Offset item number from list to begin return
-o, --org string        Organization name or id (required)
--output string         Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string     management server endpoint (required)
--space string          Space name or id (required)
--space-initiated       space-initiated
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic subscriptions:clear

Clear the Subscription objects

## Synopsis

---

Clear the Subscription objects

```
apic subscriptions:clear [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--confirm string	Confirmation for critical updates (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:clear
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic subscriptions:create

Create a Subscription object

## Synopsis

---

Create a Subscription object

```
apic subscriptions:create [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:create
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic subscriptions:delete

Delete the Subscription object by name or id

## Synopsis

---

Delete the Subscription object by name or id

```
apic subscriptions:delete [flags]
```

## Options

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:delete
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic subscriptions:get

Get the Subscription object by name or id

## Synopsis

Get the Subscription object by name or id

```
apic subscriptions:get [flags]
```

## Options

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:get
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## apic subscriptions:list

List the Subscription objects

## Synopsis

List the Subscription objects

```
apic subscriptions:list [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic subscriptions:update

Update the Subscription object by name or id

## Synopsis

---

Update the Subscription object by name or id

```
apic subscriptions:update [flags]
```

## Options

---

-a, --app string	Application name or id (required)
-c, --catalog string	Catalog name or id (required)
--consumer-org string	Consumer Organization name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for subscriptions:update
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string	management server endpoint (required)
--space string	Space name or id (required)
--space-initiated	space-initiated

## Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic task-queues

Task Queues operations

## Synopsis

---

Task Queues operations

```
apic task-queues [flags]
```

## Options

---

```
--apply-filter      Filter tasks
--cascade           Cascade the behavior
--fields string     List of field names to return
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for task-queues
--iteration string  iteration of task
--kind string       kind of item
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--query string      Add query to request
--scope string      scope
-s, --server string management server endpoint (required)
--state string      State for a webhook event in subscriber queue
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic task-queues:get

Task Queues get operations

## Synopsis

---

Task Queues get operations

```
apic task-queues:get [flags]
```

## Options

---

```
--cascade           Cascade the behavior
--fields string     List of field names to return
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for task-queues:get
--kind string       kind of item
--output string     Write file(s) to directory, use - for STDOUT. (default: cwd)
--query string      Add query to request
--scope string      scope
-s, --server string management server endpoint (required)
--state string      State for a webhook event in subscriber queue
```

## Options inherited from parent commands

---

```
--accept-license    Accept the license for API Connect
--debug             Enable debug output
--debug-output string Write debug output to file
--live-help         Enable or disable tracking of limited usage information
-m, --mode string   Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic task-queues:list

Task Queues list operations

## Synopsis

---

Task Queues list operations

```
apic task-queues:list [flags]
```

## Options

---

```
--apply_filter      Filter tasks
--cascade           Cascade the behavior
--fields string     List of field names to return
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for task-queues:list
--iteration string  iteration of task
--kind string       kind of item
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--query string      Add query to request
--scope string      scope
-s, --server string management server endpoint (required)
--state string      State for a webhook event in subscriber queue
```

## Options inherited from parent commands

---

```
--accept-license   Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tasks

Tasks operations

## Synopsis

---

Tasks operations

```
apic tasks [flags]
```

## Options

---

```
-c, --catalog string  Catalog name or id (required)
--fields string       List of field names to return
--format string       Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for tasks
--limit int32        Maximum number of items to return
--my                  my
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--originated         originated
--output string       Write file(s) to directory, instead of STDOUT (default "-")
--scope string        scope
-s, --server string   management server endpoint (required)
--space string        Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license   Accept the license for API Connect
--debug            Enable debug output
--debug-output string Write debug output to file
--live-help        Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tasks:get

Tasks get operations

## Synopsis

---

Tasks get operations



```
apic tasks:get [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for tasks:get
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tasks:list

Tasks list operations

## Synopsis

---

Tasks list operations

```
apic tasks:list [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--fields string        List of field names to return
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for tasks:list
--limit int32          Maximum number of items to return
--my                   my
--offset int32         Offset item number from list to begin return
-o, --org string        Organization name or id (required)
--originated           originated
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tasks:update

Tasks update operations

## Synopsis

---

Tasks update operations

```
apic tasks:update [flags]
```

## Options

---

```
-c, --catalog string    Catalog name or id (required)
--format string        Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help             Help for tasks:update
-o, --org string        Organization name or id (required)
--output string        Write file(s) to directory, instead of STDOUT (default "-")
--scope string         scope
-s, --server string    management server endpoint (required)
--space string         Space name or id (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles

Tls Client Profiles operations

## Synopsis

---

Tls Client Profiles operations

```
apic tls-client-profiles [flags]
```

## Options

---

```
--fields string    List of field names to return
--format string    Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for tls-client-profiles
--limit int32      Maximum number of items to return
--offset int32     Offset item number from list to begin return
-o, --org string   Organization name or id (required)
--output string    Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:clear

Clear the TLS Client Profile objects

## Synopsis

---

Clear the TLS Client Profile objects

```
apic tls-client-profiles:clear [flags]
```

## Options

---

```
--confirm string    Confirmation for critical updates (required)
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help         Help for tls-client-profiles:clear
-o, --org string    Organization name or id (required)
```

```
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:clear-all

Clear all TLS Client Profile objects in all collections

## Synopsis

---

Clear all TLS Client Profile objects in all collections

```
apic tls-client-profiles:clear-all [flags]
```

## Options

---

```
--cascade           Cascade the behavior
--confirm string    Confirmation for critical updates (required)
--format string     Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for tls-client-profiles:clear-all
-o, --org string    Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:create

Create a TLS Client Profile object

## Synopsis

---

Create a TLS Client Profile object

```
apic tls-client-profiles:create [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for tls-client-profiles:create
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string  Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:delete

Delete a TLS Client Profile

### Synopsis

---

Delete a TLS Client Profile

```
apic tls-client-profiles:delete [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for <code>tls-client-profiles:delete</code>
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:get

Get the TLS Client Profile object by name and version

### Synopsis

---

Get the TLS Client Profile object by name and version

```
apic tls-client-profiles:get [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for <code>tls-client-profiles:get</code>
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-client-profiles:list

List the TLS Client Profile objects

## Synopsis

---

List the TLS Client Profile objects

```
apic tls-client-profiles:list [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for tls-client-profiles:list
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tls-client-profiles:list-all

List all TLS Client Profile objects in all collections

## Synopsis

---

List all TLS Client Profile objects in all collections

```
apic tls-client-profiles:list-all [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for tls-client-profiles:list-all
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tls-client-profiles:update

Update the TLS Client Profile object by name and version

## Synopsis

---

Update the TLS Client Profile object by name and version

```
apic tls-client-profiles:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for tls-client-profiles:update
--id            id
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles

Tls Server Profiles operations

## Synopsis

---

Tls Server Profiles operations

```
apic tls-server-profiles [flags]
```

## Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for tls-server-profiles
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles:clear

Clear the TLS Server Profile objects

## Synopsis

---

Clear the TLS Server Profile objects

```
apic tls-server-profiles:clear [flags]
```

## Options

---

```
--cascade      Cascade the behavior
--confirm string Confirmation for critical updates (required)
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help     Help for tls-server-profiles:clear
```

```
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string   management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles:clear-all

Clear all TLS Server Profile objects in all collections

### Synopsis

---

Clear all TLS Server Profile objects in all collections

```
apic tls-server-profiles:clear-all [flags]
```

### Options

---

```
--cascade           Cascade the behavior
--confirm string    Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for tls-server-profiles:clear-all
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
--live-help           Enable or disable tracking of limited usage information
-m, --mode string     Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles:create

Create a TLS Server Profile object

### Synopsis

---

Create a TLS Server Profile object

```
apic tls-server-profiles:create [flags]
```

### Options

---

```
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help          Help for tls-server-profiles:create
-o, --org string     Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug               Enable debug output
--debug-output string Write debug output to file
```

<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles:delete

Delete a TLS Server Profile

### Synopsis

Delete a TLS Server Profile

```
apic tls-server-profiles:delete [flags]
```

### Options

<code>--cascade</code>	Cascade the behavior
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for tls-server-profiles:delete
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic tls-server-profiles:get

Get the TLS Server Profile object by name and version

### Synopsis

Get the TLS Server Profile object by name and version

```
apic tls-server-profiles:get [flags]
```

### Options

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for tls-server-profiles:get
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).



---

## apic tls-server-profiles:list

List the TLS Server Profile objects

### Synopsis

---

List the TLS Server Profile objects

```
apic tls-server-profiles:list [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for tls-server-profiles:list
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tls-server-profiles:list-all

List all TLS Server Profile objects in all collections

### Synopsis

---

List all TLS Server Profile objects in all collections

```
apic tls-server-profiles:list-all [flags]
```

### Options

---

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help       Help for tls-server-profiles:list-all
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic tls-server-profiles:update

Update the TLS Server Profile object by name and version

## Synopsis

---

Update the TLS Server Profile object by name and version

```
apic tls-server-profiles:update [flags]
```

## Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for tls-server-profiles:update
<code>--id</code>	id
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic truststores

Truststores operations

## Synopsis

---

Truststores operations

```
apic truststores [flags]
```

## Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for truststores
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic truststores:clear

Clear the Truststore objects

## Synopsis

---

Clear the Truststore objects

```
apic truststores:clear [flags]
```

## Options

---

```
--confirm string  Confirmation for critical updates (required)
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:clear
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic truststores:create

Create a Truststore object

## Synopsis

---

Create a Truststore object

```
apic truststores:create [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:create
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic truststores:delete

Delete the Truststore object by name or id

## Synopsis

---

Delete the Truststore object by name or id

```
apic truststores:delete [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:delete
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic truststores:get

Get the Truststore object by name or id

### Synopsis

Get the Truststore object by name or id

```
apic truststores:get [flags]
```

### Options

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:get
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic truststores:list

List the Truststore objects

### Synopsis

List the Truststore objects

```
apic truststores:list [flags]
```

### Options

```
--fields string  List of field names to return
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:list
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#), for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic truststores:update

Update the Truststore object by name or id

### Synopsis

---

Update the Truststore object by name or id

```
apic truststores:update [flags]
```

### Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for truststores:update
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help     Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registries

User Registries operations

### Synopsis

---

User Registries operations

```
apic user-registries [flags]
```

### Options

---

```
--fields string List of field names to return
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for user-registries
--limit int32   Maximum number of items to return
--offset int32  Offset item number from list to begin return
-o, --org string Organization name or id (required)
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug          Enable debug output
--debug-output string Write debug output to file
--live-help     Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registries:clear

Clear the User Registry objects

### Synopsis

---

Clear the User Registry objects

```
apic user-registries:clear [flags]
```

## Options

---

<code>--confirm</code>	string	Confirmation for critical updates (required)
<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for user-registries:clear
<code>-o, --org</code>	string	Organization name or id (required)
<code>--output</code>	string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code>	string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>		Accept the license for API Connect
<code>--debug</code>		Enable debug output
<code>--debug-output</code>	string	Write debug output to file
<code>--live-help</code>		Enable or disable tracking of limited usage information
<code>-m, --mode</code>	string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registries:create

Create a User Registry object

## Synopsis

---

Create a User Registry object

```
apic user-registries:create [flags]
```

## Options

---

<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for user-registries:create
<code>-o, --org</code>	string	Organization name or id (required)
<code>--output</code>	string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code>	string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>		Accept the license for API Connect
<code>--debug</code>		Enable debug output
<code>--debug-output</code>	string	Write debug output to file
<code>--live-help</code>		Enable or disable tracking of limited usage information
<code>-m, --mode</code>	string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registries:delete

Delete the User Registry object by name or id

## Synopsis

---

Delete the User Registry object by name or id

```
apic user-registries:delete [flags]
```

## Options

---

<code>--format</code>	string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>		Help for user-registries:delete
<code>-o, --org</code>	string	Organization name or id (required)
<code>--output</code>	string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code>	string	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registries:execute

Execute a User Registry operation

### Synopsis

---

Execute a User Registry operation

```
apic user-registries:execute [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:execute
<code>--operation string</code>	Operation to perform on the user registry (get_base_dn_list, validate_password) (required)
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--test-config-only</code>	test-config-only

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registries:get

Get the User Registry object by name or id

### Synopsis

---

Get the User Registry object by name or id

```
apic user-registries:get [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registries:list

List the User Registry objects

### Synopsis

---

List the User Registry objects

```
apic user-registries:list [flags]
```

### Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registries:search

Search for users in the user registry

### Synopsis

---

Search for users in the user registry

```
apic user-registries:search [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:search
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---



---

## apic user-registries:test-connection

Test a User Registry connection

### Synopsis

---

Test a User Registry connection

```
apic user-registries:test-connection [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:test-connection
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--test-config-only</code>	test-config-only

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registries:update

Update the User Registry object by name or id

### Synopsis

---

Update the User Registry object by name or id

```
apic user-registries:update [flags]
```

### Options

---

<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for user-registries:update
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic user-registry-settings

User Registry Settings operations

### Synopsis

---

User Registry Settings operations

```
apic user-registry-settings [flags]
```

## Options

---

```
-h, --help Help for user-registry-settings
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registry-settings:get

Get the User Registry Setting object

## Synopsis

---

Get the User Registry Setting object

```
apic user-registry-settings:get [flags]
```

## Options

---

```
--fields string List of field names to return
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for user-registry-settings:get
--output string Write file(s) to directory, use - for STDOUT. (default: cwd)
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic user-registry-settings:update

Update the User Registry Setting object

## Synopsis

---

Update the User Registry Setting object

```
apic user-registry-settings:update [flags]
```

## Options

---

```
--format string Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help Help for user-registry-settings:update
--output string Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
```

```
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic users

Users operations

### Synopsis

Users operations

```
apic users [flags]
```

### Options

```
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for users
--limit int32        Maximum number of items to return
--offset int32       Offset item number from list to begin return
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
--user-registry string User Registry name or id (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic users:clear

Clear the User objects

### Synopsis

Clear the User objects

```
apic users:clear [flags]
```

### Options

```
--confirm string      Confirmation for critical updates (required)
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for users:clear
-o, --org string      Organization name or id (required)
--output string      Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string  management server endpoint (required)
--user-registry string User Registry name or id (required)
```

### Options inherited from parent commands

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help      Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic users:create

Create a User object

### Synopsis

---

Create a User object

```
apic users:create [flags]
```

### Options

---

<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for users:create
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code> string	management server endpoint (required)
<code>--user-registry</code> string	User Registry name or id (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic users:delete

Delete the User object by name or id

### Synopsis

---

Delete the User object by name or id

```
apic users:delete [flags]
```

### Options

---

<code>--format</code> string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for users:delete
<code>-o, --org</code> string	Organization name or id (required)
<code>--output</code> string	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server</code> string	management server endpoint (required)
<code>--user-registry</code> string	User Registry name or id (required)

### Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output</code> string	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode</code> string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic users:get

Get the User object by name or id

## Synopsis

---

Get the User object by name or id

```
apic users:get [flags]
```

## Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for users:get
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, use - for STDOUT. (default: cwd)
<code>-s, --server string</code>	management server endpoint (required)
<code>--user-registry string</code>	User Registry name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic users:list

List the User objects

## Synopsis

---

List the User objects

```
apic users:list [flags]
```

## Options

---

<code>--fields string</code>	List of field names to return
<code>--format string</code>	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
<code>-h, --help</code>	Help for users:list
<code>--limit int32</code>	Maximum number of items to return
<code>--offset int32</code>	Offset item number from list to begin return
<code>-o, --org string</code>	Organization name or id (required)
<code>--output string</code>	Write file(s) to directory, instead of STDOUT (default "-")
<code>-s, --server string</code>	management server endpoint (required)
<code>--user-registry string</code>	User Registry name or id (required)

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic users:request-password-reset

Send reset password link

## Synopsis

---

Send reset password link

```
apic users:request-password-reset [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for users:request-password-reset
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic users:search-provider

Users search-provider operations

## Synopsis

---

Users search-provider operations

```
apic users:search-provider [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for users:search-provider
--limit int32    Maximum number of items to return
--offset int32   Offset item number from list to begin return
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
--scope string   scope
-s, --server string management server endpoint (required)
```

## Options inherited from parent commands

---

```
--accept-license  Accept the license for API Connect
--debug           Enable debug output
--debug-output string Write debug output to file
--live-help       Enable or disable tracking of limited usage information
-m, --mode string  Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic users:update

Update the User object by name or id

## Synopsis

---

Update the User object by name or id

```
apic users:update [flags]
```

## Options

---

```
--format string  Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help      Help for users:update
-o, --org string  Organization name or id (required)
--output string  Write file(s) to directory, instead of STDOUT (default "-")
-s, --server string management server endpoint (required)
--user-registry string User Registry name or id (required)
```

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic validate

Validate an API or product definition

### Synopsis

---

Validate an API or product definition

```
apic validate [FILE] [flags]
```

### Examples

---

```
Validate an API definition
$ apic validate routes.yaml
Validated routes.yaml API definition [routes:1.0]
Validate an API definition without IBM extensions
$ apic validate --no-extensions routes.yaml
Validated routes.yaml API definition [routes:1.0]
Validate a product definition and its referenced APIs
$ apic validate climb-on.yaml
Validated climb-on.yaml product definition [climb-on:1.0.0]
Validated routes.yaml API definition [valid:1.0]
Validate a product definition without validating its referenced APIs
$ apic validate --product-only climb-on.yaml
Validated climb-on.yaml product definition [climb-on:1.0.0]
```

### Options

---

<code>-h, --help</code>	Help for validate
<code>--no-extensions</code>	for API definitions, do not validate against IBM Swagger extensions
<code>-p, --product-only</code>	for products definitions, do not validate referenced APIs

## Options inherited from parent commands

---

<code>--accept-license</code>	Accept the license for API Connect
<code>--debug</code>	Enable debug output
<code>--debug-output string</code>	Write debug output to file
<code>--live-help</code>	Enable or disable tracking of limited usage information
<code>-m, --mode string</code>	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic version

Get the APIConnect toolkit version

### Synopsis

---

Get the APIConnect toolkit version

```
apic version [flags]
```

### Options

---

<code>-h, --help</code>	Help for version
-------------------------	------------------

## Options inherited from parent commands

---

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic webhooks

Webhooks operations

### Synopsis

Webhooks operations

```
apic webhooks [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for webhooks
--limit int32       Maximum number of items to return
--offset int32      Offset item number from list to begin return
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, instead of STDOUT (default "-")
--scope string      scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic webhooks:get

Webhooks get operations

### Synopsis

Webhooks get operations

```
apic webhooks:get [flags]
```

### Options

```
-c, --catalog string  Catalog name or id (required)
--fields string      List of field names to return
--format string      Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help           Help for webhooks:get
-o, --org string     Organization name or id (required)
--output string     Write file(s) to directory, use - for STDOUT. (default: cwd)
--scope string      scope
-s, --server string  management server endpoint (required)
```

### Options inherited from parent commands

```
--accept-license      Accept the license for API Connect
--debug              Enable debug output
--debug-output string Write debug output to file
--live-help          Enable or disable tracking of limited usage information
-m, --mode string    Toolkit operation mode (default "apim")
```



**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic webhooks:list

Webhooks list operations

### Synopsis

---

Webhooks list operations

```
apic webhooks:list [flags]
```

### Options

---

-c, --catalog string	Catalog name or id (required)
--fields string	List of field names to return
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for webhooks:list
--limit int32	Maximum number of items to return
--offset int32	Offset item number from list to begin return
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## apic webhooks:update

Webhooks update operations

### Synopsis

---

Webhooks update operations

```
apic webhooks:update [flags]
```

### Options

---

-c, --catalog string	Catalog name or id (required)
--format string	Output format. One of [json yaml go-template=... go-template-file=...], defaults to yaml.
-h, --help	Help for webhooks:update
-o, --org string	Organization name or id (required)
--output string	Write file(s) to directory, instead of STDOUT (default "-")
--scope string	scope
-s, --server string	management server endpoint (required)

### Options inherited from parent commands

---

--accept-license	Accept the license for API Connect
--debug	Enable debug output
--debug-output string	Write debug output to file
--live-help	Enable or disable tracking of limited usage information
-m, --mode string	Toolkit operation mode (default "apim")

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic wsdl

Manage WSDL files

### Synopsis

---

Manage WSDL files

```
apic wsdl WSDL_FILE [flags]
```

### Options

---

```
-h, --help Help for wsdl
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## apic wsdl:introspect

Introspect a WSDL file displaying all its services

### Synopsis

---

Introspect a WSDL file displaying all its services

```
apic wsdl:introspect WSDL_FILE [flags]
```

### Options

---

```
-h, --help Help for wsdl:introspect
```

### Options inherited from parent commands

---

```
--accept-license Accept the license for API Connect
--debug Enable debug output
--debug-output string Write debug output to file
--live-help Enable or disable tracking of limited usage information
-m, --mode string Toolkit operation mode (default "apim")
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

---

## API Connect REST APIs

The API Connect platform REST APIs provide complete access to the capability of the platform.

The API Connect platform REST APIs can be used for the following actions:

- Automate the administration of the platform.
- Implement scripts and tools to support a continuous integration environment for API development and publishing.
- Manage catalogs of APIs, and their subscribers.

The operations provided in the REST APIs also correspond directly with the commands in the [developer toolkit command-line tool \(CLI\)](#).

**Note:** You can now use the URL to navigate directly to specific APIs and operations in the OpenAPI Explorer Documentation for the API Connect REST APIs. You can also select the APIs for a specific version by using the drop-down menu in the header bar.

For full details, see the [API Connect REST API documentation](#).

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Example: Using the platform REST APIs to publish a product containing a SOAP API

Review this sample scenario to see how you can use the API Connect platform REST APIs to publish a product that contains a SOAP API.

This example demonstrates how to format a request to publish the product, and includes sample files that illustrate the product definition, the API definition, and the SOAP API's WSDL description.

---

### Using the Catalogs API

Publish the product using the API Connect platform **Catalogs** API:

- To publish a product to a catalog, use the following syntax:

```
POST catalogs/{org}/{catalog}/publish
```

- To publish a product to a space within a catalog, additionally specify the space with the following syntax:

```
POST catalogs/{org}/{catalog}/{space}/publish
```

The payload in both cases is a multipart, form-encoded body containing three required parts and one optional part:

- (Required) A part named **product** with media type **application/yaml** or **application/json**, containing the yaml or JSON source for the product to be published.
- (Required) One or more occurrences of a part named **openapi** with media type **application/yaml** or **application/json**, containing the YAML or JSON source for each API in the product.
- (Required for SOAP APIs) One or more occurrences of a part named **wSDL** with media type **application/wSDL**, **application/wSDL+xml**, **text/xml**, or **application/zip** containing the WSDL definitions (and referenced XSD files if applicable) associated with the API content.
- (Optional) A part named **gateway\_service\_urls** with media type **application/yaml** or **application/json** which is a YAML or JSON array of URLs to specify a subset of the configured gateway services within the catalog or space as publish targets. These URLs can be obtained using the following calls:

- o For a catalog:

```
GET /api/catalogs/{org}/{catalog}/configured-gateway-services
```

- o For a space:

```
GET /spaces/{org}/{catalog}/{space}/configured-gateway-services
```

Within the **product** content, the form of the references to APIs in the **apis** section is significant. Each API reference must use the **name** property instead of the **\$ref** form that may appear in other situations. Here is an example showing a product that refers to the API **globalweather** version **1.0.0** in the correct form:

```
info:
  version: 1.0.0
  title: GlobalWeatherProduct
  name: globalweatherproduct
plans:
  ...
apis:
  globalweather1.0.0:
    name: globalweather:1.0.0
  ...
```

---

### Example publish operation using cURL

This example publishes a product containing a SOAP API to the *Sandbox* catalog of the *onlinebanking* provider organization, using the **curl -F** option to create a set of form-encoded payload parts and the **@** prefix to read the content for each part from a file. Note the part name (**product**, **openapi**, **wSDL**) and the content type of each part given by the **type=** value.

In this example, the access token has been retrieved earlier and placed in an environment variable named **TOKEN**. The command is split onto multiple lines for ease of reading but must be entered as a single line.

Request:

```
curl -v -k
-F "product=@globalweatherproduct_1.0.0.yaml;type=application/yaml"
-F "openapi=@globalweather_1.0.0.yaml;type=application/yaml"
-F "wSDL=@globalweather.wSDL;type=application/wSDL"
-H "Authorization: Bearer $TOKEN"
-H 'Content-Type: multipart/form-data'
-H 'Accept: */*' https://dev.apic.example.com/api/catalogs/onlinebanking/sandbox/publish
```

Response:

201 Created

```
{
  "type": "product",
  "api_version": "2.0.0",
  "id": "d03129a4-9a99-4c88-85e6-d51bbf312164",
  "name": "globalweatherproduct",
  "version": "1.0.0",
  "title": "GlobalWeatherProduct",
  "state": "published",
```

```

"scope": "catalog",
"gateway_types": [
  "datapower-api-gateway"
],
"billing_urls": [],
"api_urls": [
  "https://dev.apic.example.com/api/catalogs/87227774-d297-4064-8908-b8f0b1db71a5/1bc8f4ba-4178-4a4b-93e8-96a8ca9667fd/apis/d5986b78-d556-4457-8c38-951c471f57fc"
],
"gateway_service_urls": [
  "https://dev.apic.example.com/api/catalogs/87227774-d297-4064-8908-b8f0b1db71a5/1bc8f4ba-4178-4a4b-93e8-96a8ca9667fd/configured-gateway-services/a416f39d-39e1-484c-9f7e-4a599abefec8"
],
"oauth_provider_urls": [],
"plans": [
  {
    "title": "Default Plan",
    "name": "default-plan",
    "apis": [
      {
        "id": "d5986b78-d556-4457-8c38-951c471f57fc",
        "name": "globalweather",
        "title": "GlobalWeather",
        "version": "1.0.0",
        "url": "https://dev.apic.example.com/api/catalogs/87227774-d297-4064-8908-b8f0b1db71a5/1bc8f4ba-4178-4a4b-93e8-96a8ca9667fd/apis/d5986b78-d556-4457-8c38-951c471f57fc"
      }
    ]
  }
],
"visibility": {
  "view": {
    "type": "public",
    "enabled": true
  },
  "subscribe": {
    "type": "authenticated",
    "enabled": true
  }
},
"task_urls": [],
"created_at": "2021-12-15T07:35:40.295Z",
"updated_at": "2021-12-15T07:35:40.295Z",
"org_url": "https://dev.apic.example.com/api/orgs/87227774-d297-4064-8908-b8f0b1db71a5",
"catalog_url": "https://dev.apic.example.com/api/catalogs/87227774-d297-4064-8908-b8f0b1db71a5/1bc8f4ba-4178-4a4b-93e8-96a8ca9667fd",
"url": "https://dev.apic.example.com/api/catalogs/87227774-d297-4064-8908-b8f0b1db71a5/1bc8f4ba-4178-4a4b-93e8-96a8ca9667fd/products/d03129a4-9a99-4c88-85e6-d51bbf312164"

```

The request specifies three files, which are included with this example for reference:

- [Product definition](#) -F "product=@globalweatherproduct\_1.0.0.yaml;type=application/yaml"
- [API definition](#) -F "openapi=@globalweather\_1.0.0.yaml;type=application/yaml"
- [WSDL description](#) -F "wsdl=@globalweather.wsdl;type=application/wsdl"

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.

For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

## Example Product file: globalweatherproduct\_1.0.0.yaml

The following code sample contains a Product definition.

The Product definition file globalweatherproduct\_1.0.0.yaml is referenced by the example scenario in [Example: Using the platform REST APIs to publish a product containing a SOAP API](#).

```

info:
  version: 1.0.0
  title: GlobalWeatherProduct
  name: globalweatherproduct
gateways:
- datapower-api-gateway
plans:
  default-plan:
    title: Default Plan
    description: Default Plan
    rate-limits:
      default:
        value: 100/1hour
    apis:
      globalweather1.0.0: {}
apis:
  globalweather1.0.0:
    name: globalweather:1.0.0
visibility:
  view:
    type: public
    orgs: []

```

```
tags: []
enabled: true
subscribe:
  type: authenticated
  orgs: []
  tags: []
  enabled: true
product: 1.0.0
```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Example API definition: globalweather\_1.0.0.yaml

The following code sample contains an API definition.

The API definition file globalweather\_1.0.0.yaml is referenced by the example scenario in [Example: Using the platform REST APIs to publish a product containing a SOAP API](#).

```
swagger: '2.0'
info:
  title: GlobalWeather
  description: ''
  x-ibm-name: globalweather
  version: 1.0.0
schemes:
- https
basePath: /globalweather
produces:
- application/xml
consumes:
- text/xml
securityDefinitions:
  clientID:
    type: apiKey
    in: header
    name: X-IBM-Client-Id
security:
- clientID: []
x-ibm-configuration:
  type: wsd1
  phase: realized
  enforced: true
  testable: true
  gateway: datapower-api-gateway
  cors:
    enabled: true
  wsd1-definition:
    wsd1: globalweather.wsd1
    service: GlobalWeather
    port: GlobalWeatherSoap
    soap-version: '1.1'
  assembly:
    execute:
      - invoke:
          title: invoke
          target-url: 'http://www.webserviceX.com/globalweather.asmx'
          version: 2.0.0
          header-control:
            type: blocklist
            values: []
          parameter-control:
            type: blocklist
            values: []
x-ibm-apiconnect-wsd1:
  package-version: 2.0.35
  options: {}
  messages:
    info:
      - message: >-
          The wsd1 'service' has multiple 'ports'. The api is generated using
          information in the first soap 'port'.
    warning: []
    error: []
paths:
  /GetWeather:
    post:
      summary: Operation GetWeather
      description: Get weather report for all major cities around the world.
      operationId: GetWeather
      x-ibm-soap:
        soap-action: 'http://www.webserviceX.NET/GetWeather'
        soap-operation: '{http://www.webserviceX.NET}GetWeather'
      parameters:
        - in: body
          name: body
          required: true
          schema:
```

```

    $ref: '#/definitions/GetWeatherInput'
  responses:
    default:
      description: ''
      schema:
        $ref: '#/definitions/GetWeatherOutput'
  /GetCitiesByCountry:
    post:
      summary: Operation GetCitiesByCountry
      description: "Get all major cities by country name(full / part)."
      operationId: GetCitiesByCountry
      x-ibm-soap:
        soap-action: 'http://www.webserviceX.NET/GetCitiesByCountry'
        soap-operation: '{http://www.webserviceX.NET}GetCitiesByCountry'
      parameters:
        - in: body
          name: body
          required: true
          schema:
            $ref: '#/definitions/GetCitiesByCountryInput'
      responses:
        default:
          description: ''
          schema:
            $ref: '#/definitions/GetCitiesByCountryOutput'
definitions:
  Security:
    xml:
      namespace: >-
      http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
    prefix: wsse
    description: Header for WS-Security
    type: object
    properties:
      UsernameToken:
        xml:
          namespace: >-
          http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
          prefix: wsse
          type: object
          properties:
            Username:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
                prefix: wsse
                type: string
            Password:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
                prefix: wsse
                type: string
            Nonce:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
                prefix: wsse
                type: string
            properties:
              EncodingType:
                xml:
                  namespace: ''
                  attribute: true
                  type: string
            Created:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
                prefix: wsu
                type: string
      Timestamp:
        xml:
          namespace: >-
          http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
          prefix: wsu
          type: object
          properties:
            Created:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
                prefix: wsu
                type: string
            Expires:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
                prefix: wsu
                type: string
            Id:
              xml:
                namespace: >-
                http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
                prefix: wsu
                attribute: true

```

```

    type: string
GetWeatherInput:
description: Input message for wsdl operation GetWeather
type: object
properties:
  Envelope:
    xml:
      namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
      prefix: soapenv
      type: object
      properties:
        Header:
          $ref: '#/definitions/GetWeatherHeader'
        Body:
          xml:
            namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
            prefix: soapenv
            type: object
            properties:
              GetWeather:
                $ref: '#/definitions/GetWeather_element_tns'
                required:
                  - GetWeather
            required:
              - Body
          required:
            - Envelope
x-ibm-schema:
  wsdl-port: '{http://www.webserviceX.NET}GlobalWeatherSoap'
  wsdl-operation: GetWeather
  wsdl-message-direction-or-name: GetWeatherRequest
example: >-

```

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <!-- The Security element should be removed if WS-Security is not enabled on the SOAP target-url -->
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>string</wsse:Username>
        <wsse:Password>string</wsse:Password>
        <wsse:Nonce EncodingType="string">string</wsse:Nonce>
        <wsu:Created>string</wsu:Created>
      </wsse:UsernameToken>
      <wsu:Timestamp wsu:Id="string">
        <wsu:Created>string</wsu:Created>
        <wsu:Expires>string</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tns:GetWeather xmlns:tns="http://www.webserviceX.NET"><!-- mandatory -->
      <tns:CityName>string</tns:CityName>
      <tns:CountryName>string</tns:CountryName>
    </tns:GetWeather>
  </soapenv:Body>
</soapenv:Envelope>

```

```

GetWeatherHeader:
description: Input headers for wsdl operation GetWeather
type: object
properties:
  Security:
    $ref: '#/definitions/Security'

```

```

GetWeatherOutput:
description: Output message for wsdl operation GetWeather
type: object
properties:
  Envelope:
    xml:
      namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
      prefix: soapenv
      type: object
      properties:
        Body:
          xml:
            namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
            prefix: soapenv
            type: object
            properties:
              GetWeatherResponse:
                $ref: '#/definitions/GetWeatherResponse_element_tns'
            required:
              - Body
          required:
            - Envelope
x-ibm-schema:
  wsdl-port: '{http://www.webserviceX.NET}GlobalWeatherSoap'
  wsdl-operation: GetWeather
  wsdl-message-direction-or-name: GetWeatherResponse
example: >-

```

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <tns:GetWeatherResponse xmlns:tns="http://www.webserviceX.NET">

```

```

    <tns:GetWeatherResult>string</tns:GetWeatherResult>
  </tns:GetWeatherResponse>
</soapenv:Body>
</soapenv:Envelope>
GetCitiesByCountryInput:
description: Input message for wsdl operation GetCitiesByCountry
type: object
properties:
  Envelope:
    xml:
      namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
      prefix: soapenv
      type: object
      properties:
        Header:
          $ref: '#/definitions/GetCitiesByCountryHeader'
        Body:
          xml:
            namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
            prefix: soapenv
            type: object
            properties:
              GetCitiesByCountry:
                $ref: '#/definitions/GetCitiesByCountry_element_tns'
            required:
              - GetCitiesByCountry
          required:
            - Body
        required:
          - Envelope
  x-ibm-schema:
    wsdl-port: '{http://www.webserviceX.NET}GlobalWeatherSoap'
    wsdl-operation: GetCitiesByCountry
    wsdl-message-direction-or-name: GetCitiesByCountryRequest
example: >-

```

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <!-- The Security element should be removed if WS-Security is not enabled on the SOAP target-url -->
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>string</wsse:Username>
        <wsse:Password>string</wsse:Password>
        <wsse:Nonce EncodingType="string">string</wsse:Nonce>
        <wsu:Created>string</wsu:Created>
      </wsse:UsernameToken>
      <wsu:Timestamp wsu:Id="string">
        <wsu:Created>string</wsu:Created>
        <wsu:Expires>string</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tns:GetCitiesByCountry xmlns:tns="http://www.webserviceX.NET"><!-- mandatory -->
      <tns:CountryName>string</tns:CountryName>
    </tns:GetCitiesByCountry>
  </soapenv:Body>
</soapenv:Envelope>
GetCitiesByCountryHeader:
description: Input headers for wsdl operation GetCitiesByCountry
type: object
properties:
  Security:
    $ref: '#/definitions/Security'
GetCitiesByCountryOutput:
description: Output message for wsdl operation GetCitiesByCountry
type: object
properties:
  Envelope:
    xml:
      namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
      prefix: soapenv
      type: object
      properties:
        Body:
          xml:
            namespace: 'http://schemas.xmlsoap.org/soap/envelope/'
            prefix: soapenv
            type: object
            properties:
              GetCitiesByCountryResponse:
                $ref: '#/definitions/GetCitiesByCountryResponse_element_tns'
            required:
              - Body
          required:
            - Envelope
  x-ibm-schema:
    wsdl-port: '{http://www.webserviceX.NET}GlobalWeatherSoap'
    wsdl-operation: GetCitiesByCountry
    wsdl-message-direction-or-name: GetCitiesByCountryResponse
example: >-

```

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

```



```

<soapenv:Body>
  <tns:GetCitiesByCountryResponse xmlns:tns="http://www.webserviceX.NET">
    <tns:GetCitiesByCountryResult>string</tns:GetCitiesByCountryResult>
  </tns:GetCitiesByCountryResponse>
</soapenv:Body>
</soapenv:Envelope>
GetWeather_element_tns:
  xml:
    namespace: 'http://www.webserviceX.NET'
    prefix: tns
    name: GetWeather
    type: object
    properties:
      CityName:
        xml:
          namespace: 'http://www.webserviceX.NET'
          prefix: tns
          type: string
      CountryName:
        xml:
          namespace: 'http://www.webserviceX.NET'
          prefix: tns
          type: string
    example: |-

```

```

<tns:GetWeather xmlns:tns="http://www.webserviceX.NET">
  <tns:CityName>string</tns:CityName>
  <tns:CountryName>string</tns:CountryName>
</tns:GetWeather>
GetWeatherResponse_element_tns:
  xml:
    namespace: 'http://www.webserviceX.NET'
    prefix: tns
    name: GetWeatherResponse
    type: object
    properties:
      GetWeatherResult:
        xml:
          namespace: 'http://www.webserviceX.NET'
          prefix: tns
          type: string
    example: |-

```

```

<tns:GetWeatherResponse xmlns:tns="http://www.webserviceX.NET">
  <tns:GetWeatherResult>string</tns:GetWeatherResult>
</tns:GetWeatherResponse>
GetCitiesByCountry_element_tns:
  xml:
    namespace: 'http://www.webserviceX.NET'
    prefix: tns
    name: GetCitiesByCountry
    type: object
    properties:
      CountryName:
        xml:
          namespace: 'http://www.webserviceX.NET'
          prefix: tns
          type: string
    example: |-

```

```

<tns:GetCitiesByCountry xmlns:tns="http://www.webserviceX.NET">
  <tns:CountryName>string</tns:CountryName>
</tns:GetCitiesByCountry>
GetCitiesByCountryResponse_element_tns:
  xml:
    namespace: 'http://www.webserviceX.NET'
    prefix: tns
    name: GetCitiesByCountryResponse
    type: object
    properties:
      GetCitiesByCountryResult:
        xml:
          namespace: 'http://www.webserviceX.NET'
          prefix: tns
          type: string
    example: |-

```

```

<tns:GetCitiesByCountryResponse xmlns:tns="http://www.webserviceX.NET">
  <tns:GetCitiesByCountryResult>string</tns:GetCitiesByCountryResult>
</tns:GetCitiesByCountryResponse>

```

**Note:** IBM API Connect 2018.x was EOS after 30 April 2023. See [support policy](#) for details.  
 For a more recent version, see the [IBM API Connect 10.0.5.x and later product documentation](#).

---

## Example WSDL file: globalweather.wsdl

The following code sample contains a WSDL description.

The WSDL description file globalweather.wsdl is referenced by the example scenario in [Example: Using the platform REST APIs to publish a product containing a SOAP API](#).

```
<wsdl:definitions xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.webserviceX.NET" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://www.webserviceX.NET">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET">
      <s:element name="GetWeather">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="CityName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="CountryName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetWeatherResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetWeatherResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCitiesByCountry">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="CountryName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetCitiesByCountryResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetCitiesByCountryResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="string" nillable="true" type="s:string" />
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetWeatherSoapIn">
    <wsdl:part name="parameters" element="tns:GetWeather" />
  </wsdl:message>
  <wsdl:message name="GetWeatherSoapOut">
    <wsdl:part name="parameters" element="tns:GetWeatherResponse" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountrySoapIn">
    <wsdl:part name="parameters" element="tns:GetCitiesByCountry" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountrySoapOut">
    <wsdl:part name="parameters" element="tns:GetCitiesByCountryResponse" />
  </wsdl:message>
  <wsdl:message name="GetWeatherHttpGetIn">
    <wsdl:part name="CityName" type="s:string" />
    <wsdl:part name="CountryName" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherHttpGetOut">
    <wsdl:part name="Body" element="tns:string" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountryHttpGetIn">
    <wsdl:part name="CountryName" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountryHttpGetOut">
    <wsdl:part name="Body" element="tns:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherHttpPostIn">
    <wsdl:part name="CityName" type="s:string" />
    <wsdl:part name="CountryName" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetWeatherHttpPostOut">
    <wsdl:part name="Body" element="tns:string" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountryHttpPostIn">
    <wsdl:part name="CountryName" type="s:string" />
  </wsdl:message>
  <wsdl:message name="GetCitiesByCountryHttpPostOut">
    <wsdl:part name="Body" element="tns:string" />
  </wsdl:message>
  <wsdl:portType name="GlobalWeatherSoap">
    <wsdl:operation name="GetWeather">
      <wsdl:documentation xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
        Get weather report for all major cities around the world.
      </wsdl:documentation>
      <wsdl:input message="tns:GetWeatherSoapIn" />
      <wsdl:output message="tns:GetWeatherSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
```

```

        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Get all major
            cities by country name(full / part).</wsdl:documentation>
        <wsdl:input message="tns:GetCitiesByCountrySoapIn" />
        <wsdl:output message="tns:GetCitiesByCountrySoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="GlobalWeatherHttpGet">
    <wsdl:operation name="GetWeather">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
            Get weather report for all major cities around the world.
        </wsdl:documentation>
        <wsdl:input message="tns:GetWeatherHttpGetIn" />
        <wsdl:output message="tns:GetWeatherHttpGetOut" />
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Get all major
            cities by country name(full / part).</wsdl:documentation>
        <wsdl:input message="tns:GetCitiesByCountryHttpGetIn" />
        <wsdl:output message="tns:GetCitiesByCountryHttpGetOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="GlobalWeatherHttpPost">
    <wsdl:operation name="GetWeather">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
            Get weather report for all major cities around the world.
        </wsdl:documentation>
        <wsdl:input message="tns:GetWeatherHttpPostIn" />
        <wsdl:output message="tns:GetWeatherHttpPostOut" />
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Get all major
            cities by country name(full / part).</wsdl:documentation>
        <wsdl:input message="tns:GetCitiesByCountryHttpPostIn" />
        <wsdl:output message="tns:GetCitiesByCountryHttpPostOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GlobalWeatherSoap" type="tns:GlobalWeatherSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetWeather">
        <soap:operation soapAction="http://www.webserviceX.NET/GetWeather"
            />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <soap:operation soapAction="http://www.webserviceX.NET/GetCitiesByCountry"
            />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="GlobalWeatherSoap12" type="tns:GlobalWeatherSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetWeather">
        <soap12:operation soapAction="http://www.webserviceX.NET/GetWeather"
            />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <soap12:operation soapAction="http://www.webserviceX.NET/GetCitiesByCountry"
            />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="GlobalWeatherHttpGet" type="tns:GlobalWeatherHttpGet">
    <http:binding verb="GET" />
    <wsdl:operation name="GetWeather">
        <http:operation location="/GetWeather" />
        <wsdl:input>
            <http:urlEncoded />
        </wsdl:input>
        <wsdl:output>
            <mime:mimeXml part="Body" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <http:operation location="/GetCitiesByCountry" />
        <wsdl:input>

```

```

        <http:urlEncoded />
    </wsdl:input>
    <wsdl:output>
        <mime:mimeXml part="Body" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="GlobalWeatherHttpPost" type="tns:GlobalWeatherHttpPost">
    <http:binding verb="POST" />
    <wsdl:operation name="GetWeather">
        <http:operation location="/GetWeather" />
        <wsdl:input>
            <mime:content type="application/x-www-form-urlencoded" />
        </wsdl:input>
        <wsdl:output>
            <mime:mimeXml part="Body" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCitiesByCountry">
        <http:operation location="/GetCitiesByCountry" />
        <wsdl:input>
            <mime:content type="application/x-www-form-urlencoded" />
        </wsdl:input>
        <wsdl:output>
            <mime:mimeXml part="Body" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="GlobalWeather">
    <wsdl:port name="GlobalWeatherSoap" binding="tns:GlobalWeatherSoap">
        <soap:address location="http://www.websvcicex.com/globalweather.asmx" />
    </wsdl:port>
    <wsdl:port name="GlobalWeatherSoap12" binding="tns:GlobalWeatherSoap12">
        <soap12:address location="http://www.websvcicex.com/globalweather.asmx" />
    </wsdl:port>
    <wsdl:port name="GlobalWeatherHttpGet" binding="tns:GlobalWeatherHttpGet">
        <http:address location="http://www.websvcicex.com/globalweather.asmx" />
    </wsdl:port>
    <wsdl:port name="GlobalWeatherHttpPost" binding="tns:GlobalWeatherHttpPost">
        <http:address location="http://www.websvcicex.com/globalweather.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```