IBM API Connect 5.0.x

Tables of Contents

Note: A later version of IBM API Connect is available.
For details, see the IBM API Connect Version 10 product documentation.

IBM API Connect Version 5 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
- Toolkit command summary

See More

Develop APIs

- Implement an API with LoopBack
- Create an API that calls an existing endpoint
- Define paths for a REST API
- Example

See More

Secure and manage

- Work with Catalogs and Spaces
- Manage authentication
- Configure API security and authentication
- Secure APIs using OAuth 2.0
- Create an OAuth provider

See More

Discover and use APIs

- Register your applications
- Browse the Catalog and sign up to APIs

- Customize your Developer Portal
- Troubleshoot the Developer Portal

See More

Additional Reading

- Now available – new whitepaper on API Connect deployment!
- Now available on IBM Support TV – Separate development and production endpoints for APIs in IBM API Connect
- RFP Criteria for Choosing an API Management Solution
- IBM Webinar: The Total Economic Impact of an API Management Solution
  Presenters: Anish Shah and Christopher Schmitt
- An ESB is not API Management
  by Alan Glickenhouse
- API Connect Tutorial: Mastering the API Assembly
  by Kelly Hoover
- Creating and Securing an API with an OAuth Provider API
  by Mark Miranda and Sharath Srinivas
- APIs for the Insurance Industry-Avoid the risk of falling behind
  by Alan Glickenhouse
- API versus Services - what is the difference?
  by Claus Jensen

Getting help

Be sure to check out these extra help resources.

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

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect V5 overview

API Connect V5 is an integrated API management offering, where all of the steps in the API lifecycle, and the actions that surround it, are performed within the offering.

The steps of the API lifecycle include creating, running, managing, and securing APIs, as depicted in the following diagram.

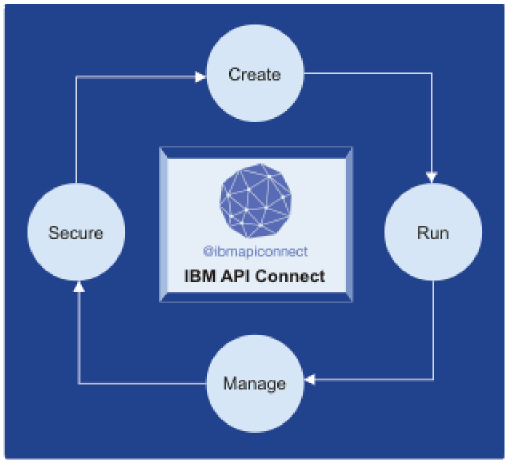


Table 1 summarizes the API lifecycle.

Table 1. Steps of the API lifecycle

Create	Develop and write the API definition and implementation, and test the API.
Run	Package and deploy the API. Ensure that the API is hosted securely on a stable platform.
Manage	Create and manage self-service portals that expose the API to API consumers. Monitor the set of rules and conditions that govern the API to ensure it is fulfilling its intended purpose, and make adjustments if necessary. Retire and archive the API when appropriate.
Secure	Incorporate access control, monitoring, and logging to properly secure the API.

With API Connect V5, you can perform all of the lifecycle steps in a single integrated offering, removing the requirement to use multiple API management offerings to obtain the same capability. API Connect V5 includes the following key capabilities to cover the lifecycle of an API:

- Automated, visual, and coding options that API providers can use to create scalable APIs
- Node.js and Java support for creating microservices applications and APIs with integrated tooling
- Integrated enterprise grade clustering, management, and security for Node.js and Java
- Lifecycle management and governance for APIs
- V5.0.8 + Set pricing details in plans to define revenue-producing subscription plans for your APIs
- Access control over APIs for both API providers and consumers by using role-based permissions, API packaging constructs, and subscription and community management
- Customizable, self service portals for publishing APIs for discovery and use
- Runtime enforcement of built-in and user-defined policies, and mechanisms to secure, control, and optimize API traffic
- API usage analytics for both API providers and consumers, with runtime and historical reporting on usage patterns and performance metrics

For information about the API Connect V5 components that provide these capabilities, and details about the defined strategy for packaging and publishing APIs for use by API consumers, see:

- Packaging strategy and terminology in API Connect
- API Connect components

You can learn more about API Connect V5 in the following topics:

- **What's new for this release**
  IBM® API Connect Version 5.0, and later releases, delivers new function over the IBM API Management Version 4.0 release.
- **Available deployment options of API Connect V5**
  API Connect V5 is offered as several deployment options, depending on your needs.
- **API Connect offerings**
  There are three subscription tiers of API Connect offerings, providing increasing capabilities.
- **API Connect V5 for IBM Cloud concepts**
  To help you get started, read about the API Connect V5 concepts and obtain a high level understanding of the API management solution.
- **Tutorials**
  You can find here tutorials for using IBM API Connect. The tutorials are divided between the developer toolkit, API Manager, and the Developer Portal.
- **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 IBM API Connect user interfaces is controlled through user roles, and the permissions that are assigned to those roles.
- **Reverting servers to stand-alone appliances**
  Describes how servers in the IBM API Connect cloud can be reimaged to function as a stand-alone appliance.
- V5.0.2 + **Define the main site in your API Connect cloud**
  By defining a main site in your API Connect cloud, you can ensure that your specified server configurations are preserved if a network

link between sites is interrupted.

- **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.
- **Preserve your cloud data**
  It is important to preserve your IBM API Connect cloud and data. Take regular backups by using the IBM API Connect command-line interface (CLI) command provided. Or, take snapshots by using your virtualization provider tools to provide a fallback if required.
- **The Command Line Interface**
  The IBM API Connect Command Line Interface (CLI) is available to administrators to manage the Management server and to maintain and update its configuration information.
- **IBM API Connect best practices**
  Consider implementing the following best practices for optimizing your IBM API Connect cloud.
- **API Connect V5 glossary**
  The glossary of API Connect V5 terms and definitions.
- **Accessibility features for 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

What's new for this release

IBM® API Connect Version 5.0, and later releases, delivers new function over the IBM API Management Version 4.0 release.

IBM API Connect provides a set of API capabilities that can be deployed on premises in your data center or on IBM Cloud. You can use IBM API Connect for defining, proxying, assembling, securing, and scaling APIs. IBM API Connect also provides detailed analytics and operational metrics. Use your company developer portal to provide links to social communities and manage applications that can be used by developers.

The IBM API Connect solution provides an intuitive user experience for managing the complete API lifecycle. From adding, publishing, and adopting APIs, to supporting, monitoring, and testing, IBM API Connect helps each company to realize the maximum value from their APIs.

IBM API Connect Version 5.0.0, and later releases, includes the following enhancements, release by release.

- Version 5.0.8
- Version 5.0.7
- Version 5.0.6
- Version 5.0.5
- Version 5.0.4
- Version 5.0.3
- Version 5.0.2
- Version 5.0.1
- Version 5.0.0

V5.0.8 +

Version 5.0.8

Migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04
  From API Connect Version 5.0.8.10 iFix 1, it is strongly recommended that you migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04, because support for Ubuntu V16.04 is being withdrawn in March 2021. For more information, see Migrating your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04.

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

From API Connect Version 5.0.8.10, 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

Detect illegal XML characters in API request headers

From API Connect Version 5.0.8.10, 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

From API Connect Version 5.0.8.10, 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.

Control whether client ID and scope are sent to a third party OAuth provider

From API Connect Version 5.0.8.8, a new **`allowed-parameters`** header enables you to control whether API Connect client ID and scope are sent to the third party OAuth provider. For more information, see OAuth introspection for third-party OAuth providers.

Access the caught exception in a catch block

From API Connect Version 5.0.8.8, a new **`getError()`** function enables you to obtain the details of the current caught exception in the **`catch`** block of an API assembly. A possible use would be to create a custom error response using the details of the caught exception. For more information, see GatewayScript code examples.

Set the maximum number of concurrent Gateway server additions

From API Connect Version 5.0.8.8, you can set a limit on the maximum number of Gateway servers that can be added to a Gateway service concurrently. In particular, this reduces the time taken to refresh Gateway servers after an upgrade. For more information, see Setting the maximum number of concurrent Gateway server additions.

Remove a Gateway server from a Gateway service

From API Connect Version 5.0.8.8, you can remove a Gateway server from a Gateway service whilst retaining it in your API Connect cloud. You can then easily re-add it a Gateway service in the future if required. For more information, see Removing and deleting servers.

Emulate the behavior of IBM API Management Version 4.0 when handling backend server errors

From API Connect Version 5.0.8.8, a new **`x-ibm-gateway-invoke-emulate-v4-invoke-error`** property is provided for emulating the IBM API Management Version 4.0 behavior when handling SOAP faults or JSON errors from a back end server, whereby a DataPower error is initiated. This property supersedes **`x-ibm-gateway-invoke-emulate-v4-soap-error`**, which is deprecated. For more information, see API properties.

Enable post processing of mapped JSON output from the Map policy

From API Connect Version 5.0.8.8, a new **`ibm-gateway-map-post-process-json-output`** API property allows you to enable the post processing of JSON output to ensure that property values are of the same data type as that defined in the schema, and that output property values that have a Badgerfish JSON syntax, due to object mapping of an XML input, are normalized. For more information, see API properties.

(5.0.8.7 iFix 4 or later) Add new certificates to your DataPower® Gateway servers.

You must complete this task once before upgrading to API Connect 5.0.8.7 iFix 4 (or later), to prevent the loss of analytics events data during the upgrade. For instructions, see Add certificates to gateways before upgrading API Connect.
If you skip this task, the upgrade will be successful but you will lose analytics event records spanning the time when the management servers start up at the upgraded level until each Gateway server is removed and re-added after the upgrade.

Attention: This is a one-time task and does not need to be repeated with subsequent upgrades.

Allow JSON payload to be accepted without parsing errors

From Version 5.0.8.7 iFix3, if an API request or response payload includes valid JSON content that contains characters that cannot be represented in the JSONX XML internal syntax that is used by the DataPower Gateway, set the **`x-ibm-gateway-api-json-parse-error-handling`** property to **`escape-unicode`** to allow the payload to be accepted without parsing errors. For more information, see API properties.

Specify which SOAP port to use when importing a WSDL service

From API Connect Version 5.0.8.7, the WSDL import options file has a **`port`** field to specify which SOAP port to use in the WSDL definition when creating an API by importing a WSDL service. For more information, see Using an options file when importing a WSDL service.

New option in the Map policy for setting input data log message severity

From API Connect Version 5.0.8.7, the Map policy has a new Severity level for input data log messages option to set the severity level of generated error messages that relate to input data. For more information, see Configuring the Map policy in the user interface.

New API property for the Map policy to control the generation of default values for required properties

From API Connect Version 5.0.8.7, an **`x-ibm-gateway-map-emulate-v4-default-required-properties`** API property is available for use with the Map policy that, when set to **`true`**, generates default values 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.

For full details, see API properties.

Removal of some commands from the developer toolkit CLI

From API Connect Version 5.0.8.7, the following commands are no longer supported in the developer toolkit CLI:

```
apic start
apic stop
apic services
apic props
apic microgateway
apic swiftserver
```

Include an options file when importing a WSDL service

From API Connect Version 5.0.8.6, 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. For more information, see Using an options file when importing a WSDL service.

Micro Gateway is deprecated in favor of DataPower Gateway

IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

Export the Management server configuration database in JSON format

From API Connect Version 5.0.8.5, a new **config dbextract** command is provided that exports the contents of the Management server configuration database in JSON format, in a .tar file. See Configuration commands.

Show and reset failed Developer Portal login attempts

From API Connect Version 5.0.8.5, you can run the **reset_locked_host** command to show you the IP addresses of all the clients that have made failed login attempts, on a per site basis. You can then clear the failed login attempts from specific or all IP addresses. See reset_locked_host and Flood control.

Additional JWT cryptographic algorithms

From API Connect Version 5.0.8.5, the Generate JWT policy supports the following additional cryptographic algorithms:

- PS256
- PS384
- PS512

See Generate JWT.

New API properties

From API Connect Version 5.0.8.5, the following new API properties are available:

- x-ibm-gateway-invoke-keep-payload: If set to `true`, the invoke policy sends a payload on an HTTP DELETE method.
- x-ibm-gateway-map-resolve-xmlinput-datatypes: If set to `false`, XML input elements are always mapped as a string. If you set to a value of `true`, numeric or boolean XML input elements are mapped as the corresponding data type from the input schema.
- x-ibm-gateway-map-xml-empty-element: Controls how the map policy handles XML input empty elements and impacts JSON output when the input document is XML.
- x-ibm-gateway-sourcecode-resolve-apic-variables: If set to `true`, API Connect variable references are resolved.
- x-ibm-gateway-schema-definition-reference-limit: Specifies the maximum allowed number of iterations of a circular schema definition.

See API properties.

Ability to disable automatic refresh of your Gateway servers after a gateway extension update

From API Connect Version 5.0.8.4, it is now possible to disable automatic refresh of gateway servers after a gateway extension update. Previous behavior was that an automatic refresh was performed after a gateway extension update. There is now an option under Gateway service settings called Automatically refresh extension on gateway servers. If you disable this option, then you can manually refresh the servers in the Gateway service rather than having the servers refreshed automatically. A manual refresh allows you to determine the timing and sequence of the updates in order to coordinate the activity with an external load balancer. Additionally, a manual refresh provides you with more control over potential downtime of API runtime traffic on gateway servers. See Configuring your Gateway server extensions.

Map input properties with null values

From API Connect Version 5.0.8.4, there is a new x-ibm-gateway-map-null-value API property for the `map` policy; setting the value of this property to `true` allows an input property with a value of `null` to be mapped to the output document. By default, an input property with a value of `null` is not mapped to the output document. See API properties.

Populate context variables for access by GatewayScript

From API Connect Version 5.0.8.4, there is a new x-ibm-gateway-custom-policy-with-gws-action property. If set to `true`, the `request.body` and `message.body` context variables will be populated for access by an `apim.getvariable('request.body')` or `apim.getvariable('message.body')` function call in a GatewayScript action of a custom policy. See API properties.

New error cases are supported by the assembly catch construct

From API Connect Version 5.0.8.3, the following new error cases are supported by the catch construct in an API assembly: BadRequestError, UnauthorizedError, and ForbiddenError. See Error cases supported by assembly catches.

Added the ability to remove analytics event fields from being collected to reduce storage requirements

From API Connect Version 5.0.8.3, you can reduce storage requirements by removing analytics event fields that you do not need to track. See Customizing the retained event record fields, Specifying the cloud settings, and API event record fields for more information.

Updated the query list and results for the detailed health check API to check for cloud dissociation

From API Connect Version 5.0.8.3, the **failIfCloudIsDissociated** parameter was added to make it easier to detect a cloud dissociation state by running the health check API. See Obtain health check data of Management servers by using REST API calls and Dissociation and your cloud for more information.

Added the **useBytesSent** query parameter to selected APIs

From API Connect Version 5.0.8.3, the **useBytesSent** parameter was added that allows the analytics field `bytes_sent` to be used to calculate the usage. See Return data usage information for all the resources used by a given application, Return data usage information for all the resources used by all applications in the given organization, Return combined data usage for all resources used by a given application, and Return combined data usage information for all the resources used by a given organization for more information.

Added the **stat show apiconfig** command to check the health of your Management server

From API Connect Version 5.0.8.3, the **stat show apiconfig** command returns information about your Management server. You can use this command to determine if your database is in good health before an upgrade, or run it regularly to ensure that your Management server is running correctly. See Testing Management servers for more information.

New isolate mode added to the **config load apiconfig** command for restoring your API Connect configuration

From API Connect Version 5.0.8.3 onwards, you can restore a previous version of your API Connect configuration in isolation mode. By using the `isolate` option on the **config load apiconfig** command, the management configuration file is loaded in isolation, in other words without any references to DataPower Gateway servers, Developer Portal servers, or any third-party systems for analytics offload. For more information, see Restoring an API Connect configuration.

You can encode "+" characters in the query parameter values of the target URL of an Invoke or Proxy policy

From API Connect Version 5.0.8.3, there is a new x-ibm-gateway-queryparam-encode-plus-char API property; if set to a value of `true`, all "+" characters in the query parameter values of the target-url of Invoke and Proxy policies are encoded to "%2F". In previous releases, "+" were always encoded to "%2F". Now, the default behavior is to **not** do the encoding. See API properties.

You can enforce the JSON parser on the response rule for an Invoke or Proxy policy

From API Connect Version 5.0.8.3, there is a new x-ibm-gateway-api-enforce-response-limits API property; setting this property to a value of true 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. See API properties.

Potential for performance improvement to the map policy

From API Connect Version 5.0.8.3, there is a new x-ibm-gateway-optimize-schema-definition API property that can provide a performance improvement to the map policy when a very complex schema definition is referenced by a policy output definition. See API properties

New API event field

From API Connect Version 5.0.8.2, the endpoint_url event record field identifies the proxy or invoke target URL on which the request failed. See API event record fields for more information.

Identifying and resolving an analytics split-brain condition in a cluster

From API Connect Version 5.0.8.2, you receive an email notification when your system identifies multiple Elasticsearch nodes as the master node. This is also known as an analytics split-brain condition. See Analytics split-brain for more information about identifying and resolving this condition.

Deleting user accounts and Developer organizations in the Developer Portal

From API Connect Version 5.0.8.2, you can delete your user account and Developer organizations in the Developer Portal. You can also change the ownership of your Developer organizations. For more information, see Deleting your Developer account, Deleting a Developer organization, and Changing the ownership of a Developer organization.

Obtain simple health check data of Developer Portal sites by using a REST API call

From API Connect Version 5.0.8.1, you can call a simple health check API to determine whether a particular Developer Portal site is working. This API is very 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. For more information, see Obtaining simple health check data of Developer Portal sites by using a REST API call.

View and select ciphers for TLS protocol versions used in TLS server profiles

From API Connect Version 5.0.8.2, you can view and edit the list of enabled ciphers for each version of the TLS protocol that is supported in a TLS profile. For more information, see Setting the ciphers for TLS Server profiles.

Secure individual APIs with TLS mutual authentication

From API Connect Version 5.0.8.1, you can secure individual APIs with TLS mutual authentication. An application that calls the API must supply a valid X509 certificate. For more information, see Composing a REST API definition.

Configure a Gateway service to use Server Name Indication (SNI)

From API Connect Version 5.0.8.1, you can use Server Name Indication (SNI) to specify which of two or more TLS profiles should be used depending on the host name. The SNI capability enables you to serve multiple endpoints through the same Gateway service without requiring them to use the same TLS certificate. For more information, see Configuring the initial Gateway service or Adding more Gateway services.

Specify multiple OAuth redirect URLs for your application in the Developer Portal

From API Connect Version 5.0.8.1, you can specify multiple URLs that authenticated OAuth flows for your application should be redirected to. For more information, see Registering an application.

Added support and a reference for Developer Portal REST APIs for analytics

Developer Portal REST APIs help you analyze your catalog APIs. For more information, see Analytics.

Added the Analytics section when creating an API

You can define and specify existing Parameters for your API that can be used to gather analytics data about the API. See Composing a REST API definition for more information.

Added the `logs` option to the **system clean** command

Specifying the `logs` option with the **system clean** command removes all of your log data from your server. For more information, see System commands.

Added the `analytics` option to the **system clean** command

Specifying the `analytics` option with the **system clean** command removes all of your analytics data from your server. For more information, see System commands.

Customize the number of replicas of your Elastic clusters

You can select automatic updating of the number of replicas, or specify a static number. See ../com.ibm.apic.cmc.doc/manage_organizations_idp.html#manage_organizations_idp for more information.

Encourage the use of two-factor authentication in the Developer Portal

You can encourage users of your Developer Portal to set up two-factor authentication (TFA) on their account by applying a TFA Rules module. For more information, see Encouraging users to set up two-factor authentication on their Developer Portal account.

Features added to the integrated billing and payment management

Administrator:

- Create monthly prepaid billing subscription Plans that your API customers can subscribe to with a credit card. See Billing for the use of your Products for more information.
- Leverage a Stripe account to manage the payments for your subscriptions.
- Specify a number of free trial days in your subscription Plan for new subscribers. Payment automatically begins after the trial days expire.

Customer:

- Subscribe to fee-based Plans in the Developer Portal that allow you to use Products that contain one or more APIs. See Tutorial: Subscribing to a Plan with pricing for more information.

Invoke automatically replaced in the gateway

The last invoke in your policy might be replaced by a proxy. This is done automatically by the gateway to improve performance. For more information, see: API properties.

The Linux distribution for the Developer Portal OVA is now based on Ubuntu Version 16.04

Support for Debian Version 7 is coming to an end in May 2018, so the Linux distribution for the Developer Portal OVA is now based on Ubuntu Version 16.04. For information about how to migrate your current Debian OVAs to the Ubuntu OVAs, see Migrating your Developer Portal OVAs from Debian V7 to Ubuntu V16.04.

New API event fields

Added the following API event fields:

- billing.trial_period_days
- billing.amount
- billing.currency
- billing.model
- billing.provider
- client_id
- immediate_client_ip
- latency_info2.task
- latency_info2.ended

See API event record fields and Obtaining analytics data by using REST API calls for more information.

New query parameters for the Redirect URL

New query parameters have been added to the information available for a third party. The new parameters are provider, providerid, and g-transid. For more information, see Authenticating and authorizing through a redirect URL.

OAuth scope can be modified by third-party responses

You can configure an external server to override the API scope value. For more information, see: Scope.

Preventing browser CORS alerts in the Test tool

The API Designer Test tool sends requests from the browser that can trigger CORS alerts. To prevent CORS alerts, the Enable Proxy check box is provided to send test messages from the server that hosts API Designer rather than from the browser. For more information, see Testing an API with the API Designer test tool.

Revoke single OAuth tokens

If you are using the DataPower Gateway, you can now revoke a single OAuth token for an application. For more information, see Creating an OAuth provider API.

Secure APIs with third party OAuth instead of Mobile First Foundation

Secure your API with a third-party OAuth provider instead of the IBM MobileFirst® Foundation authorization server. For more information, see Integrating third party OAuth provider.

Secure your APIs with OpenID Connect

You can secure your APIs with OpenID Connect(OIDC) by using a pre-supplied sample OAuth Provider API that you customize in accordance with your OIDC configuration. For more information, see Securing your APIs with OpenID Connect.

SOAP update action no longer overwrites the API

When you update a SOAP API from a WSDL definition, only those sections of the API that are affected by the new WSDL are replaced, the other sections are unchanged. In previous releases, the update action completely overwrote the configuration of the SOAP API definition, including all design properties and assembly configuration. For more information, see Updating a SOAP API.

Use Honeypot for spam protection in the Developer Portal

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. For more information, see Using Honeypot for spam protection.

Using the Views module in the Developer Portal

Create new views in the Developer Portal, such as content lists of Products, APIs, and applications, by using the Views UI module. For more information, see Using the Views module in the Developer Portal.

You can also follow a tutorial about creating a custom sort order view for a list of APIs; see Tutorial: Configuring a custom sort order view for APIs in the Developer Portal.

View cluster information by using Elasticsearch REST API calls

You can use Elasticsearch API calls to view a health status of red, yellow, or green for your identified clusters. For more information, see Obtaining cluster health information by using REST API calls.

V5.0.7+ Version 5.0.7

Added the **stat show apiconfig** command to check the health of your Management server

From API Connect Version 5.0.7.2, the **stat show apiconfig** command returns information about your Management server. You can use this command to determine if your database is in good health before an upgrade, or run it regularly to ensure that your Management server is running correctly. See Testing Management servers for more information.

Dynamically determine the health of a Developer Portal cluster

From API Connect Version 5.0.7.2, you can check the status of a Developer Portal cluster by calling a cluster health REST API. For more information, see Obtaining health check data of Developer Portal servers by using a REST API call.

Multilingual support of API and Product definitions

From API Connect Version 5.0.7.2, you can create multilingual API and Product documentation by using an `x-ibm-languages` extension directly in the OpenAPI (Swagger 2.0) definition. For more information, see Using `x-ibm-languages` to create multilingual API and Product documentation.

Integrated billing and payment management for your APIs

Starting with API Connect Version 5.0.7.2, API providers can use the monetization capability in API Connect to create pricing plans and set rate limits for their API products, collect payments from API consumers, and analyze the usage of their monetized and free API plans. Usage analytics can either be processed by using the integrated API Connect analytics tools, or by offloading them to an existing external system. Your consumers can subscribe themselves to plans, and have their payments made through a credit card processing provider. For more information, see the API Connect developerWorks blog To win in the API economy, you need a modern approach to API monetization.

XML Name Space attributes are in a different order from previous releases

Starting with API Connect Version 5.0.7.2 and beyond, users might notice that the order of XML Name Space (XMLNS) attributes in XML content in API requests and responses can differ from previous releases.

The XML specification https://www.w3.org/TR/xml/ does not suggest a preferred order for XMLNS attributes. Best practice is to not rely upon the sequence of XMLNS attributes if you write custom parsing code.

API Connect no longer allows external DTD/entity references while parsing XML.

From Version 5.0.7.1, IBM API Connect is secured to forbid external references while parsing XML. XML documents (such as custom forms, XML requests, or XML responses) being parsed by APIConnect Gateway will fail if there is a reference to an external URL. For more information, see the Tech note at "Forbidden external reference" error and controlling external DTD/entity references.

Analytics component has changed

The Analytics component is now built using the Kibana V5.1 open source analytics and visualization platform. As a result, there are some visual and operational changes to dashboards and visualizations. For a summary of the key changes, see The screen elements of a dashboard. Other changes are highlighted within the relevant procedures for the analytics tasks.

The event data that is generated in the API Connect on-premises cloud and displayed by the Analytics component can now be exported to third-party systems as a real-time data feed for centralized data consolidation, enhanced monitoring, and richer analytical data processing. The default ability to view and work with analytics data in the API Connect user interfaces is retained, but you can also now choose to disable access to analytics data within API Connect if preferred. For more information, see Configuring destination targets for API Connect analytics data.

Analytics email notifications triggered when data that is collected on the disk reaches predetermined levels

When the amount of Analytics data that is collected on the disk exceeds 70%, 80%, and 90% of the available disk space, an informational email is sent out at each level. See Adding a new data disk to a Management appliance for more information.

API Connect integrates with IBM Product Insights for viewing management and Developer Portal node resource usage.

You can view some usage resources for your API Connect management and Developer Portal nodes in the IBM Product Insights interface by registering your API Connect environment with IBM Product Insights. See Resource metrics collected by the IBM Cloud Product Insights service for more information.

API Designer and API Manager have a new look

The API Designer and API Manager user interfaces have been restyled based on the Carbon design system. This change affects only their "look and feel," not functionality.

Application lifecycle workflow

By using the application lifecycle capability, you can have separate Development and Production endpoints for the same API. Applications that are subscribed to use the API initially have Development status, and can call the API only through Development endpoints. When application testing is complete, the application developer can request to upgrade the application to Production status; when the request is approved, the application is upgraded and can call the API through Production endpoints. For more information, see Managing the application lifecycle.

Application metrics dashboard is now available for Node.js applications

When you run a Node.js application (such as a LoopBack project) locally using the Developer Toolkit, you can view application performance metrics using the built-in application metrics dashboard. For more information see, Viewing the application metrics dashboard.

Catalog supports multiple DataPower Gateway services

You can configure a Catalog to use two or more DataPower Gateway services. Then by modifying the Gateway service endpoints, and configuring your DNS appropriately, you can route API calls to the required Gateway service. For more information, see Using multiple DataPower Gateway services with a Catalog.

Collectives are deprecated in favor of Docker Swarm and Kubernetes managed containers

IBM API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.

New customers should not install API Connect collectives because this feature is no longer supported for new users.

Command-line tools now work with management server running on ports other than default 443

If you change the TCP port number on which the API Management server listens, the `apic` command-line tool will now work properly if you specify the port with the command-line `--server` option.

Developer toolkit supports API testing with the DataPower Docker container

When you test API from the Developer toolkit, you can now set an option to use the DataPower Gateway Docker container for a full set of security and policy capabilities. The toolkit synchronizes with the Gateway on save; you can now test product and plan level concepts; DataPower Gateway error logging and Request/Response logging are also integrated into the API Designer logging console.

Developer toolkit supports API testing with the special `apic-dev` Catalog name.

When you test API from the Developer toolkit, you can now use the special `apic-dev` Catalog to substitute assembly properties at run time. This behavior is adapted from the API Manager component. See, Configuring API definitions for container run times, at Migrating LoopBack applications from collectives to containers for how to configure this feature.

Developer toolkit supports vendor extensions

API Designer now supports OpenAPI (Swagger 2.0) extensions (also referred to as "vendor extensions"). For more information, see Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Designer UI). The command-line tool `apic extensions` command is also available for working with extensions. For more information, see Toolkit command summary and Extensions commands.

JSON Web Token (JWT) can now be used to secure your API

You can now secure your API with JSON Web Tokens in two ways. You can use the **jwt-generate** policy or you can use a token that was generated external to IBM API Connect.

LoopBack 3.0 is now supported by API Designer and command-line tools

When you create a new LoopBack project with the API Designer or `apic loopback` command, you now have the option of creating a LoopBack version 3.0 project. For more information on LoopBack 3.0, see loopback.io.

OAuth shared secret can be provided by the end user, or randomly generated

The default OAuth shared secret used by API Connect can be customized. For more information, see Adding a gateway server.

OAuth integration with third-party providers

IBM API Connect can be configured to use a third-party for authentication and or authorization in compliance with the OAuth 2.0 specification: https://tools.ietf.org/html/rfc7662. For more information, see Creating an OAuth security definition.

An additional header, `x-Introspect-`, is provided for passing additional information to a third party provider. For more information, see Integrating third party OAuth provider.

New OAuth query parameters

Six new OAuth query parameters have been introduced.

```
appid = application id
org = organization name
orgid = organization id
catalog = catalog name
catalogid = catalog id
transid = transaction id used in the Gateway
```

For more information, see:Authenticating and authorizing through a redirect URL

Maximum consent control

Use maximum consent to specify for how many seconds the combination of any number of access and refresh token remain valid. For more information, see: Creating an OAuth provider API.

SNI support for the management traffic between API Connect and DataPower Gateway

To inject Server Name Indication (SNI) in communications between IBM API Connect and a DataPower Gateway, you set the hostname (rather than IP address). For more information, see Adding a Gateway server.

Support for Node.js V6 added

IBM API Connect now supports Node.js V6.x.

V5.0.6 + Version 5.0.6

Added the **stat show apiconfig** command to check the health of your Management server

From API Connect Version 5.0.6.6, the **stat show apiconfig** command returns information about your Management server. You can use this command to determine if your database is in good health before an upgrade, or run it regularly to ensure that your Management server is running correctly. See Testing Management servers for more information.

(Technical preview) Build an IBM API Connect environment in a Docker container

By installing IBM® API Connect in a Docker container, you can run a complete IBM API Connect on-premises environment on your local machine. A Docker container installation of IBM API Connect is for development use only, it is **not** supported in a production environment. For more information, see Installing and configuring IBM API Connect in a Docker container.

(Technical preview) Create applications in the Swift programming language

You can create applications in the Swift programming language by using Swift Server Generator. Swift Server Generator provides developer toolkit commands for creating Kitura Swift applications based on data models that you define and attach to a data source. A full set of REST APIs for working with the back-end data is generated automatically.

Note: Support has been removed from Version 5.0.8.7.

Categorize APIs and Products in IBM API Connect

You can define categories for APIs and Products in the API Designer or API Manager UI, and have the option to expose them in the Developer Portal.

You can also configure taxonomies for your APIs and Products in the Developer Portal.

For more information, see Organizing your APIs and Products into categories and Displaying APIs and Products in categories.

Creating and configuring Rules in the Developer Portal

You can configure Rules to perform specific actions when they are triggered by specific events in the Developer Portal. For more information, see Rules in the Developer Portal.

Including metadata in the OAuth transaction

You can include arbitrary information as metadata during the OAuth authentication handshake. When the Metadata URL is configured, IBM API Connect sends a request header to the URL and stores the response in the token or payload containing the token. For more information, see OAuth metadata.

Enabling OAuth debugging support

You can activate debugging for OAuth that produces a more detailed report than just an error message. For more information, see Troubleshooting OAuth.

Testing OAuth 2.0 with the Developer Portal test tool

The testing tool in the Developer Portal supports the testing of OAuth 2.0 interactions. For more information, see Troubleshooting OAuth.

Disabling Server Name Indication (SNI)

The TLS extension, SNI, is enabled by default. Servers that do not support SNI typically ignore the extension if it is included, but in some situations compatibility issues can prevent connection. You can disable SNI with a toggle in the TLS profile. For more information, see TLS profiles.

SSLClientProfile and SSLServerProfile replacing SSLProxyProfile

Forward SSLProxy (and Crypto) is replaced with SSLClient. These new profiles support ephemeral ciphers (DHE and ECDHE), perfect forward secrecy, and Server Name Indication (SNI) extension. Note that DHE ciphers in DataPower SSLServerProfile use 2048-bit DH parameters (as server) and accept 1024-bit DH parameters (as client).

(V5.0.6.2 and later releases) Conversion of non-ASCII characters in XML bodies

Non-ASCII characters (above U+007f) in XML bodies are no longer converted to numeric character references.

Policy properties introduced

(Version 5.0.6.2 and later releases) One new policy property has been introduced to maintain feature availability. Previously, invoke policies were URL-decoded by default. The new behavior is to not decode by default. For examples and a list of invoke policy properties, see API properties.

(Version 5.0.6.3 and later releases) Two new properties for the invoke and proxy policies have been introduced to control suppression of the X-IBM-Client-Id HTTP header and, in the case of the proxy policy, the `client_id` query parameter in the request URL. In previous releases, the client ID parameter was always suppressed. For more information about the invoke policy and proxy policy properties, see API properties.

V5.0.5 + Version 5.0.5

Use the new syndication feature to partition your Catalogs

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. For more information, see Using syndication in IBM API Connect.
New developer toolkit CLI commands are provided to support the creation and management of Spaces, and there is a new `space` configuration variable. For more information, see Toolkit command summary.

Advanced XML options

You now have greater control over the namespace declarations in XML output of the map policy. For more information, see The map policy structure.

New Generate LTPA Token built-in policy

Lightweight Third Party Authentication (LTPA) is an IBM protocol that provides a cookie or binary security token based authentication mechanism in WebSphere® Application Server. Apply the Generate LTPA Token policy to your assembly so that your API can securely authenticate with applications or services that are hosted on WebSphere Application Server. Use the API Manager UI to import an LTPA key, and then apply a Generate LTPA Token policy to generate a Lightweight Third Party Authentication (LTPA) token.
For more information, see LTPA keys and Generate LTPA token policy.

Analytics enhancements

In the API Manager UI, the Analytics component includes the following updates for the syndication feature:

- The Analytics permission is now Catalog-based rather than organization-based, and includes support for two separate actions: View (which provides read-only access) and Manage (which provides write access). The ability to access and work with analytics data at a Catalog or Space level will depend on the roles you are assigned and the type of Analytics permission defined for those roles.
- An "inheritance" flow is defined for the dashboards and visualizations in a Catalog and its Spaces. This flow determines whether updates made to the dashboards and visualizations in a Catalog are reflected in a Space, and affects what you see when you attempt to edit, delete, or restore default dashboards or visualizations, or when you attempt to create, edit, or delete custom dashboards or visualizations.
For more information, see Analytics and syndication.

Customizations to the default dashboards or visualizations can now be reversed by using the restore feature to reset your changes. For more information, see Restoring the default dashboards and Restoring the default visualizations.

V5.0.5 ONLY While creating or editing a dashboard, the workflow has been improved to enable you to seamlessly create and add visualizations to the dashboard during the process. For more information, see Creating custom dashboards and Editing dashboards.

In the Cloud Manager, analytics data can now be accessed for the individual servers in the Management and Gateway services. For more information, see Monitoring the health of the individual servers.

OAuth support for test tools

The test tools in the Developer Portal, and the API explorer and assembly console that are found in the API Manager API Designer UIs now support OAuth. The test tools can act as full OAuth clients, which enables the complete testing of APIs that are secured with all of the OAuth2 flows.

Adding custom pages to APIs and Products

You can add any custom pages that you have created to any APIs and Products that exist in the Developer Portal. By adding custom pages to APIs and Products, you can include additional information to APIs and Products that might improve their use and implementation. For more information, see Add custom pages to APIs and Products.

Open API formData support for the Developer Portal test tool

The test tool in the Developer Portal now supports the use of formData in Open API documents.

Reuse code fragments in OpenAPI (Swagger 2.0) files

You can use the `$ref` field in your OpenAPI (Swagger 2.0) API definition files to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file. When IBM API Connect processes the source API definition file, the `$ref` field is replaced with the contents of the target file. For more information, see Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.

New toolkit commands to view and list subscriptions

The `apic subscriptions` and `apic subscriptions:get` commands list subscriptions in a product, application, or a Catalog and display information on a subscription, respectively. For more information, see Toolkit command summary.

New toolkit command to list members of an organization

The `apic members` command lists members of an organization. For more information, see Toolkit command summary.

Configure plan, rate-limit, and TLS profiles in Micro Gateway Datastore

Developers are now enabled to configure plan, rate-limit, and TLS profiles in the Micro Gateway Datastore for a better development experience.

Configure writable LDAP in the Developer Portal

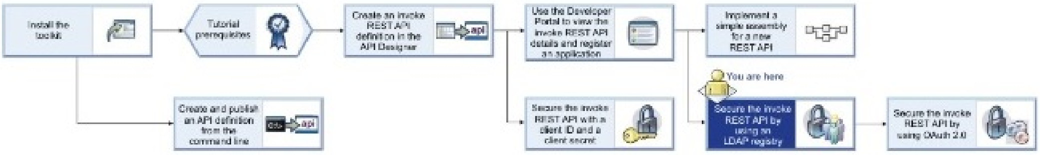
You can configure writable OpenLDAP in the Developer Portal if you already have an existing LDAP and want to include additional users.
For more information, see Configuring writable LDAP in the Developer Portal.

Obtaining metrics data for your LoopBack applications

You can monitor your LoopBack® applications by obtaining metrics data. You can send the metrics data to a variety of logging destinations. For more information, see Obtaining metrics data for your LoopBack applications.

New tutorial flow diagrams

Each developer toolkit tutorial displays a tutorial flow diagram to make it easier for you to follow the tutorials in the correct sequence.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
You are here
Create and publish an API definition from the command line
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

When you are on a tutorial page, you can click a tutorial in the diagram to open that tutorial directly. You can access the developer toolkit tutorials at Developer toolkit tutorials.

V5.0.4 + Version 5.0.4

Gateway support for custom branding

When you implement custom branding, you no longer need to add a component to map the URL. URL mapping is no longer required because the gateway detects the Catalog based on the incoming host name. For more information, see Creating and configuring Catalogs.

Advanced XML options

You now have greater control over the XML output of the map policy. You can control empty elements, and inherited namespaces. For more information, see The map policy structure.

Secure your APIs with IBM MobileFirst Foundation

You can now secure your IBM API Connect APIs by using the IBM MobileFirst Foundation authorization server.

Ability to view and export API event data from Analytics

From the API Manager user interface, you can 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. The event data that you export is saved to a comma-separated values (CSV) file. For more information, see Viewing and exporting analytics and event data.

Toolkit CLI accessibility mode

Developer toolkit accessibility mode makes the product easier to use for those with limited eyesight. To enable accessibility mode, set the `accessibility-mode` configuration variable to `enabled`. In this release, when you enter the `apic edit` command in accessibility mode, the tool prompts whether you want to open the API Designer in your web browser. For more information about setting configuration variables, see Using configuration variables.

New CLI commands

Two new `apic` commands were added: `apic orgs:get` and `apic devapps`.
For more information, see Toolkit command summary.

Automatic subscription support with the Micro Gateway

You can now enable Automatic subscription for a Catalog that uses the Micro Gateway, in addition to the DataPower Gateway. Enabling automatic subscription makes testing of your APIs in the API Manager user interface easier because a test application is used, with a pre-supplied client ID and client secret, which is automatically subscribed to all the Plans in the Catalog. As a result, you don't have to specify a plan or application when testing. For more information, see Creating and configuring Catalogs.

Ability to create an API and Product definition from a custom template using the API Designer

In the API Designer, you can now create a new API or Product definition from a custom Handlebars template file. For more information, see Composing a REST API definition and Creating a Product in the API Designer.

Link checking in the Developer Portal

You can now periodically check for any broken links in your Developer Portal. For more information, see Checking links in the Developer Portal.

Default language for code snippets in the Developer Portal

Any user of the Developer Portal can select the default programming language that their code snippets are displayed in. For more information, see Selecting the default code snippet language.

V5.0.3 + Version 5.0.3

The built-in validate policy is now available on the Micro Gateway

You can now use the validate policy with the Micro Gateway to validate the payload in an assembly flow against a JSON schema. For more information, see validate.
Note: You can continue to use the validate policy with the DataPower Gateway to validate the payload in an assembly flow against a JSON or an XML schema.

OAuth introspection endpoint

You can now add an introspection operation to an OAuth provider API. This new endpoint allows applications to present an OAuth access token and receive information about the access token in the response. For more information, see Creating an OAuth provider API.

Enhanced graphical user interface support for arrays and inline schema in the map policy

A graphical method for creating new inline schemas is now available for the map policy, enabling you to easily create schemas in the map policy that are not exposed to the users of your API. Additionally, support for iterating over different levels of arrays is provided when configuring a particular mapping without editing the OpenAPI (Swagger 2.0).

More detail in the debug view of the API Manager test tool

DataPower Gateway only Additional debug information, such as the input and output of the policy, is available for the invoke, map, and proxy policies.

New automatic subscription mode for a Catalog

DataPower Gateway only In the API Manager user interface, you can now enable Automatic subscription for a Catalog. Enabling automatic subscription makes testing of your APIs in the API Manager user interface easier because a test application is used, with a pre-supplied client ID and client secret, which is automatically subscribed to all the Plans in the Catalog, so you don't have to specify a plan or application when testing. For more information, see Creating and configuring Catalogs.

Admin guide in the Developer Portal

An admin guide is available in the Developer Portal to administrator accounts, only. The admin guide include information that ranges from basic configuration of the Developer Portal, to managing security and users. The information in the admin guide contains information from the Knowledge Centre.

Code snippet enhancements in the Developer Portal

You can choose which languages can be used to display code snippets in the Developer Portal. C and C# are also added to the collection of languages that you can enable to become available. For more information on configuring the languages that are available for code snippets, see Enabling code languages for code snippets. For more information on code snippets, see Browsing available APIs.

Creating and applying Rules in the Developer Portal

You can create rules in the Developer Portal which automatically trigger actions in response to situations or other actions. By creating rules, you can automate and anticipate responses to situations, which can help provide a more personalized and efficient user experience. For more information, see Applying rules in the Developer Portal.

API Designer can now create new LoopBack and OpenAPI projects

You can create new LoopBack and OpenAPI projects directly within the API Designer. For more information, see Creating new projects in the API Designer

The apic login command has a new `--sso` option

The `--sso` option enables you to login to IBM API Connect cloud using federated corporate ID.

Terminology changes

IBM API Connect Version 5.0.3 introduces the following terminology change:

Previous term	New term
Sandbox Catalog	Development Catalog Note: The title of the pre-supplied default development Catalog remains as "Sandbox".

V5.0.2 + Version 5.0.2

Using the Portal Delegated User Registry in the Developer Portal

By enabling the Portal Delegated User Registry in the API Manager UI, you can improve the flexibility of user registry and account management using the additional configuration options that are available in the Developer Portal. For more information, see Portal Delegated User Registry.

You can use the log in credentials that are used with external authentication providers (such as Facebook, Google, or Twitter), to access the Developer Portal. Using external authentication provider credentials reduces the number of authentication credentials that a user has. For more information, see Using external authentication provider credentials to access the Developer Portal.

Your Developer Portal site can manage users and user registries. Therefore, you can configure your LDAP user registry in the Developer Portal. For more information, see Configure your LDAP user registry in the Developer Portal.

You can modify the templates that are used for the emails that are sent by the Developer Portal, by altering the content and tokens that the emails use from within the Developer Portal. For more information, see Modifying the Developer Portal email templates.

Enhancements to the OpenAPI (Swagger 2.0) extension capability

The capability to add OpenAPI (Swagger 2.0) extensions to your APIs has the following enhancements:

- You can now add extensions to your local API definitions by using the API Designer user interface, in addition to the API Manager user interface.
- You can replace an extension with an updated version.
- The schema definition file for the extension is in YAML format rather than JSON format.

For more information, see Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Designer UI) and Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Manager UI).

Uploading a WSDL file is now supported in the API Designer

You can now create a SOAP API in the API Designer user interface by uploading a WSDL file. You can upload the file either from your local file system or from a URL.

For more information, see Adding a SOAP API definition.

Defining the main site in your API Connect cloud

By defining a main site in your API Connect cloud, you can ensure that your specified server configurations are preserved if a network link between sites is interrupted. For more information, see Define the main site in your API Connect cloud.

Apply multiple burst limits and multiple rate limits to your Plans and operations

You can now set multiple rate limits per Plan and per operation, at second, minute, hour, day, and week time intervals.

DataPower Gateway only You can also 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.

For more information, see Working with Products in the API Designer.

New built-in WS-Security policy: validate-usernametoken

Apply the validate-usernametoken policy to your APIs to validate a Web Services Security (WS-Security) UsernameToken in a SOAP payload, before allowing access to a protected resource. For more information, see validate-usernametoken.

Using templates to create APIs and Products

Using the CLI, you can create API and Product definitions from templates. 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. For more information on using templates, see:

- Creating and using API and Product definitions templates.
- Template variables for API and Product definitions.
- API and Product definition template examples.

API Designer can discover models from relational databases

You can use API Designer to create models corresponding to existing database tables. This process is called *discovery* and is supported by data source connectors for: MySQL, Oracle, PostgreSQL, and SQL Server. For more information, see Discovering models from relational databases.

API Designer can create and update a database schema based on LoopBack models

You can use API Designer to create and update a database schema based on your models, for MongoDB, MySQL, Oracle, PostgreSQL, and SQL Server connectors. This enables you to develop your models first, and create (and update) your database schema to match them. For more information, see Creating database schema from models.

You can add an existing LoopBack or OpenAPI project to the API Designer.

Once you add a project, you can then edit it with API Designer and you can switch between multiple projects. For more information, see Adding an existing project to API Designer.

V5.0.1+ Version 5.0.1

Managing disks on Management appliances

You can now protect your data by encrypting the hard drives on your Management servers. For more information, see Disk encryption.

When you upgrade a Management appliance to IBM API Connect Version 5.0, or install IBM API Connect Version 5.0 onto a new appliance, the amount of swap space and code disk size that is required is greater than in previous versions. For more information on the swap space and code disk requirements, see Swap space allocation and Increasing code disk size for appliances.

New JSON Web Token (JWT) built-in policies

JWT is a compact, URL-safe way of representing claims that are transferred between two parties. IBM API Connect now includes the following two built-in security policies that you can apply to your APIs:

- jwt-generate

  Use the jwt-generate policy to generate claims and configure whether they are to be used as the payload of a JSON Web Signature (JWS) JSON structure, or as the plain text of a JSON Web Encryption (JWE) JSON structure. For more information, see jwt-generate.

- jwt-validate

  Use the jwt-validate policy to enable the validation of a JSON Web Token (JWT) in a request before allowing access to the APIs. For more information, see jwt-validate.

Workbench moderation in the Developer Portal

You can use a dashboard to manage the review and approval process for content types in the Developer Portal. You can specify which roles can access the workbench dashboard by assigning them the appropriate permissions. For more information, see Configuring workbench moderation.

New OAuth token management system through DataPower

Manage the revocation of an OAuth 2.0 access token by using DataPower. The list of revoked access tokens is shared across your gateway cluster and access by using REST APIs, configured in your OAuth provider API.

BETA Using external authentication provider credentials to access the Developer Portal

You can use the log in credentials that are used with external authentication providers, to access the Developer Portal. Using external authentication providers credentials reduces the number of authentication credentials that a user has. For more information, see Using external authentication provider credentials to access the Developer Portal.

Web service invocation within the assembly of a REST API

You can add an existing web service to your REST API definition and then use it in your assembly, where the WSDL file of your web service is used to generate map policies to manage the invocation of the web service.

V5.0.0+ Version 5.0.0

Offline developer experience with the developer toolkit

The developer toolkit provides all users with an offline developer experience. The offline developer experience enables the user to create their APIs with the API Designer visual editor, run tests locally, manage APIs and any security policies, ready for publishing to an appliance or to IBM Cloud in the future. The APIs can also be tested by running them through the Micro Gateway in an end-to-end flow. A command line environment is also provided. You create local definition files for your APIs and then use either the editor or the toolkit commands to interact with API Manager.

For more information, see Developing your APIs and applications.

Visual editor for composing APIs

The Assemble view in the new API Manager user interface provides a visual tool for composing API assembly flows. You drag and drop components from a palette into your API assembly diagram. Additionally, a Code view provides an OpenAPI (Swagger 2.0) editor. Any changes that you make to your assembly diagram in the Assemble view are reflected in the Code view; similarly, if you change the OpenAPI (Swagger 2.0) code directly, the assembly diagram is updated accordingly. For more information, see The assemble view and The source view.

Important: Due to a change in the OpenAPI (Swagger 2.0) specification, API definitions created before the first fixpack will not pass validation upon staging of their containing Product. For information on editing your API to rectify the validation error, see Composing a REST API definition.

Node based Micro gateway

Included in the offline developer experience is the Micro Gateway. The Micro Gateway is a node.js based gateway that builds on StrongLoop® technology, and is packaged and available through npmjs.org as one of the components of the apiconnect package. The Micro Gateway receives requests, processes them as defined in the assembly, and invokes the back-end API all on a laptop. The policies available in the Micro Gateway are a subset of those available in the DataPower Edge Gateway.

Create and run

In addition to managing and securing APIs, you can now create and run APIs in IBM API Connect by using the LoopBack capabilities that are included in the single integrated offering. For more information, see Working with LoopBack projects.

Redesigned API Manager user interface

The API Manager user interface has been redesigned to enhance the API management experience. For full details about how to use the API Manager user interface, see Managing your APIs.

Improved visualization for analytics

You can create custom analytics dashboards for your Catalogs through API Manager, which consist of default or user created visualizations such as tables, graphs, and maps. The analytics performance has also improved. For more information, see API Analytics.

APIs are published in a Product

APIs are now published inside a *Product*. Products provide a method by which you can group APIs into a package that is intended for a particular use. Within a Product, APIs are contained in Plans that can be used to differentiate between offerings, and enforce rate limits. For more information, see Working with Products in the API Designer and Working with Products in the API Manager.

Removal of the Basic Developer Portal

The Basic Developer Portal has been removed from IBM API Connect. The Advanced Developer Portal is renamed to the Developer Portal, and is available to everyone.

The Developer Portal provides additional features including forums, enabling application developers to find and use APIs, blogs, comments, and ratings, together with an administrative interface for customizing the Developer Portal.

New features for the Developer Portal (formerly known as the Advanced Developer Portal)

- Improved responsive design layout for the Developer Portal.
- New social block in the Developer Portal for forum posts and tweets. For more information, see Integrating Twitter data into the social block.
- New features for the API/Product block
- You can uninstall themes or module files from the server by using the new Comprehensive Uninstaller.

- You can increase the specificity of a Developer Portal site URL by using sub paths. For more information, see Sub paths for the Developer Portal sites.

For more information, see Developer Portal.

Custom OAuth forms
You can now implement custom forms for the sign-in and authorization stages of the OAuth security flow. For more information, see Creating a custom sign-in form and Creating a custom authorization form.

New built-in policies
The following additional built-in policies are provided, which you can optionally apply to your APIs:

invoke
Call an existing service from within an operation.

if
Execute a section of the assembly when a condition is fulfilled.

throw
Configure errors returned by your API.

map
Transform variables with enhanced capabilities compared to previous versions of API Management.

xml-to-json and json-to-xml
Convert between xml and JSON schemas by using the badgerfish protocol.

activity-log
Log fields during an API call.

xslt
Perform an xslt transformation.

gatewayscript and javascript
Include a GatewayScript or JavaScript program in your assembly.

redact
Remove unwanted or confidential fields during an API call.

validate
Validate the payload in an assembly flow against a schema

set-variable
Set a runtime variable to a string value, or add or clear a runtime variable

operation-switch:
Execute sections of the assembly depending the operation that is called.

For further details, see Built-in policies.

REST authentication for product APIs under IBM ID
If you are using IBM API Connect for IBM Cloud (the SaaS offering), you can now provide your IBM ID credentials when you make REST requests to the API Manager or Developer Portal. This is useful when you attempt to automate the creation and management of applications. For more information, see Developer Portal REST APIs and Obtaining analytics data by using REST API calls

Upgrading the Management server
Effective with this release, it is important to take a virtual machine snapshot prior to upgrading the Management servers (and not only of the API Management configuration backups). Otherwise, you might be unable to restore the API Management configuration backup directly to the Management servers and would need to set up a new virtual machine.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Available deployment options of API Connect V5

API Connect V5 is offered as several deployment options, depending on your needs.

V5 Public cloud

This format leverages the advantages of hosting the services on the public IBM® Cloud architecture. The V5 Public Cloud edition of API Connect includes the following key features:

- No additional hardware requirements.

- No special firewall requirements.
- Easy to pair with other IBM Cloud services.

V5 Reserved instance

Reserved Instance is a separate API Connect deployment for a single customer that is based on the topology and functionality of the IBM Cloud public cloud service. Though the V5 Reserved instance is for a single customer, it is not custom and follows a standard configuration.

The V5 Reserved instance format is set up specifically to meet your organization’s needs and includes the following key features:

- Common login with other IBM Cloud services using IBMid.
- Isolation from others who are using the public service.
- Managed, monitored, and operated by the API Connect operations team.
- Deployed across multiple servers within the datacenter for resilience.
- Deployment across multiple data centers is an available option for an additional cost.

V5 On-premises

The on-premises deployment provides an installed version of API Connect V5 with 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.

V5 Comparison table

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

Feature	V5 Public Cloud	V5 Reserved instance	V5 On-premises
Managed by IBM operations	Yes	Yes	No
Shared gateways	Yes	No	No
Dedicated gateways	No	Yes	Yes
Remote gateways	No	Yes	Yes
VPN connectivity	No	No	Yes
Single-DC HA	Yes	Yes	Yes
Multi-DC HA	No	Available at an extra cost	Yes
Custom branding	Yes	Yes	Yes
CMC access	No	No	Yes
API Manager access	Yes	Yes	Yes
DataPower UI access	No	No	Yes
User-defined policies	No	Yes*	Yes
Gateway log offload	No	Yes*	Yes
Analytics offload	No	Yes*	Yes
Custom extensions	No	Yes*	Yes

* If any custom code is required for these features, IBM agrees to apply code (supplied by you) to your V5 Reserved instance of API Connect; however, IBM cannot write or maintain custom code in the SaaS environment.

- **API Connect V5 Reserved Instance**
  The IBM API Connect V5 Reserved Instance offers a dedicated API Connect instance that runs on IBM-managed infrastructure. V5 Reserved Instance provides value by balancing the flexibility of a shared infrastructure with the isolation of an API Connect Reserved Instance. It is a single-tenant deployment that is based on the topology and functionality of the API Connect on IBM Cloud Public.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect V5 Reserved Instance

The IBM® API Connect V5 Reserved Instance offers a dedicated API Connect instance that runs on IBM-managed infrastructure. V5 Reserved Instance provides value by balancing the flexibility of a shared infrastructure with the isolation of an API Connect Reserved

Instance. It is a single-tenant deployment that is based on the topology and functionality of the API Connect on IBM Cloud Public.

Key benefits

The API Connect V5 Reserved Instance has the following benefits that might suit your needs:

Accelerated time to value
: IBM deploys and configures your API Connect V5 environment to scale to your needs.

More control than public cloud
: You can create user-defined policies, configure syslog targets for DataPower logs, and use you own DNS/SSL configuration.

Optimize your IT spending
: There are fewer upfront capital expenses. The monthly subscription plan includes deployment on best-in-class infrastructure, maintenance and updates, routine backups, and disaster recovery.

Pricing structure

Some of the details of the monthly pricing structure include the following items:

Price structure
: Tier subscription that is based on the number of annual API calls

Billing metric
: Number of millions of API calls during a year

Billing cycle
: Annual (12 months)

Prices
: Contact your IBM sales representative for current prices

Convenience and security considerations

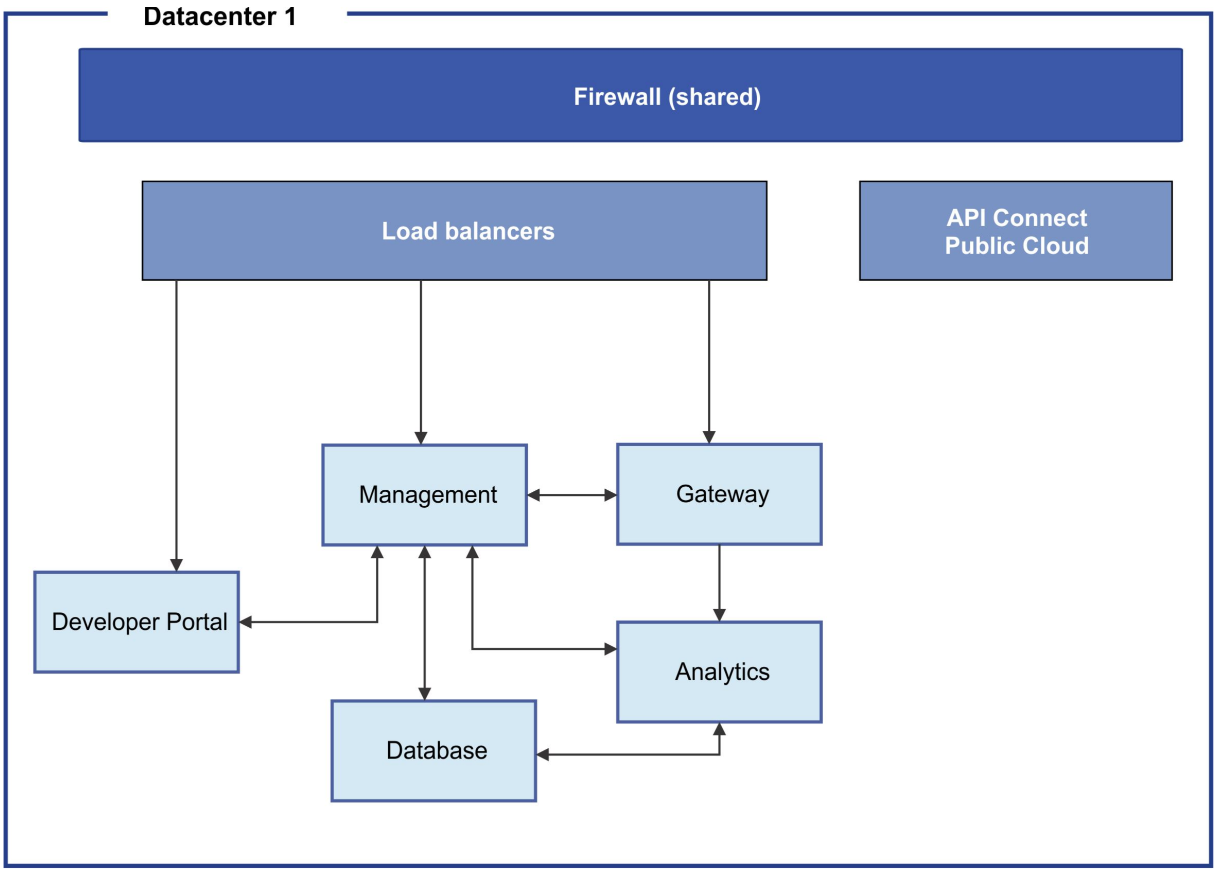
- Common log in with other IBM services that use IBMid
- Isolation from other users of the IBM Cloud public service
- Managed, monitored, and operated by the API Connect Operations team
- Deployed across multiple pods within the datacenter for resilience.
- Dual datacenter topology is available (separately charged)

Architecture

You can configure API Connect V5 Reserved Instance with a single data center, or with dual data centers with the purchase of an additional part.

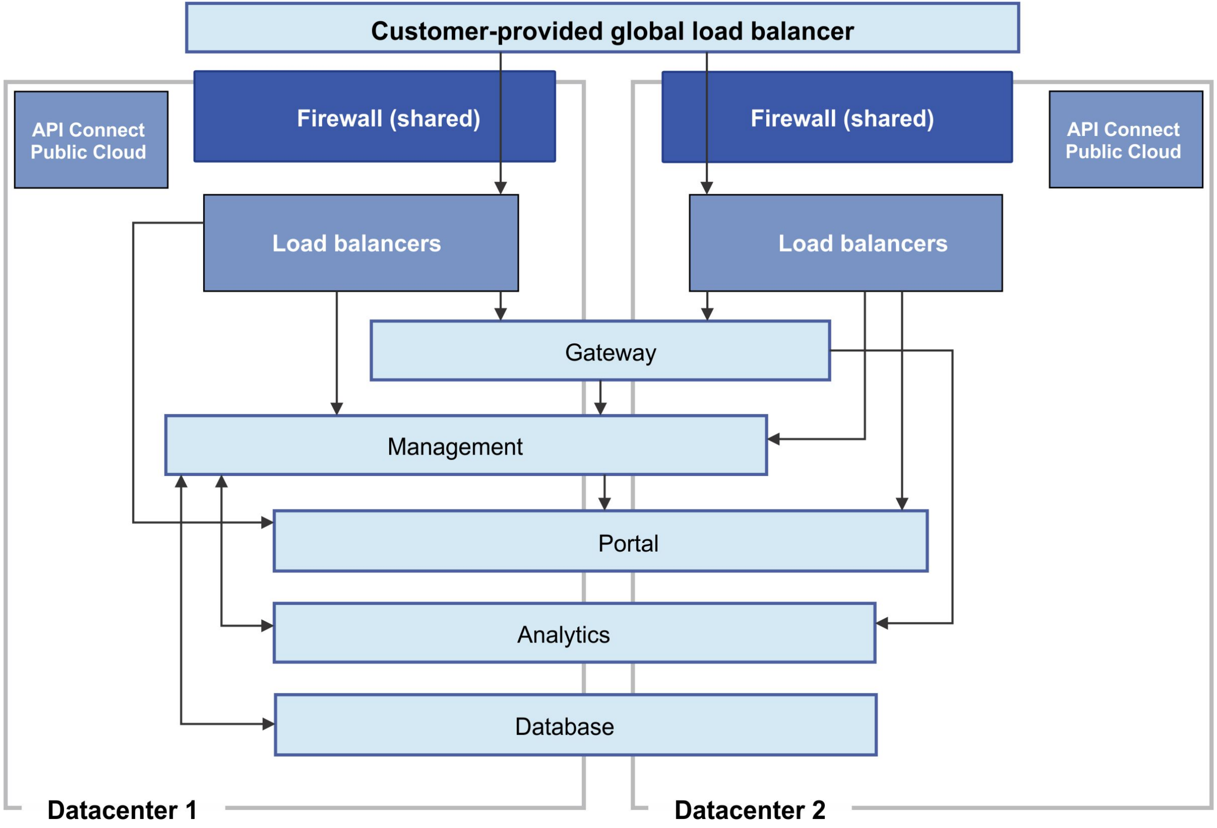
The single data center is best for smaller instances. The data is still backed up periodically, but it does not have full redundancy from different data centers.

Figure 1. V5 Reserved instance architecture with a single data center



The dual-data center configuration provides additional redundancy, and is available as an option.

Figure 2. V5 Reserved instance architecture with two data centers



Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect offerings

There are three subscription tiers of API Connect offerings, providing increasing capabilities.

Essentials

API Connect Essentials is designed for developers, and is available at no charge. This offering has less functionality than the two other offerings, and does not include IBM support. Essentials uses the Micro Gateway.

Professional

API Connect Professional builds on the Essentials offering, is suited to department-level environments, and includes IBM support. It provides greater functionality than API Connect Essentials, but is limited in comparison to the Enterprise offering.

Enterprise

API Connect Enterprise adds further capability to the Professional offering, and is aimed at enterprise-level environments. API Connect Enterprise provides advanced analytics, and includes the DataPower® Gateway.

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

Table 1. Full breakdown of each offering in relation to the API Lifecycle

	Essential (Free)	**Professional**	**Enterprise**
Create	• Auto generation of API models • Visual API creation • Single click build/package/deploy • Data source connectors: ◦ REST, SOAP, MySQL, PostgreSQL, MongoDB, Redis, Couchbase, Cloudant, DB2, Neo4j, Kafka, Memory, Mail • Advanced connectors for development and test • OpenAPI (Swagger 2.0) support, ACL	• All of the functionality that Essentials possesses • Advanced Connectors for development, test, or production use: ◦ SOAP, DB2, Oracle, MS SQL, SAP HANA	• All of the functionality that the Professional offering possesses
Run	• Node and Java compute instances	• All of the functionality that Essentials possesses • Basic clustering of Node and Java in a single data center • Unified Management via single console	• All of the functionality that the Professional offering possesses • Clustering in multiple data centers • Java: ◦ Advance clustering (auto-scaling), Log analytics, health management

	Essential (Free)	Professional	Enterprise
Manage	• API Manager (only one instance) • Developer Portal (only one instance) • REST and SOAP API support • Policy Assembly UX • Version and lifecycle management • API Discovery • API analytics (limited)	• All of the functionality that Essentials possesses • V5.0.0 ONLY V5.0.1 ONLY API Manager (up to two instances in a single data center) • V5.0.0 ONLY V5.0.1 ONLY Developer Portal clustering (up to three instances in a single data center) • V5.0.2 + API Manager (limited to installation in a single data center) • V5.0.2 + Developer Portal clustering (limited to installation in a single data center)	• All of the functionality that the Professional offering possesses • In excess of three instances in a single data center • Multiple data center deployments • API Analytics (full) • V5.0.8 + Create and maintain subscription plans with billing that customers can provide a credit card number and use your APIs.

Secure	• V5.0.0 ONLY V5.0.1 ONLY Programmable Micro Gateway (only one instance)
	• V5.0.2 + Programmable Micro Gateway (only one instance per Catalog)
	• HTTP 1.1, HTTPS 1.1
	• REST API Proxy support
	• SOAP API Proxy support
	• Built-in policies supported by Micro Gateway:
	  ◦ Client ID/Secret
	  ◦ Basic Auth
	  ◦ Basic rate limiting
	  ◦ CORS
	  ◦ Invoke (call service over HTTP)
	  ◦ Set Variable
	  ◦ JavaScript invoke
		• All of the functionality that Essentials possesses
		• V5.0.0 ONLY V5.0.1 ONLY Micro Gateway (up to two instances in a single data center)
		• V5.0.2 + Micro Gateway (limited to installation in a single data center)
			• All of the functionality that the Professional offering possesses
			• DataPower Gateway Virtual:
			  ◦ In excess of three instances in a single data center
			  ◦ Multiple data center deployment
			  ◦ API Management V4 parity
			• Additional built-in policies supported by DataPower Gateway:
			  ◦ OAuth, Advanced rate limiting, Redaction, Map, Activity Log, REST validation, GatewayScript invoke, XSLT invoke, SOAP/XML schema validation, JSON to/from XML transform, Response caching
			• Utilize existing DataPower Gateway functionality as user-defined policies

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect V5 for IBM Cloud concepts

To help you get started, read about the API Connect V5 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.

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
- 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 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 in the API Designer.

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:

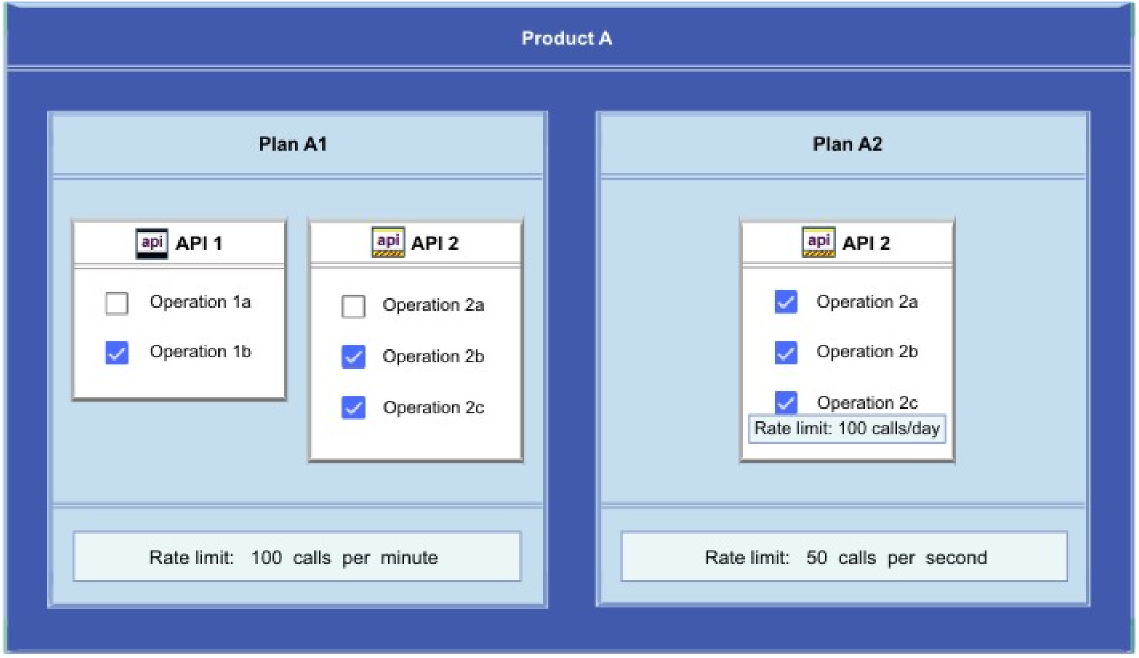
- Control which APIs an application developer can use
- Make available a collection of operations from one or more APIs
- Apply rate limits to APIs to differentiate between offerings
- Implement different rate limits to specify how many requests a consuming application is allowed to make during a specified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

A rate limit can be implemented as a default rate that is shared across all operations in a Plan, or can be set for specific operations of an API. 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.
The following diagram illustrates how Plans can be used to make operations from one or more APIs available, and to set rate limits on both Plans and individual operations:

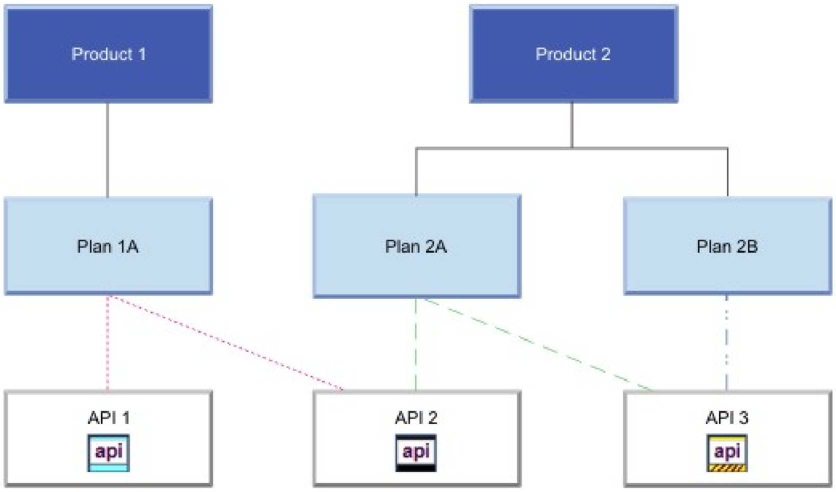


The diagram illustrates the following concepts:

- Product A contains two APIs and two Plans.
- Both APIs are included in Plan A1.
- Plan A2 includes only API 2, which is also included in Plan A1. Operation 2a for API 2 is excluded from Plan A1. All operations in API 2 are included in Plan A2.
- Different rate limits are set for the two Plans at the Plan level. For API 2 in Plan A2, a rate limit is also set specifically for Operation 2c to override the rate limit set at the Plan level.

Products are used to manage the lifecycle of the APIs they contain. The states in the Product lifecycle include draft, staged, published, deprecated, retired, and archived. A Product in draft state is moved to stage state when saved to a *Catalog* . A product is moved to published state when its Catalog is published. The APIs in a Product become accessible to API consumers when the Product is in published state and made visible on an API Connect *Developer Portal*. After a Product is published, application developers can 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 Plans and 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 also 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.

V5.0.5 + 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 developer. 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.

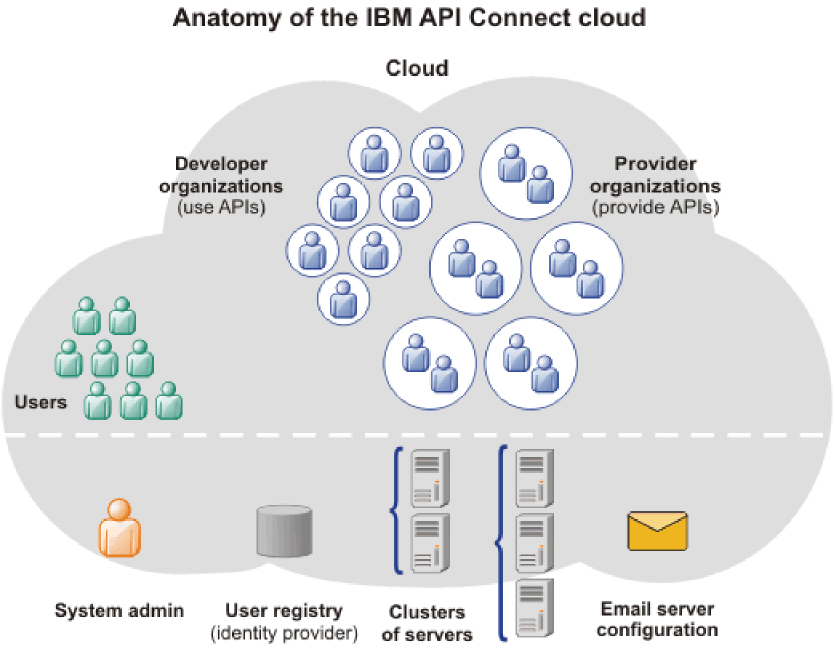
A developer organization owns only *developer* applications, and consumes the APIs and applications produced by the provider organization. For this reason, developer organizations are also called *consumer* organizations because the developers consume the APIs that are published on the Developer Portal. Standard responsibilities within a developer organization include:

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

The provider and developer 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 developer 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, developer organizations, and users. The clusters shown are logical groupings of servers with the same capability.


Anatomy of the IBM API Connect cloud
Cloud
Developer organizations (use APIs)
Provider organizations (provide APIs)
Users
System admin
User registry (identity provider)
Clusters of servers
Email server configuration

For more information about organizations, see Administering provider organizations and Administering Developer organizations.

Applications

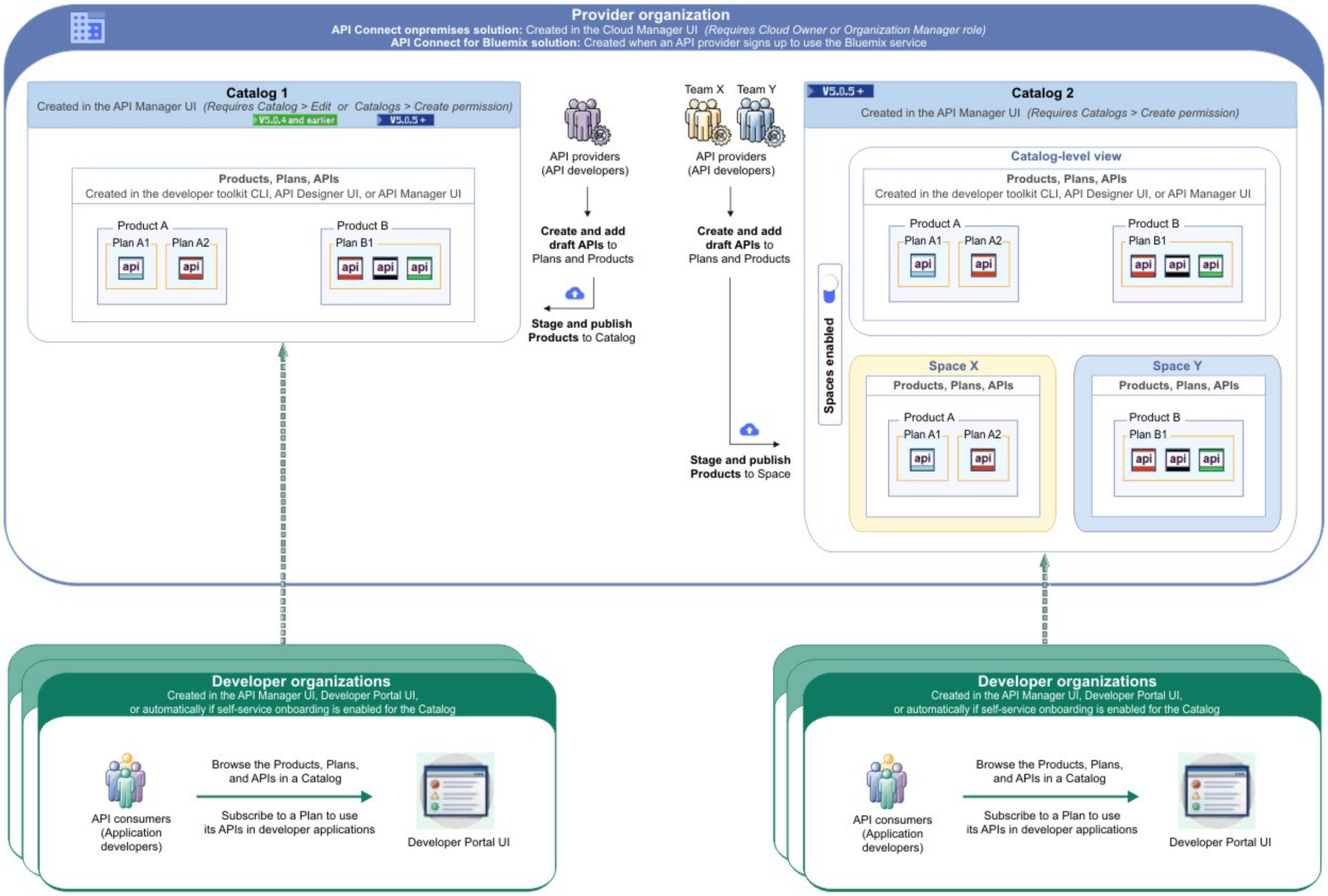
In addition to APIs, provider organizations can also create applications (with associated APIs), which are built using Node.js and Java technology. When published, these APIs and applications are called by developer applications. The developer applications contain client code that accesses APIs to interact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

For information about creating provider applications, see Developing your 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 developer 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 developer organizations are granted permission to explore, discover, and use the published Products and APIs. In each of the developer 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 developer 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.


Provider organization
API Connect onpremises solution: Created in the Cloud Manager UI (Requires Cloud Owner or Organization Manager role)
API Connect for Bluemix solution: Created when an API provider signs up to use the Bluemix service
Catalog 1
Created in the API Manager UI (Requires Catalog > Edit or Catalogs > Create permission)
V5.0.4 and earlier
V5.0.5 +
Products, Plans, APIs
Created in the developer toolkit CLI, API Designer UI, or API Manager UI
Product A
Plan A1
Plan A2
api
api
Product B
Plan B1
api
api
api
API providers
(API developers)
Create and add draft APIs to Plans and Products
Stage and publish Products to Catalog
Team X
Team Y
API providers
(API developers)
Create and add draft APIs to Plans and Products
Stage and publish Products to Space
V5.0.5 +
Catalog 2
Created in the API Manager UI (Requires Catalogs > Create permission)
Catalog-level view
Products, Plans, APIs
Created in the developer toolkit CLI, API Designer UI, or API Manager UI
Product A
Plan A1
Plan A2
api
api
Product B
Plan B1
api
api
api
Spaces enabled
Space X
Products, Plans, APIs
Product A
Plan A1
Plan A2
api
api
Space Y
Products, Plans, APIs
Product B
Plan B1
api
api
api
Developer organizations
Created in the API Manager UI, Developer Portal UI,
or automatically if self-service onboarding is enabled for the Catalog
API consumers
(Application developers)
Browse the Products, Plans, and APIs in a Catalog
Subscribe to a Plan to use its APIs in developer applications
Developer Portal UI
Developer organizations
Created in the API Manager UI, Developer Portal UI,
or automatically if self-service onboarding is enabled for the Catalog
API consumers
(Application developers)
Browse the Products, Plans, and APIs in a Catalog
Subscribe to a Plan to use its APIs in developer applications
Developer Portal UI

Related information

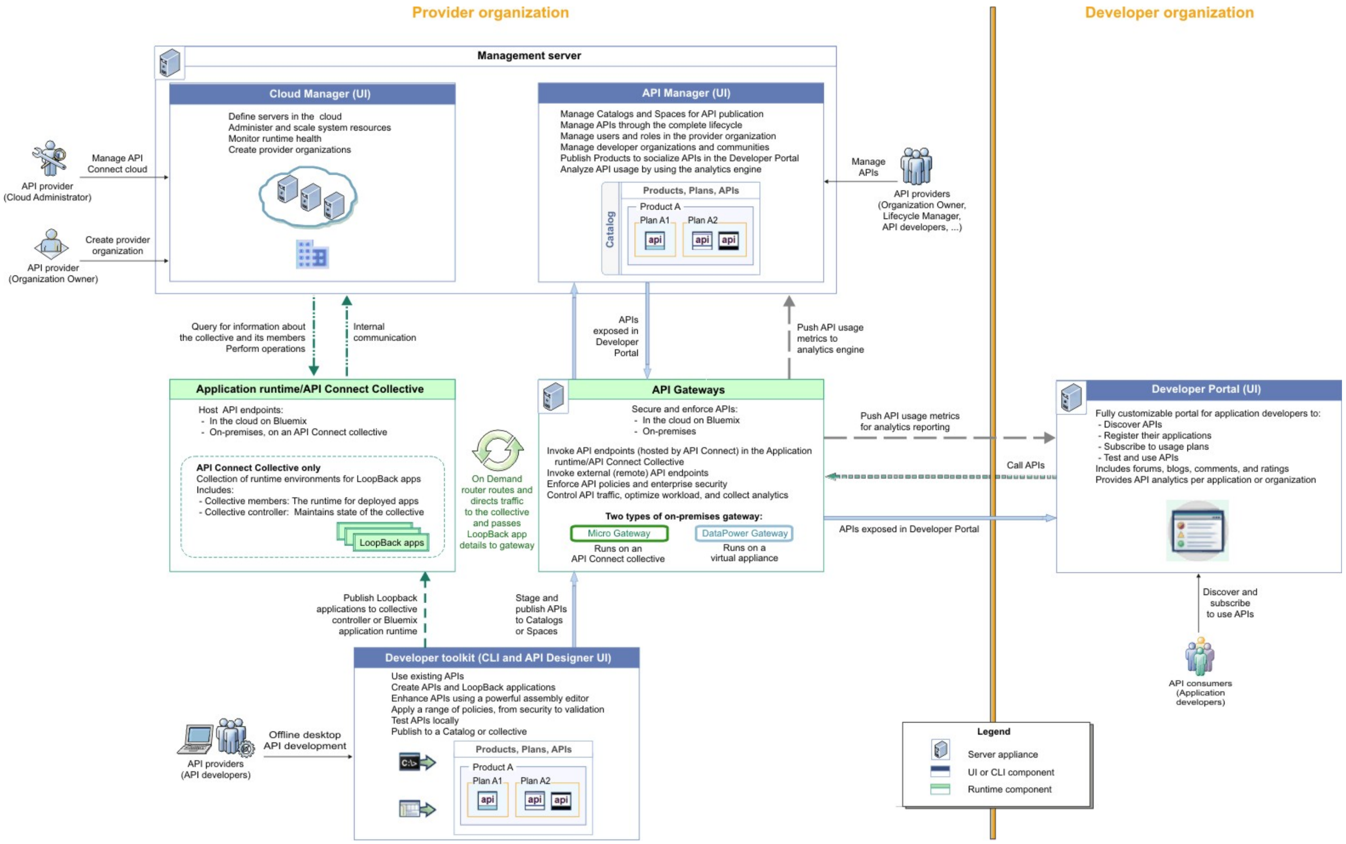
- Signing up to the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

The following diagram depicts the API Connect components, summarizes the key functions of each component, and illustrates how the components interact. The components are further described in the sections that follow.


Provider organization
Developer organization
Management server
Cloud Manager (UI)
Define servers in the cloud
Administer and scale system resources
Monitor runtime health
Create provider organizations
Manage API Connect cloud
API provider (Cloud Administrator)
Create provider organization
API provider (Organization Owner)
API Manager (UI)
Manage Catalogs and Spaces for API publication
Manage APIs through the complete lifecycle
Manage users and roles in the provider organization
Manage developer organizations and communities
Publish Products to socialize APIs in the Developer Portal
Analyze API usage by using the analytics engine
Catalog
Products, Plans, APIs
Product A
Plan A1
Plan A2
api
api
api
Manage APIs
API providers (Organization Owner, Lifecycle Manager, API developers, ...)
Query for information about the collective and its members
Perform operations
Internal communication
APIs exposed in Developer Portal
Push API usage metrics to analytics engine
Application runtime/API Connect Collective
Host API endpoints:
- In the cloud on Bluemix
- On-premises, on an API Connect collective
API Connect Collective only
Collection of runtime environments for LoopBack apps
Includes:
- Collective members: The runtime for deployed apps
- Collective controller: Maintains state of the collective
LoopBack apps
On Demand router routes and directs traffic to the collective and passes LoopBack app details to gateway
API Gateways
Secure and enforce APIs:
- In the cloud on Bluemix
- On-premises
Invoke API endpoints (hosted by API Connect) in the Application runtime/API Connect Collective
Invoke external (remote) API endpoints
Enforce API policies and enterprise security
Control API traffic, optimize workload, and collect analytics
Two types of on-premises gateway:
Micro Gateway
Runs on an API Connect collective
DataPower Gateway
Runs on a virtual appliance
Push API usage metrics for analytics reporting
Call APIs
APIs exposed in Developer Portal
Developer Portal (UI)
Fully customizable portal for application developers to:
- Discover APIs
- Register their applications
- Subscribe to usage plans
- Test and use APIs
Includes forums, blogs, comments, and ratings
Provides API analytics per application or organization
Discover and subscribe to use APIs
API consumers (Application developers)
Publish Loopback applications to collective controller or Bluemix application runtime
Stage and publish APIs to Catalogs or Spaces
Developer toolkit (CLI and API Designer UI)
Use existing APIs
Create APIs and LoopBack applications
Enhance APIs using a powerful assembly editor
Apply a range of policies, from security to validation
Test APIs locally
Publish to a Catalog or collective
Products, Plans, APIs
Product A
Plan A1
Plan A2
api
api
api
C:\>
Offline desktop API development
API providers (API developers)
Legend
Server appliance
UI or CLI component
Runtime component

- Cloud Manager
- The developer toolkit
- API Manager
- API Gateways
- Application runtime/Containerized runtime/API Connect Collective
- Developer Portal
- 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 Application runtime/Containerized runtime/API Connect Collective.
- 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).

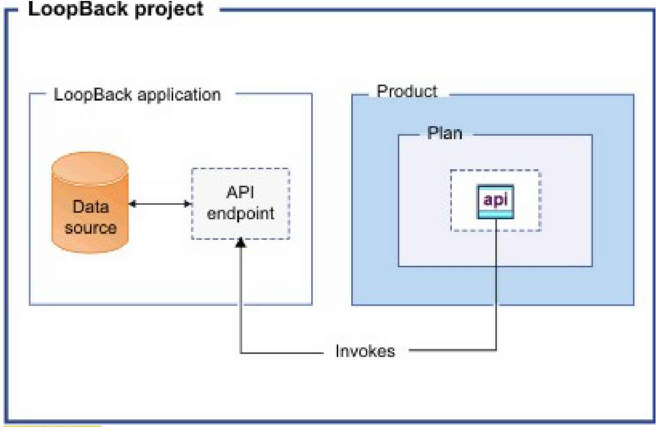
For more information about the Cloud Manager, see Managing your cloud.

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) and a corresponding graphical user interface, the API Designer. It incorporates LoopBack, an open source Node.js framework.

API developers use the API management functions in the API Designer 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 Designer 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. The following diagram illustrates the LoopBack project architecture:



V5.0.7+ Draft APIs (in their containing Products) that are created using the API Designer, CLI, or LoopBack are published to Catalogs. Applications created using LoopBack are published to containers or to an API Connect collective, from where they run when called. (For information about containers and collectives, see Application runtime/Containerized runtime/API Connect Collective.)

V5.0.6 and earlier Draft APIs (in their containing Products) that are created using the API Designer, CLI, or LoopBack are published to Catalogs. Applications created using LoopBack are published to an API Connect collective, from where they run when called. (API Connect Collective is a separate, installable component of API Connect and is described in Application runtime/Containerized runtime/API Connect Collective.)

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 developer 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 developer 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 developer 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.

Note: The API Manager includes the ability to create and edit APIs. However, the preferred practice is for these tasks to be performed by using the toolkit CLI or API Designer UI, which are provided in the developer toolkit component.
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. Two types of Gateway are available for use in API Connect:

- Micro Gateway only The Micro Gateway is a Gateway that is built on Node.js, for use by developers and single departmental projects, and it provides enforcement for the authentication, authorization, and flow requirements of an API. The Micro Gateway provides a limited set of API policies for security and traffic management. The Micro Gateway is deployed on an API Connect collective and supports a single Catalog per instance or cluster.

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

- DataPower Gateway only 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 has more policies available to it than the Micro Gateway and can handle enterprise level complex integration. DataPower Gateway supports containers for flexible runtime management.

Your API Connect offering (or *edition*) can include a Micro Gateway only, or both a Micro Gateway and virtual DataPower Gateway. Support for a physical DataPower Gateway is also available, subject to certain conditions. For more information about the API Connect offerings, see API Connect offerings.

Application runtime/V5.0.7+Containerized runtime/Deprecated in V5.0.7API Connect Collective

You can run applications and API implementations in API Connect Collectives (Deprecated in V5.0.7) or application containers.
V5.0.7+Note: API Connect Collective is deprecated in API Connect V5.0.7.

Application runtime

The *application runtime* provides a runtime environment for executing APIs in API Connect.

V5.0.7+Containerized runtime

V5.0.7+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. For more information, see Installing a containerized runtime environment.

Deprecated in V5.0.7API Connect Collective

In an on-premises environment, the API Connect Collective component additionally provides a collection of runtime environments for executing LoopBack applications that are created using the developer toolkit.

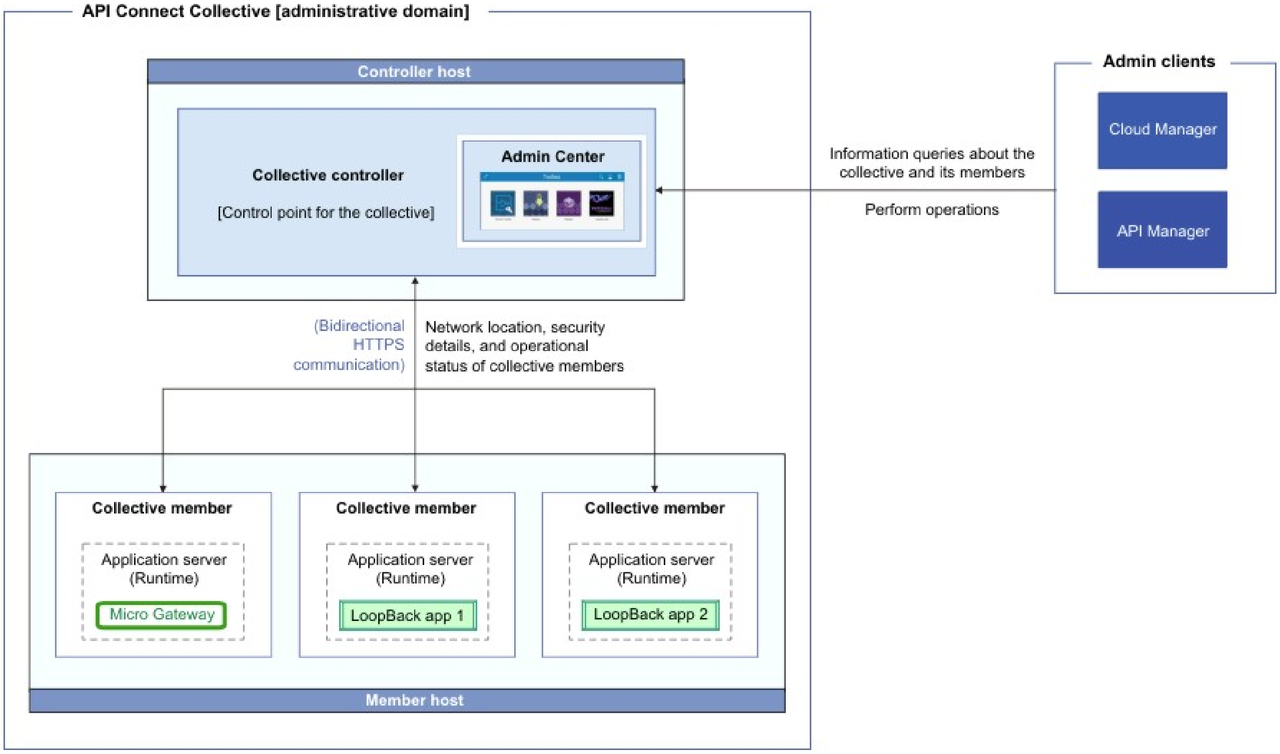
In API Connect, a *collective* is used to deploy and run LoopBack applications and the Micro Gateway (which itself is created as a LoopBack app). A collective is an administrative and operational domain for a collection of servers. In this context, a *server* refers to a LoopBack application, which is packaged as an application server for deployment to the collective. When a LoopBack application (server) is published to a collective, it must be *joined* to the collective, which then makes that application server a *collective member*. A collective member is the runtime for a deployed application. The collective members for each deployed application run on a *member host*, which is a machine that is set up to run servers in a collective, has an SSH daemon installed, and is registered with the collective.

The collective members are managed by a *collective controller*, which is a (WebSphere Application Server) Liberty Network Deployment Java server that maintains the state of the collective. The collective controller runs on a *controller host* machine. The collective controller acts as a centralized control point for the collective to perform operations such as file transfer and cluster management, and includes storage and collaboration capabilities. More specifically, the collective controller is responsible for deploying a published LoopBack application *as a server* on the member host (using SSH), joining the server to the collective (to make it a collective member), and then starting the server so that the application can run.

The collective controller and collective members use HTTPS for bidirectional communication. The collective members share information about their network location, security details, and operational status, which ensures that information can be readily retrieved without having to invoke an operation on each individual member. The collective controller also periodically monitors the health of the servers to see if they need to be restarted. The controller publishes information about the member applications to the Micro Gateway and to the DataPower Gateway (using an On Demand router), so they know what applications are available to route to. The collective also provides a *Liberty Admin Center* UI console on the collective controller, which can be used to monitor the collective. The Admin Center is accessible from the API Manager UI with read-only access. API Connect clients that run software processes can also connect to the collective controller to query for information about the collective and its membership, or to perform operations.

A collective can be configured with one or more collective controllers, depending on scalability and high availability requirements, and can have multiple member hosts. The collective controller and collective members can also either be on separate hosts or on the same host. For more information about collectives, see Liberty: Collective architecture in the WebSphere Application Server Liberty Network Deployment documentation.

The following diagram depicts the architecture for an API Connect collective to which the Cloud Manager and API Manager connect. The collective controller is on a separate host from the three collective members.


API Connect Collective [administrative domain]
Controller host
Collective controller
[Control point for the collective]
Admin Center
Admin clients
Cloud Manager
API Manager
Information queries about the collective and its members
Perform operations
(Bidirectional HTTPS communication)
Network location, security details, and operational status of collective members
Collective member
Application server (Runtime)
Micro Gateway
Collective member
Application server (Runtime)
LoopBack app 1
Collective member
Application server (Runtime)
LoopBack app 2
Member host

To enable communication between a collective and the other API Connect components, the collective and collective controller must be registered within the Cloud Manager.

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 developer 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 developer organization. For more information, see Developer Portal: discover and use APIs.

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 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 Management server also provides analytic functions that collect and store information about APIs and API users. The Cloud Manager and API Manager user interfaces run on the Management server.
- 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. The Gateway types include a Micro Gateway and a DataPower Gateway.
- **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.

Table 1. Minimum firmware requirements to run in an API Connect cloud

Server Type	Min. Required Firmware Version	Supported Hardware
Management server	IBM API Connect Version 5.0	Not applicable
Gateway server	IBM DataPower Gateway Version details are provided in the Supported Software section of the IBM API Connect Version 5.0 Detailed System Requirements report on the IBM Software Product Compatibility Reports site for each API Connect offering: (the following links display the Detailed System Requirements report for the latest version. If you want to see the report for an earlier version, open the Detailed system requirements for a specific product page, search for the IBM API Connect product, then select the required offering and version) • Essentials • Professional • Enterprise Important: The Application Optimization (AO) option is required. Ensure that the AO module is activated.	See the Supported Software section of the IBM API Connect Version 5.0 Detailed System Requirements report for each API Connect offering: (the following links display the Detailed System Requirements report for the latest version. If you want to see the report for an earlier version, open the Detailed system requirements for a specific product page, search for the IBM API Connect product, then select the required offering and version) • Essentials • Professional • Enterprise
Developer Portal	IBM API Connect Developer Portal Version 5.0	Not applicable

For more information about planning your cloud, see Planning your cloud.

Typical tasks per interface component

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

Table 2. API Connect Tasks per interface component

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

-
- If self-service onboarding is enabled for a Catalog, a developer 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 developer organization.
  For more information about the API Connect components, see API Connect components.

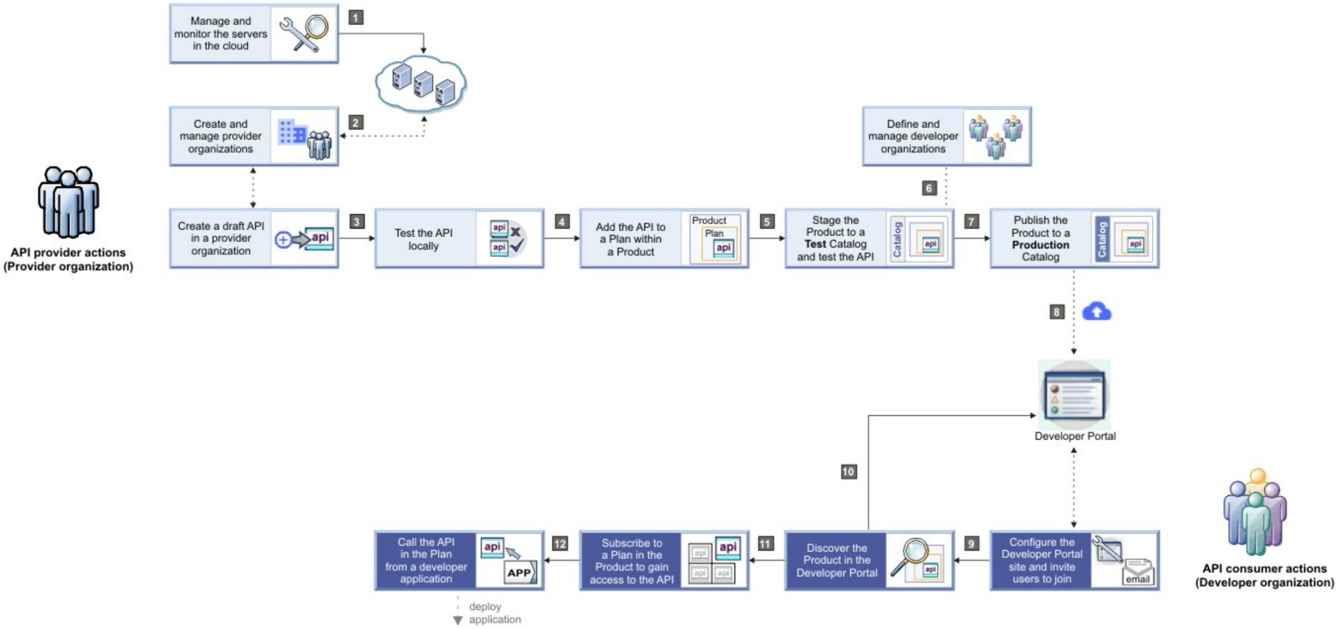
Related concepts

- Packaging strategy and terminology in API Connect

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 developer organizations.



Manage and monitor the servers in the cloud
1
Create and manage provider organizations
2
API provider actions (Provider organization)
Create a draft API in a provider organization
api
3
Test the API locally
api
api
4
Add the API to a Plan within a Product
Product
Plan
api
5
Stage the Product to a Test Catalog and test the API
Catalog
api
Define and manage developer organizations
6
7
Publish the Product to a Production Catalog
Catalog
api
8
Developer Portal
10
API consumer actions (Developer organization)
Call the API in the Plan from a developer application
api
APP
12
Subscribe to a Plan in the Product to gain access to the API
api
11
Discover the Product in the Developer Portal
api
9
Configure the Developer Portal site and invite users to join
email
deploy application

Action: 1

Cloud Owner | Cloud Administrator The minimum requirements for an on-premises API Connect solution consist of one Management server to manage and analyze APIs, one Gateway server to direct API traffic, and a server to host the Developer Portal. As a Cloud Owner or Cloud Administrator, you gather a collection of Management, 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 developer organization is assigned an owner, and might represent a business partner, or a group of internal or external developers. Developer 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 developer 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 developer 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 lifecycles 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

Developer Organization Owner After a developer organization is created, its designated Developer Organization Owner can invite other users to join the developer organization so that they can access the Developer Portal and use the APIs that have been made available to the developer organization. The Developer 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 usage metrics and analytics to the API Manager.

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

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials

You can find here tutorials for using IBM® API Connect. The tutorials are divided between the developer toolkit, API Manager, and the Developer Portal.

Introduction

The developer toolkit tutorials include tutorials for using LoopBack® functions to create and configure APIs for viewing coffee shops and tutorials for using the API Designer user interface to create and configure API definitions for BankA and securely expose their existing APIs.

The API Manager tutorials detail the creation of SOAP API definitions for BankA.

In the Developer Portal tutorials, you will use and customize a Developer Portal for BankA.

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.

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.

To access the tutorials, see the following sections:

- Developer toolkit tutorials
- API Manager tutorials
- Developer Portal tutorials

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 IBM 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; for more information, see Working with roles in the Developer Portal.

The following sections describe the roles and permissions for each of the IBM 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.

Table 1. Cloud Manager UI permissions

Permission	Action	Meaning
Analytics	View	View the cloud analytics data
Services	View	View management and gateway services and servers
	Edit	Add, edit, and delete management and gateway services and servers
Organizations	View	View provider organizations
	Edit	Add, update, and delete provider organizations and their owners
Users	View	View Cloud Manager users
	Edit	Add, update, and delete Cloud Manager users
TLS Profiles	View	View SSL identities
	Edit	Add, update, and delete SSL identities
User Registries	View	View user registries
	Edit	Add. update, and delete user registries
Settings	View	View the cloud settings
	Edit	Edit the cloud settings

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
Cloud Owner	All permissions	All actions
Cloud Administrator	Analytics	View
	Services	View, Manage
	Users	View, Manage
	TLS Profiles	View, Manage
	User Registries	View, Manage

Role	Permissions	Actions
	Settings	View, Manage
Organization Manager	Organizations	View, Manage
Topology Administrator	Analytics	View
	Services	View, Manage
	TLS Profiles	View, Manage
	User Registries	View, Manage
	Settings	View, Manage

Note: An additional role, System, provides all permissions for the Cloud Manager user interface and, in addition, provides REST access to all APIs but not to the API Manager or Developer Portal user interface themselves.

User roles and permissions in the API Manager UI

The following tables describe the API Manager UI user permissions.

V5.0.4 and earlier

Table 3. API Manager UI permissions

Permission	Action	Meaning
Roles	View	View the roles editing page
	Edit	Create, edit, and delete roles in the roles editing page
Users	View	View organization users
	Edit	Add, update, and delete organization users
TLS Profiles	View	View SSL Identities
	Edit	Create, edit, and delete SSL Identities
User Registries	View	View user registries
	Edit	Create, edit, and delete user registries
Draft APIs	View	View draft APIs
	Edit	Create, update, and delete draft APIs
Draft Products	View	View Products
	Edit	Create, update, and delete draft Products
Subscriptions	View	View Plan subscriptions
	Approve	Approve Plan subscriptions
Catalog Administration	View	View Catalogs
	Edit	Create, edit, and delete Catalogs
Developers	View	View developers and developer organizations
	Manage	Add, update, and delete developers and developer organizations
Analytics	View	View Catalog analytics

A user with Roles permission can change the permission assignments, and can create custom roles; for more information, see Creating custom roles.

V5.0.4 and earlier

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

Role	Permissions	Actions
Owner	All permissions	All actions
Administrator	All permissions	All actions
Product Manager	Users	View
	TLS Profiles	View
	User Registries	View
	Draft APIs	View, Edit
	Draft Products	View, Edit
	Subscriptions	View, Approve
	Catalog Administration	View
	Developers	View, Manage
	Analytics	View
API Developer	Users	View
	TLS Profiles	View
	User Registries	View
	Draft APIs	View, Edit

Role	Permissions	Actions
	Draft Products	View, Edit
	Subscriptions	View
	Analytics	View
Publisher	Users	View
	TLS Profiles	View
	User Registries	View
	Draft APIs	View
	Draft Products	View
	Subscriptions	View, Approve
	Catalog Administration	View, Edit
	Developers	View
	Analytics	View

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.

V5.0.5+

Table 5. Organization permissions

Permissions	Action	Permits the member to
Draft APIs	View	View draft APIs
	Edit	Edit draft APIs
Organization Settings	View	View organization's configuration settings, including roles, TLS profiles, and user registries
	Manage	Manage organization's configuration settings, including roles, TLS profiles, and user registries
Catalogs	Create	Create Catalogs in the organization; the creator of a Catalog owns that Catalog and has full administration permissions, including deletion of the Catalog
	View	View all Catalogs in the organization
	Manage	Manage all Catalogs in the organization; this includes permission to delete any Catalog
Draft Products	View	View draft Products
	Edit	Edit draft Products
Organization Members	View	View organization's members
	Manage	Manage organization's members

A user with Organization Settings > Manage permission can change the permission assignments, and can create custom roles; for more information, see Creating custom roles.

V5.0.5+

Table 6. Catalog permissions

Permissions	Action	Permits the member to
Catalog Members	View	View Catalog members
	Manage	Manage Catalog members
Catalogs Settings	View	View the Catalog's configuration settings, including policies and OpenAPI (Swagger 2.0) extensions
	Manage	Manage the Catalog's configuration settings, including policies and OpenAPI (Swagger 2.0) extensions
Subscriptions	View	View subscriptions
	Manage	Manage subscriptions
API Products	Stage	Stage Products in a Catalog
	View	View Products in a Catalog
	Manage	Manage Products in a Catalog
Subscription Approvals	View	View subscription approvals
	Manage	Manage subscription approvals
V5.0.7+ Subscription and Application Approvals	V5.0.7+ View	V5.0.7+ View subscription and application upgrade approvals
V5.0.7+	V5.0.7+ Manage	V5.0.7+ Manage subscription and application upgrade approvals
Analytics	View	View analytics
	Manage	Manage analytics
Applications	View	View applications

Permissions	Action	Permits the member to
	Manage	Manage applications
Developer Organizations and Developers	View	View developer organizations and developers
	Manage	Manage developer organizations and developers
Product Lifecycle Approvals	View	View Product lifecycle changes
	Stage	Stage Products
	Publish	Publish Products
	Deprecate	Deprecate Products
	Retire	Retire Products
	Replace	Replace Products
	Supersede	Supersede Products
Spaces	Create	Create Spaces
	View	View Spaces
	Manage	Manage Spaces

V5.0.5+

Table 7. Space permissions

Permissions	Action	Permits the member to
Space Members	View	View Space members
	Manage	Manage Spaces members
Spaces Settings	View	View the Space configuration settings
	Manage	Manage the Space configuration settings
Subscriptions	View	View subscriptions
	Manage	Manage subscriptions
API Products	Stage	Stage Products in a Space
	View	View Products in a Space
	Manage	Manage Products in a Space
Subscription Approvals	View	View subscription approvals
	Manage	Manage subscription approvals
V5.0.7+ Subscription and Application Approvals	V5.0.7+ View	V5.0.7+ View subscription and application upgrade approvals
V5.0.7+	V5.0.7+ Manage	V5.0.7+ Manage subscription and application upgrade approvals
Analytics	View	View analytics
	Manage	Manage analytics
Applications	View	View applications
	Manage	Manage applications
Developer Organizations and Developers	View	View developer organizations and developers
	Manage	Manage developer organizations and developers
Product Lifecycle Approvals	View	View Product lifecycle changes
	Stage	Stage Products
	Publish	Publish Products
	Deprecate	Deprecate Products
	Retire	Retire Products
	Replace	Replace Products
	Supersede	Supersede Products

V5.0.5+

Table 8. Default API Manager UI roles and the default permissions assigned to those roles.

Role	Component	Permissions	Actions
Organization Owner	All	All permissions	All actions
Catalog Owner	All	All permissions	All actions
Space Owner	All	All permissions	All actions
Administrator	All	All permissions	All actions
Product Manager	Organization	Draft APIs	View, Edit
		Organization Settings	View
		Catalogs	View
		Draft Products	View, Edit
		Organization Members	View

Role	Component	Permissions	Actions
	Catalog	Catalog Members	View
		Catalog Settings	View
		Subscriptions	View, Manage
		API Products	View
		Subscription Approvals	View, Manage
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View, Manage
		Analytics	View, Manage
		Applications	View, Manage
		Developer Organizations and Developers	View, Manage
		Product Lifecycle Approvals	View
		Spaces	None
	Space	Space Members	View
		Spaces Settings	View
		Subscriptions	View, Manage
		API Products	View
		Subscription Approvals	View, Manage
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View, Manage
		Analytics	View, Manage
		Applications	View, Manage
		Developer Organizations and Developers	View, Manage
		Product Lifecycle Approvals	View
API Developer	Organization	Draft APIs	View, Edit
		Organization Settings	View
		Catalogs	Create, View
		Draft Products	View, Edit
		Organization Members	View
	Catalog	Catalog Members	View
		Catalog Settings	View
		Subscriptions	View
		API Products	Stage, View, Manage
		Subscription Approvals	View
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View
		Analytics	View
		Applications	View
		Developer Organizations and Developers	View
		Product Lifecycle Approvals	View
		Spaces	None
	Space	Space Members	View
		Spaces Settings	View
		Subscriptions	View
		API Products	Stage, View, Manage
		Subscription Approvals	View
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View
		Analytics	View
		Applications	View
		Developer Organizations and Developers	View
		Product Lifecycle Approvals	View
API Administrator	Organization	Draft APIs	View
		Organization Settings	View
		Catalogs	Create, View
		Draft Products	View

Role	Component	Permissions	Actions
		Organization Members	View
	Catalog	Catalog Members	View, Manage
		Catalog Settings	View, Manage
		Subscriptions	View, Manage
		API Products	Stage, View, Manage
		Subscription Approvals	View, Manage
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View, Manage
		Analytics	View, Manage
		Applications	View, Manage
		Developer Organizations and Developers	View
		Product Lifecycle Approvals	View, Stage, Publish, Deprecate, Retire, Replace, Supersede
		Spaces	None
	Space	Space Members	View
		Spaces Settings	View
		Subscriptions	View
		API Products	Stage, View, Manage
		Subscription Approvals	View
V5.0.7+	V5.0.7+	V5.0.7+ Subscription and Application Approvals	V5.0.7+ View
		Analytics	View
		Applications	View
		Developer Organizations and Developers	View
		Product Lifecycle Approvals	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; for more information, see Working with roles in the Developer Portal.

Table 9. Developer Portal UI roles

Role	Tasks that can be performed
Developer Organization Owner	• Invite other users to join the developer organization • Change the developer organization name • View and create applications • View Products and APIs • View subscriptions and subscribe to use APIs • Use the Developer Portal test tool • V5.0.8+ Enter your credit card transaction processing information to receive payments for subscription plans.
App Developer	• View and create applications • View Products and APIs • View subscriptions and subscribe to use APIs • Use the Developer Portal test tool
Viewer	• View applications • View Products and APIs • View subscriptions • Use the Developer Portal test tool

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 developer organizations
- Using the Portal REST APIs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reverting servers to stand-alone appliances

Describes how servers in the IBM® API Connect cloud can be reimaged to function as a stand-alone appliance.

About this task

The virtual and physical appliances that are defined as the servers for an IBM API Connect cloud can be reimaged to function as a stand-alone appliance. When you revert a server to a stand-alone appliance, all data and logs previously on the server are wiped clean.

Procedure

1. Remove the server from the API Connect cloud.
2. Log in to the corresponding appliance command line interface through a secure shell (SSH).
3. Reconfigure the appliance:
   a. For DataPower® appliances: Enter the command `reinit` ***`filename`*** to reconfigure the appliance, where ***`filename`*** represents a firmware image within the image directory. This command deletes all existing configuration data. All network configuration data is also deleted and the admin user password is reset to the default setting, "admin".
   b. For non DataPower appliances, enter one of the following commands:
      - **system clean apiconfig**. This command removes only IBM API Connect configuration information, and is the fasted way to revert an existing server to a stand-alone appliance.
      - **system clean all**. This command removes all data and configurations. The result is equivalent to restoring factory default settings for the appliance. The network settings are also reset to use the default DHCP setting and the appliance tries to use a DHCP server. If there is no DHCP server, the appliance is not active on the network.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Define the main site in your API Connect cloud

By defining a main site in your API Connect cloud, you can ensure that your specified server configurations are preserved if a network link between sites is interrupted.

By defining the main site, the administrator can control the servers that are involved in automatic failover in the case of a server failure. To protect against network outages or latencies that can lead to dissociation, it is recommended that only servers in a single data center or geographic location should be registered as the main site.
Note: Defining the main site in this topic is applicable to management servers, and not the Developer Portal or Gateway servers.
With the main site disabled, a separation between two data centers can lead to dissociation. A defined main site can remain fully operational while the management servers in the remote site run in a degraded mode. For more information about dissociation, see Dissociation and your cloud and Configuration database failover timeout.

While the remote site is in a degraded mode, its servers are unable to write to the configuration database, but might continue to serve read-only queries. After communication is restored, remote servers resume operations.

If the main site goes completely down, the administrator can manually promote a remote server to resume operations.

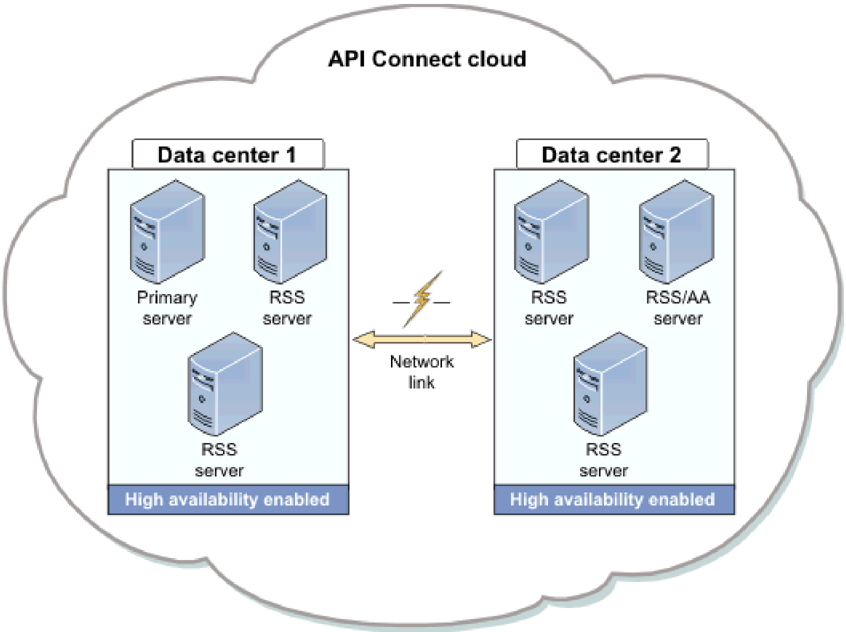
A site is a collection of servers in the same datacenter or geographical location in your API Connect cloud.

Any management server in a main site can be assigned the role of Primary server, and one of the remaining servers is automatically assigned the Active Arbitrator role depending on its current role. For more information on the role dependent priority, see Reference information for main site functionality.

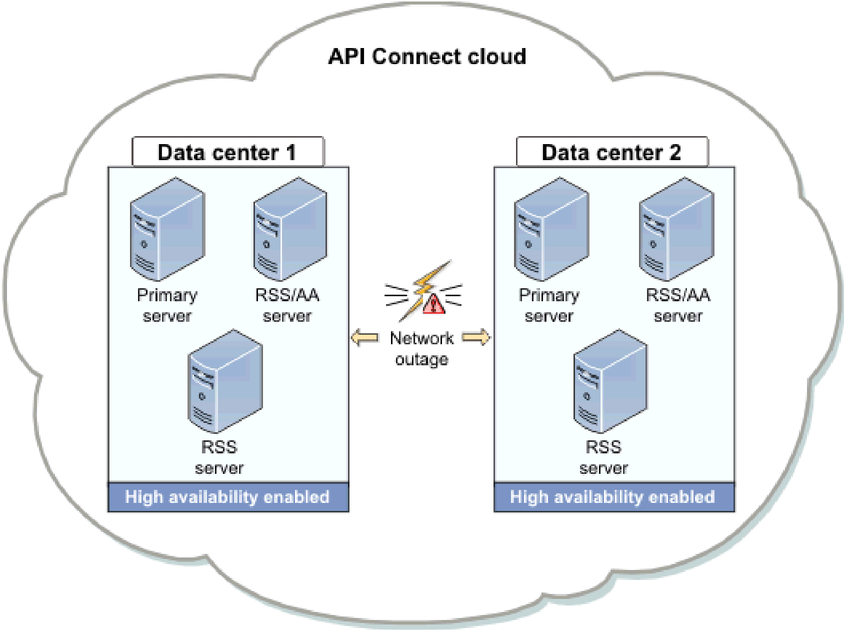
Primary
: A Primary server is responsible for any actions that involve writing.

Active Arbitrator
: An Active Arbitrator is responsible for deciding which server is promoted to become the Primary server, if the original Primary server is unable to be contacted.

The following diagram demonstrates the composition of sites in a cloud that does not have a definitive main site, where the Primary and Active Arbitrator servers are located in different sites:



If you have not defined the main site in your cloud and a network link between sites is interrupted, you might have to clean one of the sites, which involves losing all of the updates on the site. The following diagram demonstrates how a network link can cause two Primary servers to be assigned in two separates sites, which might result in one of the sites being cleaned:



However, you can define a site to be the main site, by ensuring that at least two servers in the site are assigned the Primary and Active Arbitrator roles.

- V5.0.2 + **Defining the main site**
  Dissociation is where a cloud has two or more Primary servers, which can lead to divergence in data across the cloud. To prevent dissociation of your servers in your cloud, you can prevent specific servers in your site from being automatically defined the Primary and the Active Arbitrator role.

- V5.0.2 + [Recovering your cloud with a Primary server](#)
  If your network is disrupted or disconnected, and the Primary and Active Arbitrator servers are down, you can re-define the role of the Primary to any active server, until the network is repaired.
- V5.0.2 + [Technical considerations for defining the main site](#)
  During the process of defining the main site in your API Connect cloud, there are multiple technical considerations that you might encounter.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Defining the main site

Dissociation is where a cloud has two or more Primary servers, which can lead to divergence in data across the cloud. To prevent dissociation of your servers in your cloud, you can prevent specific servers in your site from being automatically defined the Primary and the Active Arbitrator role.

About this task

The process of defining the Primary and Active Arbitrator role to servers might take a few minutes. It is recommended to wait until the commands have completed. You can run the `ha list` command to monitor the status of the servers in your site.
For more information on any of the commands or terms that are used in the following procedure, see High availability commands

.

Procedure

1. Optional: Run the `ha list` command to show the status and details of each server in your site.
2. Run the following command on every server that you do not want to define as Primary or Active Arbitrator:

   ```
   ha set host IP_address disabled
   ```

   where *IP_address* is the IP address of the server that you do not want to become the Primary or Active Arbitrator server. The following message is displayed after you run the previous command:

   ```
   Value updated in database.
   Command complete
   Main-site feature is on.
   HA setting of IP_address is now disabled.
   ```

Results

High availability is turned off for the server that you do not want to become the Primary or Active Arbitrator, because a Primary server must have high availability enabled. A server can perform write actions if it is defined as the Primary server role, while the Active Arbitrator role is defined to a different server. The output of running a `ha list` command is similar to the following output:

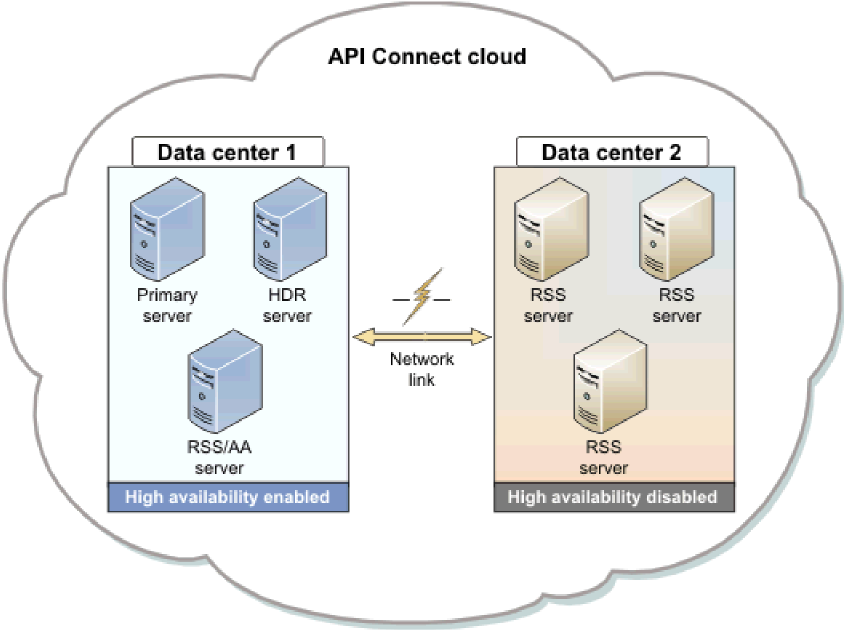
```
> ha list
Self  IP            HA   Network  Role       Runtime  FQDN
Yes   IP_address_1  Yes  Up       Primary    Up       hostname_1.com
No    IP_address_2  Yes  Up       HDR/AA     Up       hostname_2.com
No    IP_address_3  No   Up       RSS        Up       hostname_3.com
```

If you have a site with more than three servers, the output is similar to the following:

```
> ha list
Self  IP            HA   Network  Role       Runtime  FQDN
Yes   IP_address_1  Yes  Up       Primary    Up       hostname_1.com
No    IP_address_2  Yes  Up       HDR        Up       hostname_2.com
No    IP_address_3  No   Up       RSS        Up       hostname_3.com
No    IP_address_4  Yes  Up       RSS/AA     Up       hostname_4.com
```

The last server is the Active Arbitrator because it is not a Primary or HDR, and because it has high availability enabled.

The following diagram shows the composition of a API Connect cloud if the main site is defined: The site on the left has high availability enabled, and the site on the right has high availability disabled.



Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Recovering your cloud with a Primary server

If your network is disrupted or disconnected, and the Primary and Active Arbitrator servers are down, you can re-define the role of the Primary to any active server, until the network is repaired.

About this task

By recovering your cloud with a temporary Primary server, any connected servers are re-cloned so that they can join the new Primary server, and any disconnected servers are re-cloned within three minutes of reconnecting. For more information, see Reference information for assigning the main site.
For more information on any of the commands or terms that are used in the following procedure, see High availability commands

Procedure

1. Run the following command from the server that is not the Primary or Active Arbitrator:

   ```
   ha make primary
   ```

   The following message is displayed:

   ```
   Contacting Primary...
   Current Primary cannot be reached from server IP_address_3.
   This command will reclone the entire cloud with the data on server IP_address_3.
   Are you sure? (yes/no):
   ```

2. Respond to the displayed question with `yes`.
   The following message is displayed as a result:

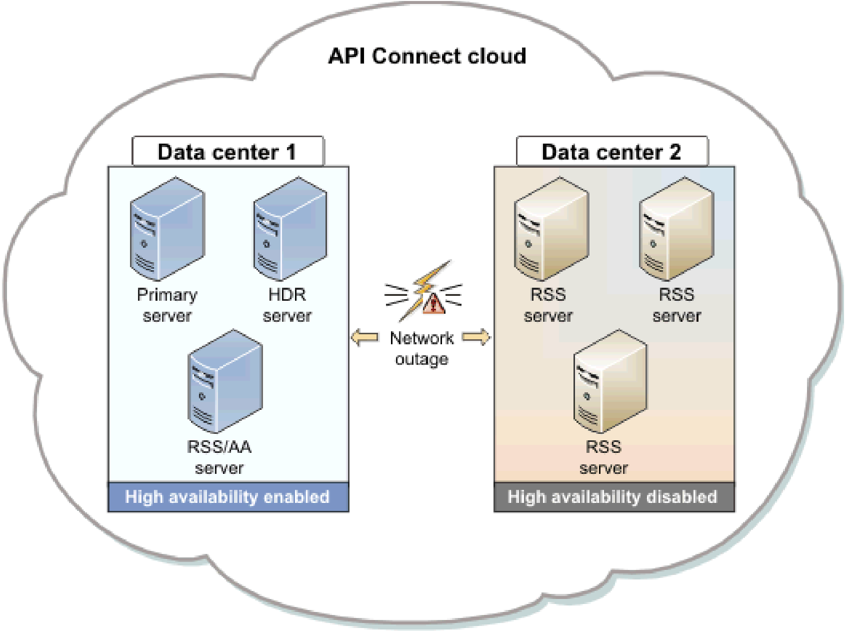
   ```
   Starting Make Primary.
   Note: Do not abort this command.

   Refreshing cloud info...
   Forcing Primary...
   Main-site feature is on.
   Disabling HA on other servers to avoid dissociation. Please reenable as approprate when cloud is
   ```

```
stable.
Recloning cloud...

Make Primary command complete.
```

Important: You must not abort the command. If you abort the command midway, you might have two Primary servers. You cannot run the **ha make primary** command on either Primary server. To solve the issue you can run the **ha make primary** command on another server that is not a Primary server, so that the two Primary servers are re-cloned.
After the command is complete, high availability is turned off for the two servers that are down to avoid re-dissociating the cloud. You must re-enable high availability on the desired servers after the network is stable.

Results

After the network is fixed and is stable, the servers that were previously down will try to start up again. After the servers have restarted, you can enable high availability on the servers, and define the Primary and Active Arbitrator roles. For more information on assigning the roles, see Assign the main site.
Another reason to re-clone your cloud is that if your cloud goes down, you can bring the cloud back into working order by running the **ha make primary** command on one of the remaining servers.
Note: After running the **ha make primary** command, the old Primary server is re-cloned and becomes RSS or HDR. It will not resume its role as the Primary server.
The following diagram shows the composition of a API Connect cloud if there is a network outage, and the site is recovered:
Note: The site on the left has high availability enabled, and the site on the right has high availability disabled.



Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Technical considerations for defining the main site

During the process of defining the main site in your API Connect cloud, there are multiple technical considerations that you might encounter.

Role dependent priority

For more information on any of the commands or terms that are used in this topic, see High availability commands.

Any server can be the Active Arbitrator, including the Primary server. However, there is a specific order of priority as to which server is defined as the Active Arbitrator, where 1 is the most preferred, and 3 is the least preferred:

1. RSS
2. HDR

3. Primary

The role of the Active Arbitrator is to handle the situation where Primary or HDR servers, or both, are down. A server can be a Primary and an Active Arbitrator, but it is not common and occurs only when there is a single active server remaining in the cloud. The Active Arbitrator is also responsible for promoting HDR.

Sequential re-clone

Only one server can clone from Primary at any given time, and therefore all other servers are down until they are able to re-clone.

Unable to backup from HDR

You cannot take backups from HDR. If you attempt to take backups from HDR, the following message is displayed:

```
config save apiconfig [...]
Informix does not allow backups on HDR.
Please take a backup from Primary or RSS instead.
The command 'ha list' will display each server's role.
Operation failed
```

Controlled failover

If you have a Primary server and want to assign the Primary role to a different server, you can run the **ha make primary** command while the current Primary server is still accessible to the target server. This is known as controlled failover.
Note: You can still alter high availability settings after running the **ha make primary** command.
If the Primary server cannot be reached when you run the **ha make primary** command, the local server forcefully takes over and re-clones every other server.

Running "ha make primary" on Primary

If you run the **ha make primary** command on the Primary, the following message is displayed:

```
ha make primary
Attempt to make primary on primary. Exiting.
```

Running ha make primary on two different RSS servers simultaneously while existing Primary is down

If you run the **ha make primary** command simultaneously on two RSS servers that are not Primary or Active Arbitrator servers, the server that has the command run on it more recently becomes the new Primary server.

HDR disconnecting from the Primary server

If HDR cannot be contacted for a long enough period of time, the Active Arbitrator server will promote a new HDR from among the remaining servers, and restart the old HDR when it reconnects.

High availability settings when adding servers

The **ha set default** and **ha show default** commands enable you to set the high availability value that is assigned to servers upon addition to the cloud.

Deleting a server from the cloud

- If the server is an HDR, a new HDR is promoted. If the server is an Active Arbitrator, a new Active Arbitrator is chosen. If RSS, nothing happens.
- If failover is happening and there is no writable database at this time, servers cannot be deleted. If HDR is deleted before a Primary server is lost, another HDR is chosen and it will take over instead.
- If re-cloning is happening, the Primary server removes the server from the list of servers that must be re-cloned.
- If you have a three server site where two servers are connected to each other and one is disconnected from them, and one of the connected servers is deleted, the disconnected server has **ha make primary** run on it. After the two sites resume communication, the new Primary server does not know about the deletion and attempts to re-clone the deleted server.

  - If the deletion completed successfully, the new Primary server is unable to connect to the deleted server. The new Primary continues to retry and times out until the server is deleted again from new Primary.
  - If the deletion did not complete successfully, the new Primary server might be able to re-clone the server. If deletion is still desired, simply delete it again.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 topic:

- Guidelines for creating a DataPower configuration that extends Gateway server behavior

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 IBM 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 by using the Cloud Manager.
4. Copy your exported configuration .zip file to a centralized file system that the Cloud Manager can access. For more information, see Configuring your IBM DataPower Extensions.

- **Guidelines for creating a DataPower configuration that extends Gateway server behavior**
  Before you develop your extensions, consider the following guidelines.

Related reference

- Guidelines for creating a DataPower configuration that extends Gateway server behavior

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Guidelines for creating a DataPower configuration that extends Gateway server behavior

Before you develop your extensions, consider the following guidelines.

You can create DataPower® Processing Rules that can extend the enforcement behavior of the API Connect Gateway server at the following locations:

- `pre-request` extension:
  - Before the Gateway server begins processing the request with policies in the assembly.
- `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.

Related tasks

- Extending the Gateway server behavior

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Preserve your cloud data

It is important to preserve your IBM® API Connect cloud and data. Take regular backups by using the IBM API Connect command-line interface (CLI) command provided. Or, take snapshots by using your virtualization provider tools to provide a fallback if required.

- **Creating a backup of an API Connect configuration**
  You can back up your API Connect configuration and user data, and store the file on an FTP server by using the command-line interface (CLI).
- **Restoring an API Connect configuration**
  As part of your disaster recovery plan, you can restore your API Connect configuration from a file that is stored on an FTP server by using the command-line interface (CLI). Restoring an API Connect configuration involves creating a new on-premises cloud.
- **Backing up an IBM API Connect cloud**
  You can back up your IBM API Connect on-premises cloud by using the tools that are provided by your virtualization environment.
- **Restoring an API Connect cloud**
  You can revert to a previous configuration of your IBM API Connect on-premises cloud by restoring servers that have been backed up.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a backup of an API Connect configuration

You can back up your API Connect configuration and user data, and store the file on an FTP server by using the command-line interface (CLI).

About this task

To provide a fallback, you can create a backup of your API Connect configuration and store the file on an FTP server. All configuration information from the Cloud Manager, and API Manager, is backed up (which includes all the user data entered in these user interfaces). Analytics data is not backed up. In the future, you can restore the API Connect configuration. For more information, see Restoring an API Connect configuration.

The Developer Portal configuration is backed up separately. For more information, see Backing up and restoring the Developer Portal.

Note: You must take a backup from the Primary Management server in the Management cluster. To identify which server is the primary server, open the Cloud Manager and, in the Clusters page, click the Server details icon for each server in the Management cluster. The Management servers in a cluster automatically synchronize so only the primary Management server backup is required.
Important: You can only restore configurations that are at the same fix pack level that the back up was taken on.
V5.0.6 + By default, the Management server automatically takes a daily configuration backup at 00:00 UTC, and stores it locally. The performance impact of taking a configuration backup is minimal, and the maximum disk space used for these daily backups is by default 500 MB on /wip. You can access this backup by using CLI commands; for details, see Configuration commands. To turn off the automatic configuration backup process, use the following command:

```
config autobackup disable
```

Procedure

1. Log in to the CLI for the Management server through a Secure Shell (SSH).
   For information on logging into, and using the CLI, see The Command Line Interface.
2. To create a backup of an API Connect configuration and user data, and store the file on an FTP server, you can use either FTP or SFTP.
   For FTP, enter the following command:

   ```
   config save apiconfig ftp  <host> [port <portnumber>] [user <username>] [file <filename>]
   ```

   For SFTP, enter the following command:

   ```
   config save apiconfig sftp  <host> [port <portnumber>] user <username> [file <filename>]
   ```

   Where
   - *host* - the name of the host where FTP is running and where the backup configuration file is stored.
   - port - the port number that you are accessing.
   - user - the user name that is used to log in to FTP.
   - file - the absolute path or relative path to the stored file on the FTP server.

   Restriction: When you use SFTP, you must specify a user name. Anonymous logins are not allowed with SFTP.
   To view the CLI help about the **config** commands, enter **config help**.
3. Make a note of the user ID or password of your Management servers at the time of the backups.
   If you change your credentials in the future, you must know the correct login details when you restore the configuration. You might have to update the servers by using the API Connect cloud console. Otherwise, the cloud console cannot communicate with the servers.

Related tasks

- Restoring an API Connect configuration

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restoring an API Connect configuration

As part of your disaster recovery plan, you can restore your API Connect configuration from a file that is stored on an FTP server by using the command-line interface (CLI). Restoring an API Connect configuration involves creating a new on-premises cloud.

About this task

You can restore a previous version of your API Connect configuration by using an earlier backup file that you created and stored on an FTP server. For more information, see Creating a backup of an API Connect configuration. All configuration information from the Cloud Manager, API Manager, and Developer Portal is backed up, and can therefore be restored. Analytics data is not backed up.

V5.0.6 + You can also load a configuration backup file that has been created by the automatic configuration backup process, by using the `local file` option on the **config load apiconfig** command. To see the names of the available configuration backup files, use the **config autobackup list** command.

V5.0.8 + From API Connect Version 5.0.8.3 onwards, a **[restore | isolate]** option is available on the **config load apiconfig** command. This option means you can select whether to load a previous version of your API Connect configuration in restore mode or in isolation mode. Use the `restore` option to load the management configuration file as is. Use the `isolate` option to load the management configuration file in isolation, in other words without any references to DataPower® Gateway servers, Developer Portal servers, or any third-party systems for analytics offload. Isolation mode is especially useful when you are testing an upgrade process. For more information, see Testing the upgrade process and path.

The following procedure outlines the three main options for restoring an API Connect configuration:

- Restore mode - load the management configuration file as is.
- V5.0.6 + From a local file - load a configuration backup file that has been created by the automatic configuration backup process.
- V5.0.8 + Isolation mode - load the management configuration file without any references to DataPower Gateway servers, Developer Portal servers, or any third-party systems for analytics offload.

For more information about the **config load apiconfig** command, see Configuration commands.

Procedure

1. For each Management server in your on-premises cloud, log in to the CLI through a Secure Shell (SSH), and clear their system states by running the **system clean apiconfig** command.
   Running this command deletes data and configurations from the Management server, so must be run with caution. However, it is a necessary step so that the other Management servers in the cloud do not conflict with the newly restored system by trying to contact it. If `system clean apiconfig` doesn't reboot the Management server, then run the `system reboot` command, so that memory and temporary disk space is cleared in readiness for the restoration.
   For information on logging in to, and using the CLI, see The Command Line Interface.

2. Log in to one of the Management servers through a Secure Shell (SSH).
   This Management server is the first Management server in a new on-premises cloud. Following the restore, this Management server is populated with the details from the backup configuration file, including the organizations, policies, and Products.

3. Restore an API Connect configuration file by using one of the following options:
   - Restore mode from a file that is stored on an FTP server (you can use either FTP or SFTP).
     For FTP, enter the following command:

     V5.0.7 and earlier

     ```
     config load apiconfig ftp  <host> [port <portnumber>] [user <username>] [file <filename>]
     ```

     V5.0.8 +

     ```
     config load apiconfig restore ftp  <host> [port <portnumber>] [user <username>] [file
     <filename>]
     ```

     For SFTP, enter the following command:

     V5.0.7 and earlier

     ```
     config load apiconfig sftp  <host> [port <portnumber>] user <username> [file <filename>]
     ```

     V5.0.8 +

     ```
     config load apiconfig restore sftp  <host> [port <portnumber>] user <username> [file
     <filename>]
     ```

     Where
     - V5.0.8 + restore - means that the API management configuration file is loaded as is. (Note that restore mode is selected by default if no option is given.)
     - *host* - is the name of the host where FTP is running and where the backup configuration file is stored.
     - port - is the port number you are accessing.
     - user - is the user name that is used to log in to FTP.
     - file - is the absolute path or relative path to the stored file on the FTP server.

     Restriction: When you use SFTP, you must specify a user name. Anonymous logins are not allowed with SFTP.
   - V5.0.6 + By using a local configuration backup file that was created by the automatic configuration backup process.
     Enter the following command:

     ```
     config load apiconfig local file <filename>
     ```

     Where ***filename*** must be a file that was created by the automatic configuration backup process. To see the names of the available configuration backup files, use the **config autobackup list** command.
   - V5.0.8 + Isolation mode from a file that is stored on an FTP server (you can use either FTP or SFTP).
     For FTP, enter the following command:

     ```
     config load apiconfig isolate ftp  <host> [port <portnumber>] [user <username>] [file
     <filename>]
     ```

     For SFTP, enter the following command:

     ```
     config load apiconfig isolate sftp  <host> [port <portnumber>] user <username> [file
     <filename>]
     ```

     Where
     - isolate - means that the management configuration file is loaded without references to DataPower Gateway servers, Developer Portal servers, or any third-party systems for analytics offload.
     - *host* - is the name of the host where FTP is running and where the backup configuration file is stored.
     - port - is the port number you are accessing.
     - user - is the user name that is used to log in to FTP.

    - file - is the absolute path or relative path to the stored file on the FTP server.

  Restriction: When you use SFTP, you must specify a user name. Anonymous logins are not allowed with SFTP.
  Note: When you restore your configuration in isolation mode, or for testing an upgrade if the new management server cannot reach the original portal server(s), and you want the Developer Portal in the restored configuration to be able to work with the existing catalogs, you must perform the following steps in the newly restored API Manager UI:

    - For each Catalog, select Settings > Portal, and change the Developer Portal setting to None. Save your changes.
    - When you want to enable a Developer Portal, for each Catalog you must complete the Settings > Portal section with the new host name for the Developer Portal URL. Save your changes.

  To view the CLI help about the **config** commands, enter `config help`.

4. You must wait until the primary Management server is restored and active before you can add other Management servers to the cluster by using the cloud console.
   Note: It can take several minutes for a virtual machine to initialize following the restore of a backup.
   Important: If you want to reuse any Management servers in the new on-premises cloud, you must first delete the existing API Connect configuration. From the CLI, for each of the Management servers, enter **system clean apiconfig**. Alternatively, you can deploy new virtual Management servers to add to your cloud.
   When the other Management servers are active and defined in the on-premises cloud, the primary Management server automatically synchronizes the API Connect configuration across all of the Management servers in the cluster.

5. For Gateway servers only, if you changed the user ID or password, you must update these credentials in the API Connect cloud console.
   Important: The restored configuration backup includes the Gateway server credentials that were defined in the cloud when the backup file was created. If you do not update the cloud console to change these credentials, the cloud console cannot communicate with the newly restored Gateway servers.

Related tasks

- Creating a backup of an API Connect configuration

Related reference

- Configuration commands

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Backing up an IBM API Connect cloud

You can back up your IBM® API Connect on-premises cloud by using the tools that are provided by your virtualization environment.

About this task

To preserve your data, in case of disaster recovery, you must back up each of the Management appliances that you have defined in your development, test, or production environments. These appliances are defined in the cloud console as Management servers.

The Management servers contain important API Connect details. The details include the record of your organization accounts, the APIs, details of the Products, and the applications that your developers registered that call your APIs.

Note: Take snapshots of all of the servers within a cluster in quick succession to ensure that the backups of each appliance contain a consistent view of the state of the cloud. For example, take snapshots of all of the Management appliances at the same time. Note that snapshots are not the same as backups. A snapshot is a change log of the original virtual disk, and is not portable. Therefore, do not rely on it as your only backup process. The virtual machine runs on the most current snapshot, not the original vmdk disk files. For optimum recovery readiness, you should take both snapshots and backups of your configuration.
For VMware, you can take one of the following approaches:

- For short-term scenarios such as taking a checkpoint immediately before you apply a fix pack upgrade, you can choose to take a VMware snapshot of the instances. You can then delete the snapshot after a few days after you confirm that the changes were successful.
- For long-term backup and restore, consult your VMware administrator about your organizations approach to VMware instance backup.

Important: Only the following backup mechanisms are supported:

- Performing a configuration backup; for more information, see Creating a backup of an API Connect configuration.
- Taking a VMware snapshot.

Virtual machine cloning and third-party solutions are **not** supported.

Procedure

There are two approaches that you can take when backing up your IBM API Connect cloud: offline backup, and online backup.

- To carry out an offline backup, complete the following steps.
  1. Turn off all Management servers.
  2. Take a snapshot of all Management servers, Developer Portal servers, and, if possible, all Gateway servers.
  3. Turn on all servers.
- To carry out an online backup, complete the following steps, ensuring that you capture the primary server last so that its data is completely up to date.
  1. Determine which Management server is the primary server by opening the Cloud Manager and, in the Clusters page, clicking the Server details icon for each server in the Management cluster. The primary server has the role PRIMARY, and the secondary servers have the role RSS.
  2. If possible, take a snapshot of all virtual Gateway servers.
  3. Take a snapshot of all secondary Management servers.
  4. Take a snapshot of the primary Management server.
  5. Take a snapshot of all Developer Portal servers.
     To avoid the risk of database reclustering issues on restoration, when you take a snapshot of a Developer Portal cluster, the database of each server should be stopped before the snapshot is taken, and then restarted again after. In this way, you can achieve a rolling snapshot of your cluster with zero downtime. If you must restore all the snapshots at the same time, then the cluster will automatically bootstrap itself with the data from the last server that was snapshot. For more information about taking snapshots of Developer Portal servers, see Backing up and restoring the Developer Portal.

What to do next

Schedule regular backups of your Management servers.

Related tasks

- Restoring an API Connect cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restoring an API Connect cloud

You can revert to a previous configuration of your IBM® API Connect on-premises cloud by restoring servers that have been backed up.

About this task

This procedure describes how to restore an entire API Connect cloud from backup, eliminating any existing Management servers and replacing them with Management servers captured at a previous time. As a result, the contents of the API Connect cloud, such as Users, APIs, and Products, are reverted to their state at the time of backup.

For VMware, restore the appropriate snapshot or backup, depending on the mechanism that you originally used to back up the virtual machine.

Note: If you restore only a subset of the Management servers in your cloud, the servers that are not restored cannot continue to run alongside the restored servers. Therefore, for best results, restore **all** Management servers. If, however, you do not restore all Management servers, the cloud can function provided that you restore the following minimal subset of servers:

- The Management server that was the primary server at the time of the backup operation.
- At least half of the Management servers that were secondary servers at the time of the backup operation.

For example, if the cloud had three Management servers at the time of the backup operation, you must restore the primary server and at least one of the secondary servers. If you do not restore a sufficient number of Management servers, the cloud will not function.

Procedure

1. Turn off all running Management servers.
2. Turn off all running Developer Portal servers.
3. (Optional): If you have a backup of the DataPower appliances, you can restore them at this point.
4. Restore the Management server that was the primary server at the time of the backup operation.
5. Restore all the Management servers that were the secondary servers at the time of the backup operation.
6. Restore all Developer Portal servers.
7. If you changed the user ID or password of any of your Management servers, you must update these credentials in the API Connect cloud console.
   Important: The restored backup includes the credentials that are defined in the cloud when the snapshot was captured. If you do not change these credentials to match the newly restored servers, the cloud console cannot communicate with the servers.
8. If you restored your DataPower appliances with backup data not taken at the same time as the Management backup, you must remove and re-add each Gateway Server to their Gateway Clusters. This ensures that each Gateway Server contains up-to-date information. If the DataPower appliance was restored from a different backup, after you remove the Gateway Server in the CMC, you may also need to delete the APIMgmt domain in the appliance to ensure it is added back correctly.

Related tasks

- Backing up an IBM API Connect cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The Command Line Interface

The IBM® API Connect Command Line Interface (CLI) is available to administrators to manage the Management server and to maintain and update its configuration information.

The CLI is preinstalled on the Management server and is ready to respond to the commands described in this documentation.

Note: The CLI is case-sensitive. You must enter commands and keywords using lowercase characters.
Only a user that is logged in as an administrator can use the CLI. More than one administrator can be logged in to a Management server at the same time.
Important: The Command Line Interface is distinct from the developer toolkit command line interface. For more information on the developer toolkit command line interface, see Getting started with the developer toolkit command-line tool.

Logging in to the CLI

You can log in to the CLI on the Management server through a Secure Shell (SSH) session, terminal emulation, or telnet. By default, secure management is enabled on the Management server and you must use secure connections to access the Management server. Secure connections use the default host key that is provided with the Management server at installation.

When you use an SSH client or a telnet client to log in to the CLI, the 5th consecutive log-in failure (regardless of which user or any span of time) triggers a lock-out. The lock-out is characterized by a limit of 1 login attempt per minute without regard for the user. A successful log-in by any user resets the log-in failure count and ends the lock-out.

Logging out of the CLI

At the command prompt, use the **exit** command to log out of the CLI. This also closes the terminal emulation or telnet session connection.

Viewing CLI help

In the CLI, you can view a list of command categories by typing **help** or **?**. To view syntax help for the commands in a specific category, enter `*Category* help`. For example, `net help` lists syntax information for each network command.

Tab completion also guides you through the CLI syntax. From the CLI, you can enter a partial command and press the Tab key. Pressing the Tab key completes the command or provides a list of options to complete the command syntax

Note: Tab completion is sensitive to spaces in the CLI syntax. For example, typing system show in the CLI and then pressing the Tab key appears to have no results. Add a space after show and press the Tab key to view valid command parameters.

Command conventions

The following conventions are used to illustrate command syntax rules:

Table 1.

Convention	Description
keyword *<value>*	Most command parameters combine a keyword and a value. Some parameters might not require a value.
<value>	Values for parameters are enclosed in angle brackets. In many cases, the text that is shown indicates the type of information you supply, such as *<hostname>*. Values can be explicit, such as *<yes>*
[x]	Optional parameters are enclosed in brackets.
{x y z}	Groups of mandatory parameters are enclosed in braces.
x | y | z	Choices are separated by bars, select only one.
x...	Parameters that might occur more than once are followed by an ellipse.

- **Logging in to the CLI with a Secure Shell session connection**
  By default, secure management is enabled on the Management server and you must use secure connections to access the Management server. Secure connections use the default host key that is provided with the Management server at installation. You can log in to the CLI on the Management server by using a Secure Shell session (SSH).
- **Logging in to the CLI with VMware**
  You can log in to the CLI on the Management server by using a VMware vSphere console.
- **Logging in to the CLI with a XenServer**
  You can log in to the CLI on the Management server by using a XenServer console.
- **Logging in to the CLI with a telnet connection**
  When secure management access is disabled for the Management server, you can connect to the Management server via the network through a telnet session.
- **Command Line Interface commands**
  A list of the various command categories available from the Command Line Interface (CLI), and describes the purpose of each command category:
- **Authorization commands**
  Use authorization commands to manage user accounts.
- **Configuration commands**
  Use configuration commands to manage both management and integration configuration for the Management server.
- **Debugging commands**
  Use debugging commands to view queues and process stacks.
- **Management commands**
  A list of the management (mgmt) commands available.
- **Network commands**
  Network commands that you can use to manage the network configuration.
- **Network introspection commands**
  Network introspection (Netspect) commands allow you to manage network configurations and query the network for DHCP server, DNS server, gateway, and route information.
- **Status commands**
  Status commands allow you to view the Management server status.
- **System commands**
  System commands allow you to manage the operation of the Management server, which includes managing licenses for the Management server and connectors.
- **Time commands**
  Time commands allow you to set or synchronize the date and time on the Management server.
- **High availability commands**
  Use high availability commands to monitor the status of all Management servers in your IBM API Connect cloud.
- **Supported key algorithms, ciphers, and MACs**
  API Connect supports a range of key algorithms, ciphers, and MACs when connecting to or from the management server.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Logging in to the CLI with a Secure Shell session connection

By default, secure management is enabled on the Management server and you must use secure connections to access the Management server. Secure connections use the default host key that is provided with the Management server at installation. You can log in to the CLI on the Management server by using a Secure Shell session (SSH).

Procedure

1. In Windows, start the application that performs SSH connections.
   For example, open PuTTY.
2. Enter the *management server-host-name*.
3. If needed, select SSH as the connection protocol or set one of the other connection parameters that are offered by the application.
4. Initiate the connection. The login prompt is displayed at first access.
5. Log in to the Management server with a user name and password, which have administrative privileges. The default user name and password is:
   - Login: `admin`
   - Password: `!n0r1t5@C`
     Note: 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, please ensure that you have typed the password correctly.

Results

When you log in, the prompt changes to the host name of the Management server. For example, ***management server-host-name*/APIManagement**.

If you do not select a host name, the prompt defaults to the IP address of the Management server. For example, ***ip-x-xx-xxx-xx*/APIManagement**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Logging in to the CLI with VMware

You can log in to the CLI on the Management server by using a VMware vSphere console.

Procedure

1. Open your VMware vSphere client.
2. Enter the following credentials.
   - IP address or name of the host or vCenter Server.
   - User name
   - Password
3. Optional: Use your Windows session credentials to log in by clicking the Use Windows session credentials check box.
4. Click Login.
5. Right-click the name of the appliance that you want to work with.
6. Click Open Console.
7. Log in to the Management server with a user name and password, which have administrative privileges. The following lists the default user name and password:
   - Login: `admin`
   - Password: `!n0r1t5@C`

   Note: 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, please ensure that you have typed the password correctly.

Results

You see the following prompt ***management server-host-name*/APIManagement** and you can start entering commands.

Logging in to the CLI with a XenServer

You can log in to the CLI on the Management server by using a XenServer console.

Procedure

1. Open the XenCenter client.
2. Connect to your XenCenter instance.
3. Enter your user name and password then click Connect.
4. Click the name of the appliance that you want to work with.
5. Click the Console tab.
6. Click anywhere within the console area and press enter.
7. Log in to the Management server with a user name and password, which have administrative privileges. The following lists the default user name and password:
    - Login: `admin`
    - Password: `!n0r1t5@C`

   Note: 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, please ensure that you have typed the password correctly.

Logging in to the CLI with a telnet connection

When secure management access is disabled for the Management server, you can connect to the Management server via the network through a telnet session.

About this task

Note: By default, secure management access is enabled for the Management server; therefore, you can not connect to the Management server by using a telnet connection. For more information about enabling and disabling secure management access, see mgmt secure.

Procedure

1. In Windows, select Start > Run.
2. Enter `telnet` *`management server-host-name`*.
   The login prompt is displayed at first access.
3. Log in to the Management server with a user name and password, which have administrative privileges. The default user name and password is:
    - Login: `admin`
    - Password: `!n0r1t5@C`
      Note: 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, please ensure that you have typed the password correctly.

Results

When you log in, the prompt changes to the host name of the Management server.

If you have not selected a hostname, the prompt defaults to the IP address of the Management server.

Command Line Interface commands

A list of the various command categories available from the Command Line Interface (CLI), and describes the purpose of each command category:

Table 1.

Command Category	Description
Auth Commands	Use authorization commands to manage user accounts.
Config Commands	Use configuration commands to manage both management and integration configuration for the Management server.
Debug Commands	Use debugging commands to view queues and process stacks.
Mgmt Commands	Use management commands to enable and manage security settings.
Net Commands	Use network commands to manage network configuration.
Netspect Commands	Use network introspection commands to manage network configurations and query the network for DHCP server, DNS server, gateway, and route information.
Stat Commands	Use status commands to view Management server status.
System Commands	Use system commands to manage the operation of the Management server, which includes managing licenses for the Management server and connectors.
Time Commands	Use time commands to set or synchronize the date and time on the Management server.

Authorization commands

Use authorization commands to manage user accounts.

Table 1.

Command name	Action	Syntax
auth set user	The **auth set user** command changes the admin password for the Management server.	`auth set user <user>`
auth set recovery	The **auth set recovery** command allows the admin password to be reset from the serial console for a Management server. The default is on If the option is enabled, you can type `resetpass` for the user name at the serial login and any non-empty password. Immediately after you log in, you will be asked to access the appliance. To verify access, disconnect the Network adapter, then reconnect it (see Note). This verifies that the serial console is not being accessed over a remote port replicator. If you do not have access to the appliance, you must set the recovery option to off.	`auth set recovery <on|off>`
auth show recovery	The **auth show recovery** command shows whether recovery is on or off. In the on state, the admin password can be reset from the serial console for a Management server. If the option is enabled, you can type `resetpass` for the user name at the serial login and any non-empty password. Immediately after you log in, you will be asked to access the appliance. To verify access, disconnect the Network adapter, then reconnect it (see Note). This verifies that the serial console is not being accessed over a remote port replicator. If you do not have access to the appliance, you must set the recovery option to off.	`auth show recovery`

Note: To disconnect and reconnect the Network adapter, modify the adapter settings of the Management server.

For example, in VMWare, complete the following steps:

1. Login to the VMWare client.
2. Right click the Management server virtual appliance and select Edit Settings.
3. In the hardware settings for the network adapter, clear the Connected check box, then save the settings.
4. Repeat steps 2 and 3, selecting the Connected check box to reconnect the network adapter..

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuration commands

Use configuration commands to manage both management and integration configuration for the Management server.

Table 1. Configuration commands

Command name	Action	Syntax
V5.0.8 + **config dbextract** (from IBM® API Connect Version 5.0.8.5)	V5.0.8 + Export the contents of the Management server configuration database in JSON format, in a .tar file.	V5.0.8 + `config dbextract sftp <sftp_host_name> user <username> file [<path>/]<filename>`

Command name	Action	Syntax
config load apiconfig	The **config load apiconfig** command loads a management configuration file from an FTP or SFTP server. After the configuration file is loaded on the Management server, the Management server automatically reboots. Do not use reserved words in the user name, host name, or file name. Reserved words include: ftp, sftp, ibm, port, user, and file. Note: When you use SFTP, you must specify a user name. Anonymous logins are not allowed with SFTP. V5.0.6 + You can also load a configuration backup file that has been created by the automatic configuration backup process, by using the `local file` option. To see the names of the available configuration backup files, use the **config autobackup list** command. V5.0.8 + From API Connect Version 5.0.8.3 onwards, you can also specify an isolation mode by using the `isolate` option. This option loads the management configuration file in isolation, in other words without any references to DataPower® Gateway servers, Developer Portal servers, or any third-party systems for analytics offload. Isolation mode is especially useful when you are testing an upgrade process. For more information, see Testing the upgrade process and path.	FTP syntax: `config load apiconfig ftp <host> [port <number>] [user <username>] file [<path>/]<filename>` SFTP syntax: `config load apiconfig sftp <host> [port <number>] user <username> file [<path>/]<filename>` Configuration backup file syntax: V5.0.6 + `config load apiconfig local file <filename>` V5.0.8 + Isolation mode syntax: **`config load apiconfig [restore	isolate]`** Where • **`restore`** means that the API management configuration file is loaded as is. (Note that restore mode is selected by default if no option is given.) • **`isolate`** means that the API management configuration file is loaded without references to DataPower Gateway servers, Developer Portal servers, or any third-party systems for analytics offload. For example, to load a management configuration file in isolation mode from an SFTP server, use the following syntax: **`config load apiconfig isolate sftp <host> [port <number>] user <username> file [<path>/]<filename>`**
config reset	Resets the connection rules for the Cloud Manager, Developer Portal, and API Manager. Port is reset to 443 and the SSL Profile resets to Default SSL Profile.	`config reset services`

Command name	Action	Syntax

config save	The **config save apiconfig** command saves a management configuration file to an FTP or SFTP server.
	Note: Do not use reserved words in the user name, host name, or file name. Reserved words include: ftp, sftp, ibm, port, user, and file.
		FTP syntax:
		When you save the configuration file to an FTP server, the Management server uses anonymous if you omit the user name.
		`config save apiconfig ftp <host> [port <number>] [user <username>] file [<path>/]<filename>`
		Example: Saves the current system configuration as old_sys_config.cfg to the FTP server that is running on host tern. The FTP user's login name is Joe.
		`config save apiconfig ftp tern user joe file old_sys_config.cfg`
		SFTP syntax:
		When you save the configuration file to an SFTP server, you must specify a user name. Anonymous logins are not allowed with SFTP.
		`config save apiconfig sftp <host> [port <number>] user <username> file [<path>/]<filename>`

V5.0.6+ **config autobackup**	V5.0.6+ Enable or disable the automatic configuration backup process. This process runs once a day at midnight UTC, and keeps configuration backup files on the local disk up to the maximum allowed space. The default maximum allowed space is 524288000 bytes but you can change this by using the **config autobackup set maxsize** command.	V5.0.6+ `config autobackup enable|disable`

V5.0.6+ **config autobackup status**	V5.0.6+ Show the status of the automatic configuration backup process, and the maximum number of bytes that the configuration backup files are allowed to use on disk.	V5.0.6+ `config autobackup status`
		Example output:
		`Enabled: true`
		`Running: false`
		`Max storage (bytes): 524288000`

V5.0.6+ **config autobackup list**	V5.0.6+ List the configuration backup files that have been created by the automatic configuration backup process.	V5.0.6+ `config autobackup list`

V5.0.6+ **config autobackup export local file**	V5.0.6+ Export a configuration backup file that has been created by the automatic configuration backup process. You can export the file to an FTP or SFTP server.
	To see the names of the available configuration backup files, use the **config autobackup list** command
		V5.0.6+ FTP syntax:
		`config autobackup export local file <filename> ftp <host> [port <number>] [user <username>]`
		SFTP syntax:
		`config autobackup export local file <filename> sftp <host> [port <number>] user <username>`

V5.0.6+ **config autobackup set maxsize**	V5.0.6+ Set the maximum number of bytes on disk that can be used by backup files created by the automatic configuration backup process. The default maximum value is 524288000 bytes.	V5.0.6+ `config autobackup set maxsize <value_in_bytes>`
config autodiskadd	Control the automation of disk addition to the data partition of the management server. When set to *disable*, the data disk that is added to the vm does not become a part of the data partition. When set to *enable*, and the server is rebooted, the data disk is added to the data partition, and increases the data disk size.	`config autodiskadd <enable/disable/status>`

Related tasks

- Restoring an API Connect configuration

Debugging commands

Use debugging commands to view queues and process stacks.

Table 1.

Command name	Action	Syntax
debug postmortem export	The **debug postmortem export** command exports the postmortem archive to the FTP or SFTP Server you specify. The Management server does not require that you specify an export file name; however, as a best practice, you should include the Management server serial number, date, and timestamp. If you do not specify an export file name, the Management server exports a file named postmortem.tar.gz. Note: Do not use reserved words in the user name, host name, or file name. Reserved words include: ftp, sftp, postmortem, debug, export, ibm, port, user, and file.	FTP Syntax: `debug postmortem export ftp <hostname> [port <number>][user <user>][file <filename> ]` where: • *hostname* is the name of the FTP Server. • *number* is the port number to use to connect to the FTP Server. • *user* is the username that is used to log in to the FTP Server. • *filename* is the name of the file that stores the postmortem archive. For example: • `debug postmortem export ftp 192.168.1.2 user user1 file logs/B2XXW56_04152005_172341.tgz` • `debug postmortem export ftp ftpserver.yourcompany.com user user1 file logs/B2XXW56_04152005_172341.tgz` SFTP Syntax: To export the postmortem by using an SSH File Transfer Protocol, use the following SFTP syntax. Note: When you export a postmortem file to an SFTP server, you must specify a user name. Anonymous logins are not allowed with SFTP. `debug postmortem export sftp <hostname> [port <number>] user <user> [file <filename> ]` where: • *hostname* is the name of the SFTP Server. • *number* is the port number to use to connect to the SFTP Server. • *user* is the username that is used to log in to the SFTP Server.

Command name	Action	Syntax
		• *filename* is the name of the file that stores the postmortem archive. For example: • `debug postmortem export sftp 192.168.1.2 user user1 file logs/B2XXW56_04152005_172341.tgz` • `debug postmortem export sftp ftpserver.yourcompany.com user user1 file logs/B2XXW56_04152005_172341.tgz`
debug postmortem generate fulllogs	The **debug postmortem generate fulllogs** command generates a postmortem archive of all the Management server logs, which includes new logs in addition to archived logs, and possibly several stacks that are generated by previously running the **debug show stack** command. The postmortem archive remains on the Management server until you reissue the command. When you issue this command, the Management server generates a new archive that overwrites the previous postmortem archive.	`debug postmortem generate fulllogs`
debug postmortem generate newlogs	The **debug postmortem generate newlogs** command generates a postmortem archive of all the latest Management server logs and possibly several stacks that are generated by previously running the **debug show stack** command. The postmortem archive remains on the Management server until you reissue the command. When you reissue this command, the Management server generates a new archive that overwrites the previous postmortem archive.	`debug postmortem generate newlogs`
debug show stacks	The **debug show stacks** command displays current runtime stack traces.	`debug show stacks`
debug system	Use the **debug system** command to start and stop the Management server run time. When you issue the debug system stop command, all running orchestration jobs are canceled and the Management server does not process any new orchestration jobs.	`debug system {start|stop }`
debug tail file	The **debug tail file** command displays log file contents as they are added to the log.	`debug tail file<`*`filename`*`>`
debug top	The **debug top** command dynamically displays process status.	`debug top`

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Management commands

A list of the management (mgmt) commands available.

Note: If you specify community names that require quotation marks, you must use double quotation marks.

Table 1.

Command name	Action	Syntax
mgmt motd	Use this command to set a message of the day that is displayed when a user logs in to the CLI. You can view the message of the day that was set, by choosing the show parameter. Choose the clear parameter and no message of the day is displayed when a user logs in to the CLI.	**`mgmt motd {set|show|clear}`**
mgmt secure	Enables or disables Telnet access to the Management server. Verify the current security setting by using the **mgmt secure setting** command	**`mgmt secure { on	off	setting}`**

Command name	Action	Syntax
mgmt snmp set	Use this command to set up or update current SNMP configurations, and to enable and disable the SNMP MIB2 service.	1. `mgmt snmp set { community | description | location | contact } <value>` 2. `mgmt snmp set { on | off }`
mgmt snmp show	Displays current SNMP configuration settings.	`mgmt snmp show { community | description | location | contact | all | status }`
mgmt ssh add	Adds a public key for the specified host to the SSH Store on the Management server. If strict SSH key checking is enabled on the Management server, public keys for all participating endpoint systems that the Management server connects to with SFTP must be stored in the SSH Store. Restart the Management server after you enter this command. Note: The keytype is a required parameter.	`mgmt ssh add <rsa|dsa > host <name|ipaddr|name,ipaddr>`
mgmt ssh del	Deletes the public key for the specified host from the SSH Store on the Management server. Restart the Management server after you enter this command. Note: The keytype is a required parameter.	`mgmt ssh del host <hostname>|host IP address>`
mgmt ssh keycheck	Enables or disables strict SSH key checking. In strict mode, SFTP connectors connect to participating endpoint systems, only if the public key for that participating endpoint system matches the public key that is stored in the SSH Store of the Management server. In easy mode, SFTP connectors connect to participating endpoint system if either: • The public key for that system matches the public key that is stored in the SSH Store • No public key exists in the SSH Store. If no key exists, the connector loads the current public key from the participating endpoint system to the SSH Store	`mgmt ssh keycheck<strict|easy >`
mgmt ssh list	Displays all or a specified host that has public keys in the SSH Store. Public keys in the SSH Store are used with SFTP connectors to ensure secure connections.	`mgmt ssh list [host <name | host IP address>}`
mgmt ssh show keycheck	Displays the current setting of SSH key checking: strict or easy.	`mgmt ssh show keycheck`
mgmt syslog del config	To delete system log (syslog) configurations.	`mgmt syslog del config` See Note[1].
mgmt syslog set remote host	To specify the host name of the remote host to which OS level health information is sent. Optionally, specify the port number on the remote host.	`mgmt syslog set remote host <host name> [port <portnum>]` See Note[1].
mgmt syslog set remote ip	To specify the IP address of the remote host to which OS level health information is sent. Optionally, specify the port number on the remote host.	`mgmt syslog set remote ip <host IP address> [port <portnum>]` See Note[1].
mgmt syslog show config	Displays current system log (syslog) configuration settings.	`mgmt syslog show config` See Note[1].
mgmt authkeys set	Use this command to write scripts against API Connect appliances. For more information, see Shell scripts, public keys and API Connect appliances.	`mgmt authkeys set`
mgmt authkeys show	Use this command to displays all authkeys that have been added.	`mgmt authkeys show`
mgmt authkeys clear	Use this command to clear all authkeys that have been added.	`mgmt authkeys clear`

Command name	Action	Syntax
V5.0.2 + **mgmt authkeys append**	Use this command to add authorization keys without deleting existing keys. For more information, see [Shell scripts, public keys and API Connect appliances]().	`mgmt authkeys append` and, when prompted, enter the relevant authorization key.

- **Shell scripts, public keys and API Connect appliances**
  Your API Connect appliances define your cloud and determine how to distribute the work of managing, analyzing, routing, and storing data. Effectively managing and monitoring the health of your appliances is critically important. Shell scripts, which can contain both operating system commands and shell built-in commands, provides you with an easy way to carry out a number of tasks, including enabling public key logins, displaying keys and disabling and clearing of public keys. Public key authentication is an alternative means of identifying yourself to a login server, eliminating the need to enter a password.

Related reference

- Supported key algorithms, ciphers, and MACs

[1] The **syslog** command sends syslog messages by using UDP **only**; TCP is **not** supported.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Shell scripts, public keys and API Connect appliances

Your API Connect appliances define your cloud and determine how to distribute the work of managing, analyzing, routing, and storing data. Effectively managing and monitoring the health of your appliances is critically important. Shell scripts, which can contain both operating system commands and shell built-in commands, provides you with an easy way to carry out a number of tasks, including enabling public key logins, displaying keys and disabling and clearing of public keys. Public key authentication is an alternative means of identifying yourself to a login server, eliminating the need to enter a password.

Enabling public key login

Public keys provide a secure way of logging into a virtual private server. You can place the public key on any server and then unlock it by connecting to it with a client that already has the private key.

Procedure

To enable a public key login, complete the following steps:

1. Open a Command Line Interface (CLI) session on the appliance and run the following command:

   ```
   mgmt authkeys set
   ```

2. Paste at least one public key.
3. Press CNTRL-D *twice* to save the key. If you wish to cancel the save, press CNTRL-C.

Results

The public key is enabled.

Displaying authorized public keys

After you successfully import your key, you can confirm its status. You may also want to view a listing of all your public keys.

Procedure

1. To display your current authorized public keys, enter the following command:

   ```
   mgmt authkeys show
   ```

2. From the appliance from which you took the public key, log in to the appliance.
   Your key is authenticated and you are granted access. Providing a password is not required.

3. From the appliance from which you took the public key, execute a CLI command, for example:

   ```
   system show status
   ```

   The system status is displayed:

   ```
           System: Up
           Network:        Up
           Runtime:        Up
   ```

   Note: The ssh opens a single CLI session and closes the session after running the command. An example of a simple script follows:

   ```
   jdoe@doe-desktop:~$ cat collectinfo.sh
   #!/bin/sh
   RUNCMD="ssh admin@9.30.192.108"
   $RUNCMD net show hostname
   $RUNCMD system show status
   jdoe@doe-desktop:~$ ./collectinfo.sh
   Active Hostname (DHCP from eth0): ip-9-30-192-108

   Appliance Status
   ----------------
           System: Up
           Network:        Up
           Runtime:        Up
   ```

   Note: Attempting to execute the script on a single line fails, as shown in the following example:

   ```
   jdoe@doe-desktop:~$ cat collectinfo.sh
   #!/bin/sh
   RUNCMD="ssh admin@9.30.192.108"
   $RUNCMD 'net show hostname' 'system show status'
   jdoe@doe-desktop:~$ ./collectinfo.sh
   syntax error
   ```

Results

Your public keys, along with the system status is displayed.

Disabling public key login or clearing an authorized key

For security reasons, you may want to disable public key login or clear an authorized key.

Procedure

To disable public key login or clear an authorized key, complete the following step:

```
mgmt authkeys clear
```

Results

Public key login is disabled or the authorized key is cleared.

Related reference

- Management commands

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Network commands

Network commands that you can use to manage the network configuration.

For information about how to configure multiple NICs, see Adding a second NIC.

Table 1.

Command name	Action	Syntax
net add etchost	Adds an entry to the /etc/hosts configuration file in the Management server. The IP address is typically the management IP address of the Management server. It is best practice to specify a fully qualified host name (for example, *myhost.mydept.mycompany.com*). Also, although alias is an optional parameter, it is best practice to specify an alias (for example, *myhost*). Here is an example of a **net add etchost** command. `net add etchost address 111.222.333.444 hostname myhost.mydept.mycompany.com alias myhost` If an entry was previously added with a host name that is not fully qualified, that entry must first be removed by using the **net del etchost** command before you add an entry with a fully qualified host name. Verify that the entry was successfully deleted by using the **net show etchost** command.	`net add etchost address <ip-address> hostname <fully-qualified-hostname> [alias <short-alias-name>]`
net add route address	Adds a static route to the routing table.	`net add route address <destination> mask <netmask> [gateway <gateway>] eth<n>`
net del etchost	Removes one or more entries from the /etc/hosts configuration file in the Management server. Specify the identical parameters that were used when the entry was added by using the **net add etchost** command. You can verify the details by using the **net show etchost** command.	`net del etchost address <ip-address> hostname <fully-qualified-hostname> [alias <short-alias-name>]`
net del route address	Removes a static route from the routing table.	`net del route address <ip-address> mask <mask>`
net flush dhcp	Erases cached DHCP values from the Management server.	`net flush dhcp`
net ping	Queries the specified host to determine if it is responding on the network.	`net ping {<ip-address>|<nodename>}`
net restart	Saves and applies any new network settings. You must use this command after net set, net add, or net del commands to apply the changes.	`net restart`
net restore	Restores the memory to active or backup settings.	`net restore {active|backup}`
net traceroute	Traces the network route to the specified node.	`net traceroute <ip-address>|<nodename>`

net validate	Tests the current in-memory settings and displays any warnings or errors that might prevent the network from operating correctly.
	When all network settings are acceptable, issuing this command returns the following statement, `Current network settings can be applied.`
	When an invalid setting is entered, issuing this command returns the appropriate warning or error messages. For example,
	`warning: Route to 9.43.79.4 requires static Data IP address`
	`error: Gateway 9.9.9.9 is not reachable from emgmt`
	`Current network settings cannot be applied`
		`net validate`

For the **net set** and **net show** commands, see the following topics:

- **The net set command**
  Sets the configuration for the network interfaces of the Management server, DNS server, host name, and default gateway. Choose the `net set autohost` command to automatically add the hostname/IP address to the Management server. You can also set the amount of time before reporting a lost carrier. You can execute multiple `net set` commands, then run `net show memory` to confirm your different settings. Once confirmed, you can run `net restart` for the settings to take effect.
- **The net show command**
  This command displays all network configuration for the Management server. This command can also display specific configuration information for the network interfaces, DNS server, host name, default gateway, routing tables, and socket connections. If you enable the net set autohost command, use the net show autohost to view the current settings of the autohost.

The net set command

Sets the configuration for the network interfaces of the Management server, DNS server, host name, and default gateway. Choose the `net set autohost` command to automatically add the hostname/IP address to the Management server. You can also set the amount of time before reporting a lost carrier. You can execute multiple `net set` commands, then run `net show memory` to confirm your different settings. Once confirmed, you can run `net restart` for the settings to take effect.

To improve startup time in networks that do not have a DHCP server, configure all items to either static values or none so the Management server does not search for a DHCP server. For best results, configure all your settings (such as IP address, hostname, domain, etc.) to use *either* Static or Dynamic addressing and not a mixture of both.

If you configure any value to be obtained through DHCP, startup time could be delayed as the Management server tries to locate an available DHCP server.

Table 1.

Command name	Action	Syntax
net set autohost	Allows the Management server to find its IP address from its own hostname. When you enabled this option and the Management server cannot find an IP address corresponding to its hostname, either through DNS or an entry you entered manually into the /etc/hosts directory, the Management server automatically adds an entry into the /etc/hosts directory to allow the CIOS to operate normally. Enable or disable this option through the **net set autohost** command.	`net set autohost [enable|disable ]`
net set carrier	Specifies the time, in seconds, until a missing Ethernet link carrier is reported as an error. Setting the value to zero disables the carrier checking and no errors are reported.	`net set carrier timeout <`*`timeout`*`>`

net set domain	Sets the domain name for this Management server. The effect is that most queries for names within this domain can use short names relative to the local domain.	`net set domain {dhcp eth<`*`n`*`>|static <`*`value`*`>|none}`
		Example 1:
		`net set domain dhcp eth0`
		Example 2: Set a specific domain name.
		`net set domain static mydomain.com`

net set eth<*n*>
net set interface
	Sets the IP address, Netmask and Broadcast address for the specified interface.
	Note: The V3.0.3 net set eth<n> command replaces the V3.0.2 and V3.0.1 net set interface command.
		`net set eth<`*`n`*`> {dhcp eth<`*`n`*`>|address <`*`ipaddress`*`> mask <`*`mask`*`> [bcast <`*`bcast`*`>]}I`
		Example:
		`net set eth0 address <10.20.75.169> mask <255.255.255.0>`

net set gateway	Specifies the gateway and the interface to use for that gateway. The Management server supports only one default gateway for both management and data traffic (as opposed to one gateway per interface), but additional routes can be added separately.	`net set gateway {dhcp eth<`*`n`*`>|static <`*`ipaddress`*`> eth<`*`n`*`>|none}`

Command name	Action	Syntax
net set hostname	Sets the name of the Management server. This should be the short host name, without the domain name. For example, if the fully qualified domain name of the server is **`server1.mydomain.com`**, set the host name to `server1`. V5.0.0 ONLY V5.0.1 ONLY Warning: To change the host name of a Management server that is already installed, do **not** run this command; contact IBM® support for assistance.	`net set hostname {dhcp eth<`*`n`*`>|static <`*`value`*`>}`
net set link eth<*n*>	Sets the Ethernet link to autonegotiate (the default) or to specific values. In most cases autonegotiate is preferable, but setting specific values can sometimes help improve throughput when network conditions are not optimal.	`net set link eth<`*`n`*`> {autonegotiate|speed <`*`speed`*`> duplex {half|full}}`
net set mtu eth<*n*>	Sets the Maximum Transmission Unit (MTU) size between a range of (68-9000). The default MTU is 1500.	`net set mtu eth<`*`n`*`> <`*`MTU`*`>`
net set nameserver	Sets DNS server list (maximum 2).	`net set nameserver {dhcp eth<`*`n`*`>|static <`*`value`*`>|none}`

net set ntp	Sets the time server.	`net set ntp {dhcp eth<`*`n`*`>|static <`*`value`*`>|none}`
		Example: Ask a DHCP server on the management NIC for an NTP server.
		`net set ntp dhcp eth0`

net set search	Specifies the search list. If you do not set this option, the search list is the same as the value of the domain setting.	`net set search {dhcp eth<`*`n`*`>|static <`*`value`*`>|none}`
		Example 1: Search for host names in specified domains.
		`net set search static a.mydomain.com b.mydomain.com`
		Example 2: Do not search for host names in other domains.
		`net set search none`

Related information

- Configuring to use DHCP addressing
- Configuring to use static addressing

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The net show command

This command displays all network configuration for the Management server. This command can also display specific configuration information for the network interfaces, DNS server, host name, default gateway, routing tables, and socket connections. If you enable the net set autohost command, use the net show autohost to view the current settings of the autohost.

You can issue a collection of network settings changes in memory with the **net show** command. The Management server does not persist these changes. The **net show** command only becomes permanent after you issue a net restart command.

- Active - Displays the network settings for the Management server that is currently running.
- Memory - Displays the network settings of a staging Management server that becomes active when you issue the net restart command.
- Backup - Specifies the Management server that was active before you issued the previous net restart command.

Table 1. . The following table provides descriptions for options available in the various command syntaxes:

Options	Description	Syntax
all	Displays all network configuration information for the Management server.	`net show all`
active	Displays complete network configurations.	`net show active`
autohost	Allows the Management server to find its IP address from its own hostname. When you enabled this option and the Management server cannot find an IP address corresponding to its hostname, either through DNS or an entry you entered manually into the /etc/hosts directory, the Management server automatically adds an entry into the /etc/hosts directory to allow the runtime to operate normally. Enable or disable this option through the **net set autohost** command.	`net show autohost`
backup	Displays complete network configurations.	`net show backup`
carrier	Specifies the time, in seconds, until a missing Ethernet link carrier is reported as an error. Setting the value to zero disables the carrier checking and no errors are reported.	`net show carrier [all|active|memory|backup]`
domain	Displays the domain name for this Management server. The effect is that most queries for names within this domain can use short names relative to the local domain.	`net show domain [all|active|memory|backup]` If not specified, the last parameter defaults to active.
etchost	Displays the entries within /etc/hosts.	`net show etchost`
eth<*n*>	Displays the IP address, Netmask and Broadcast address for the specified interface.	`net show eth<n> [all|active|memory|backup]`
gateway	Specifies the gateway and the interface to use for that gateway.	`net show gateway [all|active|memory|backup]` If not specified, the last parameter defaults to active.
hostname	Displays the name of the Management server. This should be the short hostname, without the domain name.	`net show hostname [all|active|memory|backup]` If not specified, the last parameter defaults to active.
interface	See eth<*n*>. Note: The V3.0.3 net show eth<n> command replaces the V3.0.2 and V3.0.1 net show interface command.	
link	Sets the Ethernet link to autonegotiate (the default) or to specific values. In most cases autonegotiate is preferable, but setting specific values can sometimes help improve throughput when network conditions are not optimal.	`net show link [all|active|memory|backup]`

Options	Description	Syntax
memory	Displays complete network configurations.	`net show memory`
mtu	Displays Maximum Transmission Unit (MTU) size for data in an IP packet.	`net show mtu [all|active|memory|backup]` If not specified, the last parameter defaults to active.
nameserver	Displays the DNS server list (maximum 2).	`net show nameserver [all|active|memory|backup]` If not specified, the last parameter defaults to active.
node	Displays the IP address for the node.	`net show node {<`*`ip address`*`>|<`*`hostname`*`> }`
ntp	Displays the network protocol for clock synchronization between a Management server and network.	`net show ntp [all|active|memory|backup]` If not specified, the last parameter defaults to active.
route	Displays active routes.	`net show route [active|memory|backup]` If not specified, the last parameter defaults to active.
search	Specifies the search list. If you do not set this option, the search list is the same as the value of the domain setting.	`net show search [all|active|memory|backup]` If not specified, the last parameter defaults to active.
sockets	Displays a list of socket addresses.	`net show sockets`
status	Displays the network status of the Management server, including the last attempt to start networking, the last time networking started, the last time networking stopped, the reason for last networking stop and the current networking activity.	`net show status`

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Network introspection commands

Network introspection (Netspect) commands allow you to manage network configurations and query the network for DHCP server, DNS server, gateway, and route information.

Note: Do not use reserved words in the user name, host name, or file name. Reserved words include: ftp, sftp, export, ibm, port, user, and file.

Table 1. Netspect commands

Command name	Action	Syntax
netspect arp	Issues an ARP request for the IP address you specify and displays the result of the ARP reply or displays the ARP cache content.	`netspect arp <lookup ipv4-address|show cache>`
netspect capture clean	Deletes all existing packet capture files.	`netspect capture clean`
netspect capture show	Displays the captured packets. Issuing this command also stops any running packet capture.	`netspect capture show`
netspect capture start	Initiates a packet trace and produces a gzip (.gz) file. Packet traces can be useful for debugging network connection errors. Only one packet trace capture can run at a time. The capture continues running until you stop it using <CTRL + C> or until a maximum number of packets is reached. Note: When you start a capture with packet size parameter set to full, `pktsize full`, fewer packets can be captured. You can transfer packet traces to another host using the **netspect export** command. Note: Capturing a packet trace overwrites the previous capture.	`netspect capture start [all]|[eth<n>]|[pktsize <header|full>]|[find <hostname|network>]` where • `hostname` - Captures to/from specified host (nn.nn.nn.nn). • `network` - Captures to/from certain network in CIDR.
netspect connect	Using the protocol you specify, this command attempts to open a connection to the network host port and displays the result of the connection attempt. If a connection is established, the connection is closed prior to this command returning.	`netspect connect host <host-id> <protocol> port <port-num>` where • `host-id` - The IP address of a network host, or the fully qualified name in the format `hostname.domain_name`. • `protocol` - The protocol to use for the connection. Must be "tcp". • `port-num` - Any valid tcp port number.
netspect dhcp eth<n>	Queries the network for DHCP servers on the network interface you specify. Also displays DHCP server supplied network configuration information, plus the offered IP address and its lease terms.	`netspect dhcp eth<n>`
netspect dns	Queries the system configured DNS server for the specified network host (i.e. perform either a forward lookup: return the IP address for a given host name, or perform a reverse lookup: return a host name for a given IP address), and display the result of the lookup.	`netspect dns lookup host <host-id>`

Command name	Action	Syntax
netspect export	Exports a packet capture file to an FTP Server. For more information about capturing packet traces, see the **`netspect capture start`** command.	**`netspect export <ftp|sftp> host <host-id> [port <number>] [user <user>] [file <filename>]`** where: - `host-id` - The name or IP address of the network host. - `port` - A valid port on the specified network host to which to connect. - `user` - Specifies the username used to log into the host. Note: When using sftp, this parameter must be specified. Anonymous logins are not allowed with sftp. - `filename` - File path on the destination host. **Example 1:** netspect export ftp. **`netspect export ftp host 123.12.12.123`** **Example 2:** netspect export sftp. **`netspect export sftp host 123.12.12.123 user myname file /home/myname/mycap.gz`**
netspect gateways	Queries the gateways and routes for the specified IP protocol family, and reports whether or not the gateway responded to the query/ping. The IP protocol family can be either ipv4, which is the default or ipv6.	**`netspect gateways <ip-protocol>`**
netspect help	Displays help text for the specified netspect command. If a command is not specified, then a help summary for all netspect commands is displayed.	**`netspect help <command>`**
netspect ifconfig eth<n>	Displays network interface configuration information for the specified interface. If a network interface name is not specified, then interface configuration information is displayed for all network interfaces.	**`netspect ifconfig eth<n>`**
netspect ping	Attempts to query the specified network host, and displays the result of the query attempt.	**`netspect ping host <host-id>`**
netspect routes	Displays currently configured routes for the specified IP protocol family. The IP protocol family can be either IPv4 (the default if none specified) or IPv6.	**`netspect routes [<ip-protocol> ]`**
netspect summary	Provides a summary of network configuration information.	**`netspect summary`**
netspect traceroute	Attempts to determine the route to the specified network host, and displays the route packets took to get to the specified network host.	**`netspect traceroute host <host-id>`**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Status commands

Status commands allow you to view the Management server status.

Table 1.

Command name	Action	Syntax

Command name	Action	Syntax
stat show analytics all	Displays summary information about the state of the Analytics cluster health, nodes, and indices. For details, see stat show analytics commands.	**stat show analytics all**
stat show analytics health	Provides a red/yellow/green status indicator for the Analytics cluster's health. For details, see stat show analytics commands.	**stat show analytics health**
stat show analytics nodes	Describes the Analytics cluster topology and provides a quick picture of performance statistics. For details, see stat show analytics commands.	**stat show analytics nodes**
stat show analytics indices	Provides an overview view of your indices and their related shards. For details, see stat show analytics commands.	**stat show analytics indices**
stat show all	Displays system status information, such as: disk usage, memory usage, uptime, and active processes.	`stat show all`
V5.0.6 + **stat show apiconfig**	V5.0.6 + Runs health tests on the Management server database, and returns the following information: Internal check summary Lists PASS/FAIL information from running Informix verification utilities. All checks must report a status of `Pass`. Memory segments Shows the amount of RAM, both as a percentage and as an aggregate amount, that the Informix engine is using on the current server. The percentage is compared to its maximum allowed memory that is provided to the virtual machine. Review this to make sure that the Informix engine has enough available RAM. Database spaces Shows how much disk space the Informix engine is using on the current server. Review this to ensure the database engine has enough available disk space. Note that only Primary servers use `tempsblobsp`, so its output is suppressed on other servers. Online server roles Lists online servers, roles, and database transaction log replication state. Database transaction log replication state is represented by a combination of the LogID and LogPos values. LogID is an indication of which transaction log is currently being used; LogPos is the most recently used position within that log. If you watch these values over time LogPos grows to a preconfigured limit, then LogID is incremented by one and LogPos is reset to zero and begins growing again. Database content changes appear first on the primary servers, and then are copied to HDR and RSS servers. In a healthy configuration, the values for LogID/LogPos on the primary server are quickly reflected on the other servers.	V5.0.6 + `stat show apiconfig [verbose]`

- **stat show analytics commands**
  Use the `stat show analytics` commands to monitor the Analytics storage cluster performance and to check the health of indices and shards.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

`stat show analytics` commands

Use the `stat show analytics` commands to monitor the Analytics storage cluster performance and to check the health of indices and shards.

There are 4 `stat show analytics` commands, which are explained in the following sections:

- `stat show analytics all` displays information from the other 3 commands to summarize the state of the health, nodes, and indices.
- `stat show analytics health` provides a red/yellow/green status indicator for the Analytics cluster's health.
- `stat show analytics nodes` describes the Analytics cluster topology and provides a quick picture of performance statistics.
- `stat show analytics indices` provides an overview view of your indices and their related shards.

stat show analytics all

The **stat show analytics all** command provides a summary of the health, nodes, and indices status for the Analytics cluster.

Sample output:

```
stat show analytics all

ANALYTICS HEALTH:
{
"cluster_name" : "apiconnect_analytics",
"status" : "green",
"timed_out" : false,
"number_of_nodes" : 2,
"number_of_data_nodes" : 2,
"active_primary_shards" : 30,
"active_shards" : 60,
"relocating_shards" : 0,
"initializing_shards" : 0,
"unassigned_shards" : 0,
"delayed_unassigned_shards" : 0,
"number_of_pending_tasks" : 0,
"number_of_in_flight_fetch" : 0,
"task_max_waiting_in_queue_millis" : 0,
"active_shards_percent_as_number" : 100.0
}

ANALYTICS NODES:
ip heap.percent ram.percent cpu load_1m load_5m load_15m node.role mast er name
10.100.7.18 49 95 10 0.01 0.04 0.00 mdi - 1wFRdEy
10.100.7.10 39 91 9 0.08 0.05 0.01 mdi * pRcb9np

ANALYTICS INDICES:
health status index uuid pri r ep docs.count docs.deleted store.size pri.store.size
green open gw-5de8532de4b083e85eda1108-2020.12 uvbjgW8dRN-UJV17Fknj1Q 1 1 8905 20 3.7mb 1.9mb
green open cmc-2021.01 reGpvAAaQFKLbvwrPe-DAw 1 1 535742 0 287.9mb 143.9mb
green open gw-5de85467e4b083e85eda110e-2021.03 4LcHhtFQTJuNMR0vIRtOlQ 1 1 220 0 456kb 228kb
green open apim-5b7c9a23e4b0464ff9917d30-2020.12 948LuYMOQwerhzS7ezzrcw 1 1 68 0 495.5kb 327.9kb
green open gw-5de85467e4b083e85eda110e-2021.01 adO-Mw5yQV-DoeXmo8pBkg 1 1 8928 0 3.6mb 1.8mb
green open gw-5e90a8d6e4b02066184cb465-2021.02 HyLy8YJtQBKgtdCLDlar2A 1 1 8062 0 3.9mb 1.9mb
green open gw-5e90a8d6e4b02066184cb465-2021.01 tlKTaTE7ROqkCbfLmoUjLg 1 1 8927 0 4.5mb 2.2mb
green open apim-default y56dVS7hTlOYXoVfTiYZMA 1 1 1 0 110.1kb 55kb
green open .kibana 1omwR24gQcuEZtNOc4svew 5 1 35 1 251.5kb 125.7kb
green open gw-5e90a8d6e4b02066184cb465-2021.03 oppBPVdXQJqemb6tE_k8FQ 1 1 220 0 712.4kb 356.2kb
green open gw-5de85467e4b083e85eda110e-2020.12 WxosDxaPRICjYOZM3tpE_A 1 :
health status index uuid pri rep docs.count docs.deleted store.size pri.store.size
green open gw-5de8532de4b083e85eda1108-2020.12 uvbjgW8dRN-UJV17Fknj1Q 1 1 8905 20 3.7mb 1.9mb
green open cmc-2021.01 reGpvAAaQFKLbvwrPe-DAw 1 1 535742 0 287.9mb 143.9mb
green open gw-5de85467e4b083e85eda110e-2021.03 4LcHhtFQTJuNMR0vIRtOlQ 1 1 220 0 456kb 228kb
green open apim-5b7c9a23e4b0464ff9917d30-2020.12 948LuYMOQwerhzS7ezzrcw 1 1 68 0 495.5kb 327.9kb
green open gw-5de85467e4b083e85eda110e-2021.01 adO-Mw5yQV-DoeXmo8pBkg 1 1 8928 0 3.6mb 1.8mb
green open gw-5e90a8d6e4b02066184cb465-2021.02 HyLy8YJtQBKgtdCLDlar2A 1 1 8062 0 3.9mb 1.9mb
green open gw-5e90a8d6e4b02066184cb465-2021.01 tlKTaTE7ROqkCbfLmoUjLg 1 1 8927 0 4.5mb 2.2mb
green open apim-default y56dVS7hTlOYXoVfTiYZMA 1 1 1 0 110.1kb 55kb
green open .kibana 1omwR24gQcuEZtNOc4svew 5 1 35 1 251.5kb 125.7kb
green open gw-5e90a8d6e4b02066184cb465-2021.03 oppBPVdXQJqemb6tE_k8FQ 1 1 220 0 712.4kb 356.2kb
green open gw-5de85467e4b083e85eda110e-2020.12 WxosDxaPRICjYOZM3tpE_A 1 1 8906 21 3.6mb 1.8mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.03 iVkxD4AESveA5Lv2EDyJTA 1 1 9 0 285.5kb 133kb
green open gw-5de8532de4b083e85eda1108-2021.02 6c7OXh1JTp29FVPmx0BBsQ 1 1 8064 0 3.5mb 1.7mb
green open cmc-2021.03 EFRMOm8oQ-SArKTb9xfxeg 1 1 13221 0 7mb 3.5mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.01 rscXI5fgRhSAG51qAnB6AA 1 1 8 0 251.2kb 116.2kb
green open gw-5ef8d3b0e4b0217ee4f66995-2020.12 SD09yKsmQxaNCPLOs_uKpw 1 1 8905 19 4.4mb 2.2mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.02 lsmmDs-RR2qT7aBEigYKvQ 1 1 376 0 1.2mb 666.4kb
green open cmc-2021.02 BGGgFuJ8SRiTnMLdVXWS3g 1 1 370379 0 193.5mb 96.7mb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.03 m8sPyG13QUObYgWlYmwosg 1 1 220 0 289.2kb 144.6kb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.01 wglxskB2SReSqHNlTI1Xgg 1 1 8924 0 4.4mb 2.2mb
green open gw-5e90a8d6e4b02066184cb465-2020.12 lseFmnHiTsmyGarnEm3u5A 1 1 8904 20 4.3mb 2.2mb
green open gw-5de8532de4b083e85eda1108-2021.03 DSezPyliSrSWiHzdpXlXeg 1 1 220 0 618kb 317.2kb
green open cmc-2020.12 q6pDD8FDRha9I3S9vq7o3g 1 1 534605 1080 288.1mb 145mb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.02 ytY_YCoeRIa0k54aQZgslQ 1 1 8062 0 3.9mb 1.9mb
green open gw-5de85467e4b083e85eda110e-2021.02 YipQvmZcSq6McGR6cx85xg 1 1 8064 0 3.3mb 1.6mb
green open gw-5de8532de4b083e85eda1108-2021.01 N-za6uOeRXSdKZXbltfcrQ 1 1 8927 0 3.8mb 1.9mb
```

For more information on the different types of status information included in the results, continue to the following sections and read about the health, nodes, and indices information.

stat show analytics health

The **stat show analytics health** command provides a red/yellow/green status indicator for the Analytics cluster's health, based on the health of the shards (where data is stored) and indices (which track the stored data).

Elasticsearch documentation: Cluster health

Sample output:

```
stat show analytics health
{
"cluster_name" : "apiconnect_analytics",
"status" : "green",
"timed_out" : false,
"number_of_nodes" : 2,
"number_of_data_nodes" : 2,
"active_primary_shards" : 30,
"active_shards" : 60,
"relocating_shards" : 0,
"initializing_shards" : 0,
"unassigned_shards" : 0,
"delayed_unassigned_shards" : 0,
"number_of_pending_tasks" : 0,
"number_of_in_flight_fetch" : 0,
"task_max_waiting_in_queue_millis" : 0,
"active_shards_percent_as_number" : 100.0
}
```

The cluster health status is: **green**, **yellow** or **red**. On the shard level, a **red** status indicates that the specific shard is not allocated in the cluster, **yellow** means that the primary shard is allocated but its replicas are not (each shard can have 0 or more replicas distributed among the nodes to provide failover), and **green** means that all shards are allocated. The index level status is determined by the worst shard status, and the overall cluster status is determined by the worst index status.

For more information on indices and shards, see the Elasticsearch glossary. There is also a discussion of shards and indices on Stack Overflow that you might find helpful.

stat show analytics nodes

The **stat show analytics nodes** command describes your cluster topology and provides a quick picture of performance statistics.

Elasticsearch documentation: cat nodes

Sample output:

```
stat show analytics nodes
ip heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
10.100.7.18 49 95 11 0.00 0.02 0.00 mdi - 1wFRdEy
10.100.7.10 40 91 9 0.00 0.01 0.00 mdi * pRcb9np
```

The response includes the following information about the Analytics nodes:

- IP - The IP address of the node
- heap.percent - Used heap percentage
- ram.percent - Used RAM percentage
- cpu - Recent system CPU usage as percent
- load_1m - Most recent load average for last 1 minute
- load_5m - Load average for last 5 minutes
- load_15m - Load average for last 15 minutes
- node.role - Permissions for the node Master eligible node (m); Data node (d); Ingest node (i); Coordinating node only (-)
- master - Elected master (*); Not elected master (-)
- name - Node name

stat show analytics indices

The **stat show analytics indices** command provides an overview view of your indices and their related shards, with information on the health of each index, as well as how many shards make up an index, the number of docs at the Lucene level, deleted docs, primary store size, and total store size (all shards including replicas). The information spans nodes.

Elasticsearch documentation: cat indices

Sample output:

```
stat show analytics indices
health status index uuid pri rep docs.count docs.deleted store.size pri.store.size
green open gw-5de8532de4b083e85eda1108-2020.12 uvbjgW8dRN-UJV17Fknj1Q 1 1 8905 20 3.7mb 1.9mb
green open cmc-2021.01 reGpvAAaQFKLbvwrPe-DAw 1 1 535742 0 287.9mb 143.9mb
green open gw-5de85467e4b083e85eda110e-2021.03 4LcHhtFQTJuNMR0vIRtOlQ 1 1 225 0 619.6kb 309.8kb
green open apim-5b7c9a23e4b0464ff9917d30-2020.12 948LuYMOQwerhzS7ezzrcw 1 1 68 0 495.5kb 327.9kb
green open gw-5de85467e4b083e85eda110e-2021.01 adO-Mw5yQV-DoeXmo8pBkg 1 1 8928 0 3.6mb 1.8mb
green open gw-5e90a8d6e4b02066184cb465-2021.02 HyLy8YJtQBKgtdCLDlar2A 1 1 8062 0 3.9mb 1.9mb
green open gw-5e90a8d6e4b02066184cb465-2021.01 tlKTaTE7ROqkCbfLmoUjLg 1 1 8927 0 4.5mb 2.2mb
green open apim-default y56dVS7hTlOYXoVfTiYZMA 1 1 1 0 110.1kb 55kb
green open .kibana 1omwR24gQcuEZtNOc4svew 5 1 35 1 251.5kb 125.7kb
green open gw-5e90a8d6e4b02066184cb465-2021.03 oppBPVdXQJqemb6tE_k8FQ 1 1 225 0 876.4kb 438.2kb
green open gw-5de85467e4b083e85eda110e-2020.12 WxosDxaPRICjYOZM3tpE_A 1 1 8906 21 3.6mb 1.8mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.03 iVkxD4AESveA5Lv2EDyJTA 1 1 9 0 285.5kb 133kb
green open gw-5de8532de4b083e85eda1108-2021.02 6c7OXh1JTp29FVPmx0BBsQ 1 1 8064 0 3.5mb 1.7mb
green open cmc-2021.03 EFRMOm8oQ-SArKTb9xfxeg 1 1 13516 0 7.1mb 3.5mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.01 rscXI5fgRhSAG51qAnB6AA 1 1 8 0 251.2kb 116.2kb
green open gw-5ef8d3b0e4b0217ee4f66995-2020.12 SD09yKsmQxaNCPLOs_uKpw 1 1 8905 19 4.4mb 2.2mb
green open apim-5b7c9a23e4b0464ff9917d30-2021.02 lsmmDs-RR2qT7aBEigYKvQ 1 1 376 0 1.2mb 666.4kb
green open cmc-2021.02 BGGgFuJ8SRiTnMLdVXWS3g 1 1 370379 0 193.5mb 96.7mb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.03 m8sPyG13QUObYgWlYmwosg 1 1 225 0 453.2kb 226.6kb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.01 wglxskB2SReSqHNlTI1Xgg 1 1 8924 0 4.4mb 2.2mb
green open gw-5e90a8d6e4b02066184cb465-2020.12 lseFmnHiTsmyGarnEm3u5A 1 1 8904 20 4.3mb 2.2mb
green open gw-5de8532de4b083e85eda1108-2021.03 DSezPyliSrSWiHzdpXlXeg 1 1 225 0 781.6kb 399kb
green open cmc-2020.12 q6pDD8FDRha9I3S9vq7o3g 1 1 534605 1080 288.1mb 145mb
green open gw-5ef8d3b0e4b0217ee4f66995-2021.02 ytY_YCoeRIa0k54aQZgslQ 1 1 8062 0 3.9mb 1.9mb
green open gw-5de85467e4b083e85eda110e-2021.02 YipQvmZcSq6McGR6cx85xg 1 1 8064 0 3.3mb 1.6mb
green open gw-5de8532de4b083e85eda1108-2021.01 N-za6uOeRXSdKZXbltfcrQ 1 1 8927 0 3.8mb 1.9mb
```

An important result to note is the index health. The index level health status is determined by the worst shard status related to that index. A red index means that at least one primary shard (and all of its replicas) containing data tracked by the index is missing. This means that you are missing data: searches will return partial results, and indexing into that shard will return an exception.

On the shard level, a red status indicates that the specific shard is not allocated in the cluster, yellow means that the primary shard is allocated but its replicas are not, and green means that all shards are allocated.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

System commands

System commands allow you to manage the operation of the Management server, which includes managing licenses for the Management server and connectors.

Note: Do not use reserved words in the user name, host name, or file name. Reserved words include: ftp, sftp, ibm, port, user, and file.

Table 1.

Command name	Action	Syntax
system autoreboot	Allows you to enable or disable automatic reboot for a Management server. Note: During a Management server firmware upgrade, the Management server reboots even if the automatic reboot is disabled.	`system autoreboot <on | off>`
system autoreboot setting	Use the setting parameter to print autoreboot settings.	`system autoreboot setting`

Command name	Action	Syntax

system clean	Clears the system states of the Management server. When you issue a system clean all command, the Management server may reboot.
	Running the `all` option results in the removal of all data and configurations. The results are equivalent to restoring factory default settings for the appliance. After the reboot, you will be asked to accept the license agreement before logging in to the cloud console for the appliance.
	V5.0.8 + You can remove all of the analytics information by entering the `analytics` option. When you run this command with the `analytics` option, it restarts the node to commit the changes.
	Remember: You must run the **system clean analytics** command on all of the nodes of a multiple-node cluster to permanently remove the information. If you only run it on one node, then the analytics information is copied back to the server during its next synchronization.
	V5.0.8 + For an environment with multiple management server nodes, complete these steps:
	1. On each of the management server non-primary nodes, stop the system by entering the following command:
	`debug system stop`
	Note: While the primary management server is restarted, the other management server non-primary nodes also lose connectivity.
	2. On the management server primary node, enter the following command:
	`system clean analytics`
	The node automatically restarts.
	3. After the management server primary node finishes restarting, enter the following command on each of the management server non-primary nodes:
	`system clean analytics`
	Each of the nodes automatically restarts.
	V5.0.8 + You can also remove the contents of the server logs by entering the `logs` option.
		`system clean all|apiconfig|analytics|logs`
		where:
		• `all` - Resets the appliance to factory default settings.
		• `apiconfig` - Removes all the topology, organizations, Products, analytics data, and API configuration from the Management server.
		• V5.0.8 + `analytics` - Removes all of the analytics information from the system.
		• V5.0.8 + `logs` - Removes the log information from the system, which can take a large amount of space. You can select from the following options:
		◦ Export log: This clears all of the log records that are created when you export your analytics data.
		◦ Elasticsearch: This clears all of the log records that are created by Elasticsearch actions.
		◦ No: This cancels the log clean command, and does not remove any log information.
		• `sessions` - Removes Web UI session data.

system persist	Enables or disables the disk cache on the Management server. The default and recommended setting is `system persist on`. When persistence is enabled, writes are synchronized to ensure that all requested data is completely written to disk. This behavior helps protect data integrity, particularly when accessing external transactional systems such as databases. When persistence is disabled, on an appliance that supports this feature, performance might be improved. However, there is a risk of losing job progress state.	`system persist { on | off }`
system persist setting	Displays the current system persist setting: on or off. Use the system persist command to specify whether the disk cache is enabled or disabled.	`system persist setting`
system poweroff	Shuts down all routing services and then powers off the Management server. If the force parameter is used, power is cut off immediately.	`system poweroff`
system show publickey	Authorize transfers from this API Connect server without providing a password.	`system show publickey`
system reboot	Shuts down all routing services and then reboots the Management server. Note: When the system reboots, any network settings that have not been saved are lost. To commit network settings to the Management server, use the **net restart** command.	`system reboot`
system restart	Restarts the runtime and cloud console, without affecting the network connectivity.	`system restart`
system show platform	Displays information about the Management server, such as: ROM version, appliance platform, serial number, and MAC addresses.	`system show platform`

Command name	Action	Syntax

system show status	Available for Standalone Management servers as well as the Active Management server in an HA pair. Shows whether the following components are up or down: System, Network, and Runtime.
	Note:
	• V5.0.1+ The Runtime status returned by the **system show status** command indicates whether the server processes are running correctly and are available for use.
	• V5.0.0 ONLY The Runtime status returned by the **system show status** command indicates only whether the server processes have been started; the **ha list** command confirms whether they are running correctly and are available for use.
		`system show status`
		Example output:
		`Appliance Status`
		`----------------`
		`    System: Up`
		`   Network: Up`
		`   Runtime: Up`

system show version	Displays the version of the operating system on the Management server.	`system show version`

system update firmware	Updates the Operating System on the Management server by using an image from a file; you use command parameters to specify the location of the file, and the mechanism to be used to obtain the file.
	You can obtain the image file by using any of the following mechanisms:
	• FTP
	• SFTP
	• HTTP
	• HTTPS
	• URL
		`system update firmware from ftp <hostname_or_ip_address> [port <number>] [user <user>] file <filepath>`
		`system update firmware from sftp <hostname_or_ip_address> [port <number>] user <user> file <filepath>`
		`system update firmware from http <hostname_or_ip_address> [port <number>] [user <user>] file <filepath>`
		`system update firmware from https <hostname_or_ip_address> [port <number>] user <user> file <filepath>`
		`system update firmware from url <fileurl> [user <user>]`
		Note: When using sftp, you must specify the `user` parameter. Anonymous logins are not allowed with sftp.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Time commands

Time commands allow you to set or synchronize the date and time on the Management server.

Note: To avoid time drift on Virtual Appliances, run NTP on the host and guest. Running NTP sets the system time to UTC. Do not use the **time set clock** command to reset time on a Virtual Appliance.

Table 1.

Command name	Action	Syntax

Command name	Action	Syntax
time set	Sets individual time and date components relative to the time zone you choose. The Management server restarts after you issue this command using the `clock` parameter. If you do not know the syntax for a specific time zone, choose the last option and do not specify a value for zone.	`time set { { | clock<string> } | { zone<zone>} | { zone} }` Example 1: - time set clock<*string*> - User input: `time set clock Sat Jan 1 00:00:00 2005` Example 2: - time set zone<*zone*> - User input: `time set zone America/Los_Angeles`
time show	Display all time settings or individual time components (the current time, date, or time zone).	`time show`

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

High availability commands

Use high availability commands to monitor the status of all Management servers in your IBM® API Connect cloud.

Table 1.

Command name	Action	Syntax

Command name	Action	Syntax
ha list	Display the network, database, and runtime status for each management server in your IBM API Connect cloud. The following information is displayed for each server: Self Displays the value Yes or No to indicate which server the command was issued from. IP The server IP address. HA Displays the value Yes or No to indicate whether the server can participate in high availability operations, and therefore whether it is able to be the Primary server and make decisions about which server should be the Primary server. A server that does not participate in high availability operations still receives database updates, and allows for load distribution. Change this setting by using the **ha set host** command. Network The network status. Displays the value Up or Down. Role The database role, which can take the following values: - Primary: the only server in the cloud that is responsible for writing to the database. - Active Arbitrator: the connection manager that decides which server to promote to the Primary server if the current Primary server becomes unavailable. - HDR: the high availability server that is constantly synced with the Primary server. API Connect does not use HDR outside of the main site feature. - RSS: a server that can be read from, and which forwards database write operations to the Primary server. Runtime The status of all applications on the server; for example, Cloud Manager and Developer Portal. Displays the value Up or Down, indicating whether the server processes are running correctly and are available for use. FQDN The fully qualified domain name of the server.	`ha list`
ha list [timeout] [json]	Optional flags to define the maximum length of time you will wait for a server to respond before there is a timeout, and to convert your output into JSON.	`ha list [timeout] [json]`
ha make primary [host <host>] [force]	Converts given server into Primary. Defaults to the local server. Note: If cloud's current Primary is unavailable, this may result in data loss.	`ha make primary [host <host>] [force]`
ha set host <host> <enabled|disabled>	Configures server in cloud to allow/disallow it to become Primary or Active Arbitrator (AA).	`ha set host <host> <enabled|disabled>`
ha set default <enabled|disabled>	The high availability setting to use for new servers added to this cloud.	`ha set default <enabled|disabled>`
ha show <default|host <host>>	Output the current high availability setting of the cloud or given server.	`ha show <default|host <host>>`

Example output

```
Self  IP        HA   Network     Role      Runtime  FQDN
Yes   1.2.3.4   Yes  Up          RSS/AA    Up       myserver1.com
No    1.2.3.5   Yes  Up          Primary   Up       myserver2.com
No    1.2.3.6   No   Up          RSS       Up       myserver3.com
```

Related concepts

- Define the main site in your API Connect cloud

Related information

- Dissociation and your cloud
- Configuration database failover timeout

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Supported key algorithms, ciphers, and MACs

API Connect supports a range of key algorithms, ciphers, and MACs when connecting to or from the management server.

When connecting to or from the management server by using SSH or SFTP, the following key algorithms, ciphers, and MACs are supported:

- Key algorithms:
    - ecdh-sha2-nistp256
    - ecdh-sha2-nistp384
    - ecdh-sha2-nistp521
    - diffie-hellman-group-exchange-sha256
- Ciphers:
    - aes128-ctr
    - aes192-ctr
    - aes256-ctr
    - aes128-cbc
    - 3des-cbc
    - cast128-cbc
    - aes192-cbc
    - aes256-cbc
- MACs:
    - hmac-sha1
    - umac-64@openssh.com
    - hmac-sha2-256
    - hmac-sha2-512

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

IBM API Connect best practices

Consider implementing the following best practices for optimizing your IBM® API Connect cloud.

- Define two Management servers to ensure High Availability (HA) and failover.
- Upgrade all defined servers at the same time to keep the versions synchronized.
  To determine the version on a virtual appliance, complete these steps:
    1. Log on to the CLI on the appliance through a Secure Shell (SSH).
    2. Enter the following command:

       ```
       system show version
       ```

  To determine the version on a physical appliance, refer to the instructions provided with the appliance.

  Ensure that the minimum software and hardware requirements are installed. For more information, see IBM API Connect Version 5.0 requirements.
- Take a backup of the Management server periodically to ensure that you have a backup of the IBM API Connect configuration information. For more information, see Preserve your cloud data.
- When you create a custom dashboard for viewing analytics data, or when you update an existing dashboard, export the definition of the dashboard to retain a backup copy. It is possible for the dashboard to become inaccessible if the Elasticsearch index that stores

the definition becomes corrupted. See Exporting dashboards for the procedure.

- Check the health data of your Management servers with REST APIs. See Obtain health check data of Management servers by using REST API calls for more information.
- V5.0.6 + Test the health of your Management servers and database regularly by running the `stat show apiconfig` command in the Command Line Interface. This indicates whether your database is healthy. For more information, see Testing Management servers.

Related information

For Developer Portal best practices, see Developer Portal best practices for administrators.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect V5 glossary

The glossary of API Connect V5 terms and definitions.

API Developer
An API Developer creates and configures APIs, Products, and policies for provider organizations of 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.

API operation
A unit of a REST API that can be invoked. An API operation comprises an HTTP verb and a URL path that is subordinate to the context root of the API.

application
A piece of client code that accesses APIs to interact with a service, system, or content. Applications are typically web applications or mobile apps.

assembly
Application programming interface that provides rich functionality for interacting with an application: makes side calls to external services and then transforms and aggregates the response before a response is relayed to the calling application.

Catalog
A Catalog is a staging target, and 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 that is 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 application secret with their application key. The application secret functions effectively as a password known only to the application. The application secret is passed by the client by using an HTTP query parameter.

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

community
A collection of developer 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.

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

Management server

A Management server stores all of the cloud configuration, and controls communication between the other servers within API Connect.

Management service

The Management service consists of one or more Management servers.

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 developer organization owns only applications. An organization has at least one owner. An organization can be a project team, department, or division.

Path

A 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 available a collection of operations from one or more APIs, 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 policy is a piece of 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

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. You can create Plans only within Products, and these Products are then published in a Catalog.

proxy

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

role

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

V5.0.2 and earlier Sandbox Catalog

V5.0.2 and earlier In a Sandbox Catalog, approvals are bypassed for publishing and lifecycle actions. Pending approvals are canceled when a non-Sandbox Catalog is converted to a Sandbox. A Sandbox Catalog is used for testing APIs that are under development.

V5.0.3+ Development Catalog

V5.0.3+ In a development Catalog, approvals are bypassed for publishing and lifecycle actions. Pending approvals are canceled when a non-development Catalog is converted to a development Catalog. A development Catalog is used for testing APIs that are under development.

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.

server

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

service

A service is a server process that is an entity in a network of servers.

V5.0.5+ Space

V5.0.5+ A Space is 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 Profile

An SSL 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

A subscription is 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.

user registry

A user registry is a way of securing access to Catalogs and APIs. The user can protect APIs with a user registry so that user credentials must be supplied when an API is called.

vendor extension

A vendor extension is added to a REST API to extend the OpenAPI (Swagger 2.0) specification.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Accessibility features for 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 API Connect V5. 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Legal information

Notices, and terms and conditions for information centers.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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® 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 http://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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

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

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/SSMNED_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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.
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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Planning, installing, and upgrading IBM API Connect

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.

In addition to installing your IBM API Connect cloud, you can upgrade any existing installations of IBM API Management Version 4.0 or IBM API Connect Version 5.0 to the latest version of IBM API Connect.

- **IBM API Connect Version 5.0 requirements**
  Ensure that you install the minimum API Connect operating system requirements.
- **Planning your cloud**
  When you install IBM API Connect, you must define an on-premises cloud. To determine the topology of appliances for this cloud consider the number of Management and Gateway servers that are required to address your API usage.
- **Installing IBM API Connect**
  IBM API Connect can be installed by using VMware, or Citrix XenServer.
- **Upgrading your API Connect cloud**
  You can upgrade your API Connect on-premises cloud to take advantage of the most recent fixes and minor enhancements.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

IBM API Connect Version 5.0 requirements

Ensure that you install the minimum API Connect operating system requirements.

Hardware requirements

Check for the most recent hardware requirements for IBM® API Connect Version 5.0. For more information, see the Hardware section of the IBM API Connect Version 5.0 Detailed System Requirements report on the IBM Software Product Compatibility Reports site according to which API Connect offering you are using:

(the following links display the Detailed System Requirements report for the latest version. If you want to see the report for an earlier version, open the Detailed system requirements for a specific product page, search for the IBM API Connect product, then select the required offering and version)

- Essentials
- Professional
- Enterprise

For a description of the API Connect offerings, see API Connect offerings.

Software requirements

Check for the most recent software requirements for IBM API Connect Version 5.0. For more information, see the Supported Software section of the IBM API Connect Version 5.0 Detailed System Requirements report on the IBM Software Product Compatibility Reports site according to which API Connect offering you are using:

(the following links display the Detailed System Requirements report for the latest version. If you want to see the report for an earlier version, open the Detailed system requirements for a specific product page, search for the IBM API Connect product, then select the required offering and version)

- Essentials
- Professional
- Enterprise

For a description of the API Connect offerings, see API Connect offerings.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Planning your cloud

When you install IBM® API Connect, you must define an on-premises cloud. To determine the topology of appliances for this cloud consider the number of Management and Gateway servers that are required to address your API usage.

Ensure that you have the minimum hardware and software requirements installed. For more information, see IBM API Connect Version 5.0 requirements.

Note: If you are creating a cluster environment with multiple servers, you must have an NTP server. For more information on configuring an NTP server, see Configuring an NTP server.
At least one Management server and one Gateway server are required to create a cloud capable of running the API Connect solution. For a description about the specific type of servers, see Firewall requirements.
The API Connect cloud can include a mixture of different physical and virtual appliances. You can also choose to define an environment made up of only virtual appliances. The following table identifies the supported form-factors for each server.

Table 1. Supported form factors for IBM API Connect servers

	Management server	**Gateway server**
Virtual Appliance	Supported	Supported
Physical Appliance	Not applicable	Supported

The minimum server configuration of one Management server and one Gateway server can support a basic test environment. However, if you are planning a production environment, then you might want to build in some resilience to avoid a single point of failure. Regardless of what you initially define, you can add extra servers to your cloud in the future.

To help with your topology planning, consider the questions in the following table to determine your API Connect infrastructure.

Type of server?	**Number of servers? What is your expected API usage?**	**Physical instance?**	**Virtual instance?**
Management	Minimum of 1	Not applicable	A Management server is always virtual
Gateway	Minimum of 1	Do you have an IBM WebSphere® DataPower® Integration Appliance that you can use?	You can use a virtual appliance. Note: If you use a mixture of physical and virtual appliances for the Gateway servers, the load balancing configuration might not use the full capabilities of the physical appliance.

For detailed guidance on deploying IBM API Connect in various environments and topologies, see the API Connect whitepaper.

- **IBM API Connect topology**
  Depending on what you want to use your API Connect cloud for, consider the topology that you want to implement.
- **Firewall requirements**
  Consider the port configuration that is required between the Gateway (DataPower) servers and the Management servers for an IBM API Connect cloud.
- **Adding a second NIC**
  The Management server can support two NICs.

- **Load balancing in IBM API Connect**
  In the IBM API Connect cloud, multiple Management servers and multiple Gateway servers can be used to achieve high availability or resilience. If multiple servers are used in a cloud, consider the following aspects of load balancing: API calls, the user interfaces, and communications between servers in the API Connect cloud.
- **Configuring Catalogs and testing the Developer Portal**
  As an optional postrequisite to the installation of the Developer Portal, you can configure Catalogs and test that configuration before using the Developer Portal.
- **High availability for the Developer Portal**
  High Availability can be achieved by clustering Developer Portal sites, so that Developer Portal users will always have access to the site.
- **Enabling database encryption for the Developer Portal**
  You can enable encryption for the communication data for databases between nodes in your Developer Portal cluster.

Related concepts

- Firewall requirements

Related tasks

- Installing IBM API Connect

Related reference

- IBM API Connect Version 5.0 requirements

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

IBM API Connect topology

Depending on what you want to use your API Connect cloud for, consider the topology that you want to implement.

The minimum server configuration that you can have in API Connect is one Management server and one Gateway server. However, this configuration does not provide any high availability.

By configuring two Management servers and two Gateway servers, your cloud benefits from high availability (HA). When you define more than one Management server, configure an external load balancer. Then, if one server is removed for planned maintenance or an unplanned failure, your users can continue to access the cloud. The more servers that you add the greater the ability of your cloud to withstand outages and continue to provide processing capability.

Consider whether you want to define a separate environment, within your cloud, for development work, functional testing, or staging. Separating your cloud usage can help to prevent problems when you are testing new APIs before you make them available to application developers.

Although you can populate your cloud with only virtual servers, you might want to use physical DataPower® appliances for your production Gateway servers. The physical Gateway servers provide improved performance throughput when compared with virtual Gateway servers.

Before you decide how many Management servers to add to your API Connect cloud, consider how the data is handled. Management servers control two different types of data:

- Configuration (for example, Users, APIs, Products)
- Analytics (for example, API usage, performance)

Configuration data is tightly controlled. At any time, only one Management server (the Primary) is allowed to write the data. The other servers (Secondary or RSS) maintain a complete local cache for performance and high availability (HA) reasons. Significant performance improvements are achieved by balancing connections across the Management servers.

When planning for high availability, consider the following:

- Defining the main site in your API Connect cloud
- Configuration database failover timeout

- Dissociation and your cloud

Analytics data is handled differently. All of the data that is written to the Analytics system is replicated asynchronously on the back-end server. When Analytics data is sent from a Gateway server to a Management server, that data is balanced across the available Management servers. If there are two or more Management servers in the cloud, at least two of the Management servers hold each Analytics data record for redundancy purposes. Therefore, in this scenario, a minimum of two copies are kept for each data record.

When you configure a Management server, you can define a second hard disk to store the analytics data that is captured for your cloud. The size of the hard disk can vary from the advised minimum of 350 GB for a production cloud to a maximum of 2 TB. It is important to consider the disk size that you need to hold the analytics data. The second hard disk size can be modified in the settings of the deployed VM after deployment but before it is powered on for the first time.

Restriction: The maximum supported combined size of all disks is 2 TB per Management server.
Restriction: If you define two Management servers that both use a second hard disk, you do not have twice the disk space for the analytics data. With two Management servers, the analytics data is synchronized across the disks to provide high availability.
As part of your topology planning, consider the server configuration that you want to implement. For more information, see Firewall requirements.

Another consideration, is the IP address that you need to configure for the Gateway server.
Important: The actual IP address is determined from the interface definition on the DataPower appliance, and is typically accessible on the Internet. This address is also generated into the front side protocol handler of the API Connect service that mediates the client messages. If this DataPower appliance is part of a clustered configuration, this Ethernet interface is used for configuring self-balancing and the virtual IP address. This interface might be the same as the one used for the XML Management interface. If the server is added to a cluster that uses DataPower self-balancing, a Standby Control configuration is added automatically.

Related concepts

- Firewall requirements

Related tasks

- Installing IBM API Connect

Related reference

- IBM API Connect Version 5.0 requirements

Related information

- Dissociation and your cloud
- Define the main site in your API Connect cloud
- Configuration database failover timeout

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Firewall requirements

Consider the port configuration that is required between the Gateway (DataPower®) servers and the Management servers for an IBM® API Connect cloud.

Required Ports between zones

The following example of a network diagram 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. *[You can hover over the numbers in the diagram to see the default port numbers, which are also shown in Table 1.]*

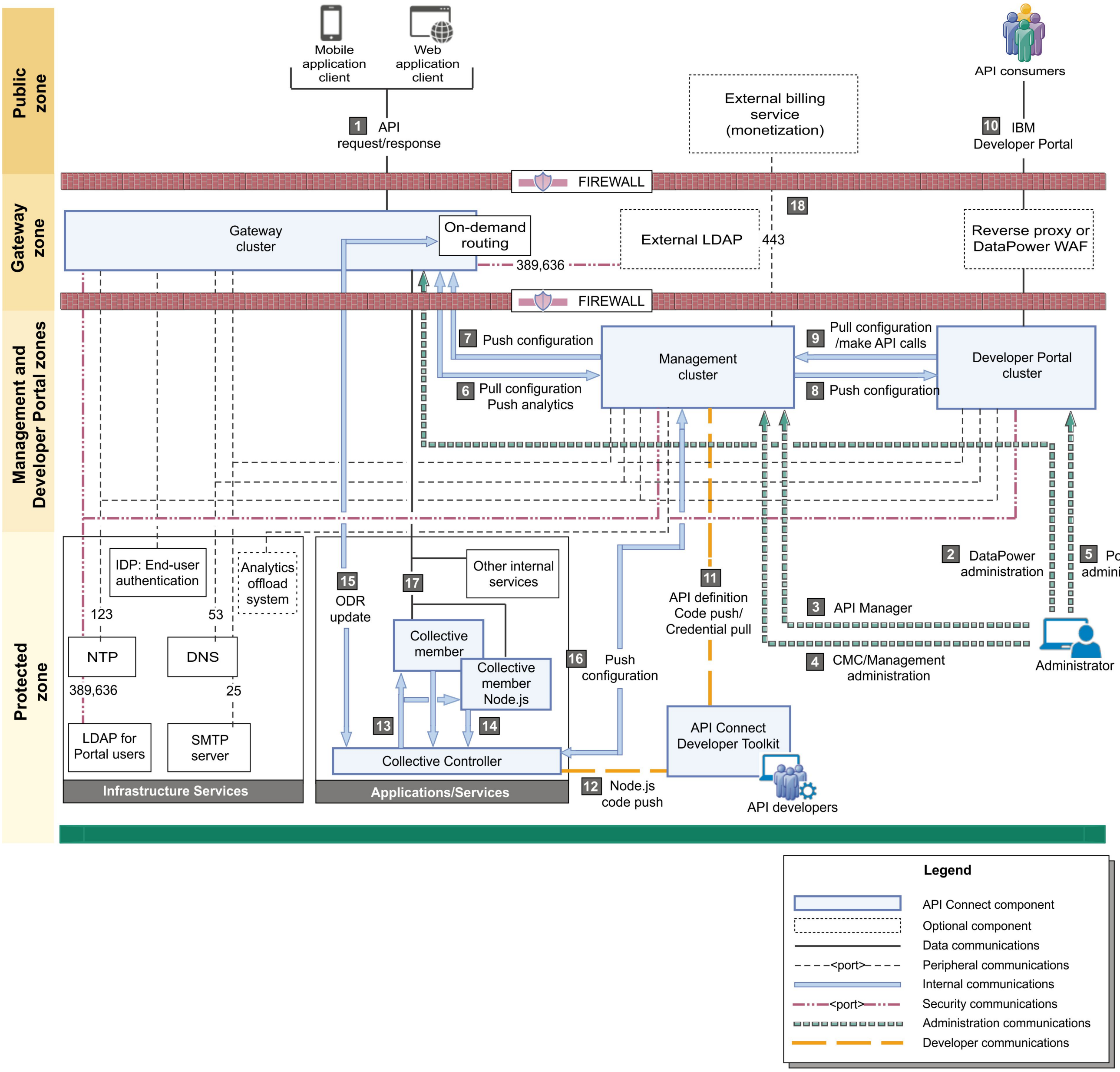


Table 1. Key for the network diagram example

	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	CMC/Management administration – Internal operators who are managing the Management servers.	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	Pull configuration, Push analytics – Gateway servers communicate with Management servers.	2443, 9443 HTTPS from Gateway servers to Management servers.
7	Push configuration – Management servers communicate with Gateway servers.	443, 5550 HTTPS from Management servers to Gateway servers.

	Usage description	Default port number
8	Push configuration/webhooks – Management servers push configuration and webhooks to the Developer Portal.	22 SSH Management servers to Developer Portal servers. 2443 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.
11	Push API definition to Management server. Pick up credential for microservice code push.	443 HTTPS from Protected zone to Management zone.
12	Push microservice Node.js project to Collective Controller.	22 SSH, 9443 HTTPS within Protected zone (Applications subzone).
13	Application deployment and management operations. Controller communicates with Member servers.	22 SSH, <admin port greater than 9440> HTTPS within Protected zone (Applications subzone).
14	Internal communication. Member server to Collective Controller.	9443 HTTPS within Protected zone (Applications subzone).
15	On-demand routing update over long-lived connection. Gateway servers communicate with the Collective Controller.	9443 HTTPS from Gateway zone to Protected zone (Applications subzone).
16	Internal communication. API Management to Collective Controller.	9443 HTTPS from Management zone to Protected zone.
17	Application request routing – DataPower to Collective member.	<port greater than 9080> HTTP/HTTPS from Gateway zone to Protected zone.
18	External billing service – Management cluster connecting to external billing service (when configured for billing). If you are using Stripe as your external billing service, you must enable connections with the Stripe API at: https://stripe.com/files/ips/ips_api.json. You can also view the Stripe IP addresses at the following URL: https://stripe.com/docs/ips.	443 HTTPS from Management zone to Public zone.

Note:
You can customize the port values for the Cloud Manager, API Manager, and Developer Portal user interfaces. However, the Cloud Management Console (CMC) does not notify the Developer Portal when the Cloud Manager port connection changes. As a result, if you changed from the default port (443), the command **set_apim_host** must be run to specify the custom port. For instructions, see set_apim_host. The port number entered for the Developer Portal must match that of the Portal API Port configured in Cloud Manager > UI Settings.

You can view defined ports from the Cloud Manager, click Settings > TLS Profiles. For more information, see Specifying the cloud settings.

Communications inside the Gateway zone

There are a number of important points to note regarding the communications within the Gateway zone:

- Port 5550 is used, by default.
- Port designation is configurable on each Gateway server. However, if you change the port on a Gateway server, you must also change that Gateway server definition in the Cloud Manager (in API Connect).
- 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 all of the Management servers.

Communications inside the Management zone

The Management zone represents the region containing the collection of API Connect Management servers that store the cloud configuration and control communication.

There are a number of important points to note regarding the communications within the Management zone:

- Ports 11526 and 21526 are used to synchronize configuration data (including Organizations, Users, and Products).
- Port 9600 is used to replicate analytics data.
- Ports 443 and 9022 are used to manage the topology.
- Port 2443 is used to communicate HTTPS content between Management servers.
- All Management servers must be able to reach all of the other Management servers.
- All Management servers must be able to reach all Gateway servers.
- All Management servers must be able to communicate HTTPS content with the external billing service when billing is configured. If you are using Stripe as your external billing service, you must enable HTTPS communication on port 443 with the Stripe API: https://stripe.com/files/ips/ips_api.json.
  Note: API Connect does not support using webhooks with Stripe. You can see a list of the Stripe IP addresses at: https://stripe.com/docs/ips.

Communications inside the Developer Portal zone

The Developer Portal zone represents the region containing the collection of Developer Portal servers.

- Ports 3306, 4567, 4568, 4444, and 30865 are all used to enable communication between the Developer Portal nodes that are in the cluster.
- Opening port 22 is used to allow communication between the Developer Portal nodes via SSH. Port 22 is an inbound port.
- All Developer Portal nodes must open ports 443 and 80 so that machines can serve web traffic. Port 443 is an inbound port.
- All Developer Portal servers must be able to communicate HTTPS content with the external billing service when billing is configured. If you are using Stripe as your external billing service, you must enable HTTPS communication on port 443 with the Stripe API: https://stripe.com/files/ips/ips_api.json.
  Note: API Connect does not support using webhooks with Stripe. You can see a list of the Stripe IP addresses at: https://stripe.com/docs/ips.
- Port 2443 is an outbound port, and it enables functionality for background synchronization and webhooks. By opening the port, the Developer Portal can get APIs from its associated API Manager.

Common outbound ports

Configure these ports for all servers as appropriate for your enterprise:

- 25 SMTP (only for Management servers; configurable)
- 53 DNS (Name resolution)
- 123 NTP (Clock synchronization)
- 162 SNMP Traps (currently only for Gateway servers)
- 389 LDAP (configurable)
- ICMP (Between Management zone and Developer Portal zone)

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.

When adding the Gateway server to the cluster in the Cloud Manager UI, specify the name of the interface that is used for processing incoming API calls; consult with your network administrator to ascertain the IP address that receives this traffic, and specify the interface that is configured for that IP address. If you are using a dedicated interface for administrative access, mgt0 for example, you must specify the administrative IP address when adding the Gateway server. For more information, see Adding a Gateway server.

Webhook communication in API Connect

Webhooks are a subscription service, where the Developer Portal sends messages to a Management node to be subscribed to events that occur. The Developer Portal connects to port 443 on the Management server to subscribe to webhooks.

When an event occurs, a Management server contacts the Developer Portal to inform it that the event occurred and proceeds to send the event data. The event data is sent to the Developer Portal on port 2443.

Related tasks

- Installing IBM API Connect

Related reference

- IBM API Connect Version 5.0 requirements
- IBM API Connect topology

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a second NIC

The Management server can support two NICs.

About this task

The Management server supports a dual NIC configuration, which allows you to configure one NIC for internal use and utilize the other NIC for more public exchanges. The first NIC, eth0, communicates with all Management servers. The second NIC, eth1, can expose your APIs to a broader external audience.

In this dual NIC configuration, the Cloud Manager, API Manager, and Developer Portal components of the IBM® API Connect solution are served by both NICs.

Note: SSH login is supported for the eth0 interface, but not eth1.

Procedure

1. From the VM console, deploy the OVA template.
2. Right-click on Management Server VM and select Edit Settings.
3. Add NIC and click OK.
4. The VM restarts automatically after a configuration change. If an automatic restart does not occur, restart the VM manually.
   Note: An automatic restart is the expected behavior of the VM after a configuration change. If the VM auto-detects a change in it's configuration, it forces a restart. Contact IBM Support for more information about the automatic restart.

Results

The second NIC uses DHCP to get an IP address. If DHCP is not available, you can manually provide an IP address. For more information, see Configuring to use static addressing.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Load balancing in IBM API Connect

In the IBM® API Connect cloud, multiple Management servers and multiple Gateway servers can be used to achieve high availability or resilience. If multiple servers are used in a cloud, consider the following aspects of load balancing: API calls, the user interfaces, and communications between servers in the API Connect cloud.

Load balancing API calls

Inbound API calls can be load balanced by using the self-balancing capabilities of the DataPower® appliance, or by using an external load balancer of your choice.

To use the DataPower appliance for load balancing, you must supply an IP address or a host name that resolves to that IP address. The IP address must not be used for any existing machine and must be accessible to the consumers of your APIs. The IP address is used to automatically configure a standby group on DataPower. For more information, see Configuring the initial Gateway cluster and Standby controls.

If you are using an external load balancer, the cluster address that is specified in the Gateway Cluster settings pane in the API Manager cloud console must resolve to the load balancer. Also, the domain name that is specified for the Developer Portal and its subdomains must resolve to the load balancer. For more information, see Configuring the initial Gateway cluster. For information about which ports are used, see Firewall requirements.

Load balancing the user interfaces

If you have more than one Management server, you must use an external load balancer to provide load balancing for the user interfaces: cloud console, API Manager and Developer Portal. The cluster address that is specified in the Management Cluster settings pane in the API Manager cloud console must resolve to the load balancer. For more information, see Configuring the Management service

../com.ibm.apic.cmc.doc/create_node_mgmt_1.html. For information about which ports are used, see Firewall requirements.

Tip: You can call a health check API from your external load balancer to determine whether a specific Management server is up (ready to receive requests) or down (there is an issue with the server and it cannot receive requests). For more information, see Obtain health check data of Management servers by using REST API calls.

Load balancing the user interfaces for the Developer Portal

The load balancer must be configured to ensure that the HTTP Host header that arrives at the Developer Portal machine is set to the host name of the Developer Portal site.
Note: You can apply these principles to use a reverse proxy in front of the Developer Portal. However, you are not allowed to modify the Developer Portal URL in a reverse proxy, or modify the Host header. It must be a transparent proxy.
To check the health of a Developer Portal server, it is sufficient to configure a health check only on port 443. If there is a database problem on the server, the web server shuts down, so the load balancer needs to perform a health check only on port 443 of each portal server to decide where to route traffic.

V5.0.8 + Tip: You can call a simple health check API from your external load balancer to determine whether a specific Developer Portal site in a cluster is running. For more information, see Obtaining simple health check data of Developer Portal sites by using a REST API call.
In the following example:

```
Load Balancer hostname: balancer.myorg.com IP: 10.0.0.1
          AdvPortal1 hostname: adp1.myorg.com IP: 10.0.0.2
          AdvPortal2 hostname: adp2.myorg.com IP: 10.0.0.3
          AdvPortal3 hostname: adp3.myorg.com IP: 10.0.0.4
          Site1: developer.myorg.com
          Site2: special-developers.myorg.com
```

The DNS entries, `developer.myorg.com` and `special-developer.myorg.com`, are mapped to the IP address of the load balancer, 10.0.0.1. The mapped IP address sends the request to one of the following IP addresses while maintaining the HTTP Host header of the original request:

- 10.0.0.2
- 10.0.0.3
- 10.0.0.4

The HTTP Host header of the original request is either `developer.myorg.com` or `special-developers.myorg.com`, depending on what you have entered in your browser's address bar.
The URL that you enter in a browser to access the Developer Portal site must match the URL that is specified in the API Manager UI Catalog for the IBM Developer Portal.

If you are using datapower to proxy or load balance your Developer Portal, you must ensure that the variable *Rewrite Host Names When Gatewaying* in the Global HTTP options tab is set to FALSE. For more information, see Rewriting Host names in the DataPower user interface when you redirect.
CAUTION:
Rewriting the Developer Portal URL in the Gateway is not recommended. By rewriting the Developer Portal URL, links in the following emails will not work:

- User activation emails
- Password reset emails
- Emails from the Developer Portal

External integration to authentication providers will not work because the Developer Portal will send the incorrect URL as the redirect_uri. In addition, loading resources such as files, js, and .css might not work.
Note: The Developer Portal listens only on ports 80 and 443. Port 80 redirects immediately to port 443 and uses the HTTPS protocol.

Rewriting `Host` names in the DataPower user interface when you redirect

To demultiplex, some protocols have distinct name-based elements that are separate from the URL. HTTP uses the Host header for this purpose; this functionality can be disabled.

To disable the functionality, you must search for `Multi-Protocol Gateway`, in the search bar that is located in the upper left of the screen. After you open the Multi-Protocol Gateway window, you can disable the functionality in the HTTP Options tab.

By disabling the functionality, the remote server receives a request that reflects the information as it arrived at the DataPower service. Disabling the functionality for web servers that issue redirects is preferred, as they depend on the **`Host`** header for the value of their redirect.

Note: By default, when a request is proxied by DataPower, DataPower does not propagate the original value that is receives from the **`Host`** header. DataPower rewrites to represent the final route. Disabling the functionality enables you to proxy the received header.

Supporting mutual TLS on the load balancer

Whether mutual TLS support is enabled on the load balancer effects how certificates are updated.

The standard configuration for a load balancer does not enable mutual TLS. When mutual TLS is not enabled, the external load balancer will terminate TLS on ingress (either anonymous or mutual), and connect to API Connect using anonymous (1-way) on egress. In this setup, certificates require updating only on the load balancer.

If mutual TLS is enabled between API Connect and the load balancer, then certificates on API Connect require updating (as well as the certificates on the load balancer). Depending on the load balancer configuration, either the identity store, trust store or both will require updating.

Load balancing communications between servers

Communications between multiple Gateway servers, between multiple Management servers, and between Management servers and Gateways servers within the same cloud are handled automatically in the cloud configuration. However, if your topology is complex, you might have to ensure that the necessary ports are open across firewalls. For information about which ports are used, see Firewall requirements.

You can use a load balancer to optimize internal communications between the Gateway servers and Management servers. A load balancer is typically used when you want to gain finer control over load distribution, particularly if your API Connect cloud spans two of more different geographical locations. For more information, see Using an internal load balancer to optimize internal communications.

- **Obtain health check data of Management servers by using REST API calls**
  Call a health check API from your external load balancer to dynamically determine whether a specific Management server is up (ready to receive requests), or down (there is an issue with the server and it cannot receive requests). In addition there is an accompanying health check API that can be called and, if authenticated, it returns more detailed information about the health status of the Management server.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtain health check data of Management servers by using REST API calls

Call a health check API from your external load balancer to dynamically determine whether a specific Management server is up (ready to receive requests), or down (there is an issue with the server and it cannot receive requests). In addition there is an accompanying health check API that can be called and, if authenticated, it returns more detailed information about the health status of the Management server.

Tests run during the health checks

You can run either a simple health check or a detailed health check, but both checks run the same tests. The detailed health check returns more information for the same tests than the simple health check. The tests report the following information:

1. Whether the server is considered to be in an active state.
2. Whether the server is able to read documents from the configuration database.
3. Whether the server is able to write documents to the configuration database.
4. Whether the server is able to use the CouchDB database.
5. Whether the server is a member of a management service with the provided display name.
6. Whether the server is in the list of servers.
7. Whether the cloud is dissociated.

Simple health check API

The simple health check API runs a series of tests on the Management server to determine whether the server is working correctly enough to be part of a load balancer configuration. See Tests run during the health checks for a list of the tests that are included in the simple health check.

To run a simple health check, configure your load balancer to make the following call:

```
https://address_of_mgmt_server/v1/servers/self/lb-health-check
```

Where *address_of_mgmt_server* is the IP address or hostname of a specific Management server. This cannot be the load balancing address. You can optionally add the display name of the management service to the call. Adding the name ensures that the load balancer includes only management servers for a specific IBM® API Connect cloud, and is not misconfigured to reference one or more management servers in a different cloud. You can customize the default display name `Management` of the management service of your clouds to be more specific. For example, you might use something like `Production`, `Development`, or `Test` as a display name. If you want to use the additional name in the call, issue the following call:

```
https://address_of_mgmt_server/v1/servers/self/lb-health-check?clusterName=name_of_mgmt_service
```

Where the following substitutions are made according to your environment:

- *address_of_mgmt_server* is the IP address or hostname of a specific Management server. This cannot be the load balancing address.
- *name_of_mgmt_service* is the display name of the management service in the Cloud Manager. See Configuring the Management service for more information.

The following response messages are returned:

- On success (if all tests pass), the API responds with an HTTP status code of `200`, and the response body contains `{"status":"success"}`.
- On failure (if any one test fails), the API responds with an HTTP status code of `409`, and the response body contains `{"status":"failure"}`.
- Any other `3XX`, `4XX`, or `5XX` status code is a health check failure.

This API is available both through HTTP and HTTPS protocols, and does not require authentication.

Detailed health check API

The detailed health check API runs a series of tests on the Management server to determine whether the server is working correctly enough to be part of a load balancer configuration, and returns the results of those tests. See Tests run during the health checks for a list of the tests that are included in the detailed health check.

To run a detailed health check, configure your load balancer to make the following call:

```
https://address_of_mgmt_server/v1/servers/self/lb-health-check/details
```

Where *address_of_mgmt_server* is the IP address or hostname of a specific Management server. This cannot be the load balancing address. You can optionally add the display name of the management service to the call. Adding the name ensures that the load balancer includes only management servers for a specific IBM API Connect cloud, and is not misconfigured to reference one or more management servers in a different cloud. You can customize the default display name `Management` of the management service of your clouds to be more specific. For example, you might use something like `Production`, `Development`, or `Test` as a display name. If you want to use the additional name in the call, issue the following call:

```
https://address_of_mgmt_server/v1/servers/self/lb-health-check/details?clusterName=name_of_mgmt_service
```

Where the following substitutions are made according to your environment:

- *address_of_mgmt_server* is the IP address or hostname of a specific Management server. This cannot be the load balancing address.
- *name_of_mgmt_service* is the display name of the management service in the Cloud Manager. See Configuring the Management service for more information.

Important: The detailed health check API call must be authenticated (you must pass Cloud Manager admin credentials to the Management server). The credentials that are required for the detailed health check must be prefixed with `'cmc/account_name'`. An authentication error occurs if you do not apply the `'cmc/'` prefix.
The following response messages are returned:

- On success (if all tests pass), the API responds with an HTTP status code of `200`, and the response body contains details of each individual test.
- On failure (if any one test fails), the API responds with an HTTP status code of `409`, and the response body contains details of each individual test, including information about the test or tests that have failed.
- Any other `3xx`, `4xx`, or `5xx` status code is a health check failure.

Query parameters

The following table shows the query parameters that can be used with both health check APIs.

Table 1. Query parameters for the health check APIs

Name of parameter	Type	Default	Description
`onSuccess`	Integer	`200`	The HTTP status code to respond with on success.
`onFailure`	Integer	`409`	The HTTP status code to respond with on failure.
`testIsActive`	Boolean	`true`	Controls whether the test to check if the server is considered to be in an active state, is run.
`testIsReadable`	Boolean	`true`	Controls whether the test to check if the server is able to read documents from database, is run.
`testIsWritable`	Boolean	`false`	Controls whether the test to check if the server is able to write documents to the database, is run.
`testIsCouchUp`	Boolean	`true`	Controls whether the test to check if the server is able to use the CouchDB database, is run.
`testIsInCluster`	Boolean	`true`	Controls whether the test to check if the server is a member of a management service with the provided display name, is run. Note: If set to *true*, the `clusterName` parameter must also be set.
`clusterName`	String		Display name of the management service to use when running the test to check whether the server is a member of a specific management service. Note: If the `testIsInCluster` parameter is set to *true*, the `clusterName` must be set.
`testIsInServers`	Boolean	`true`	Controls whether the test to check if the server is in the list of servers, is run.
V5.0.8 + `failIfCloudIsDissociated` (version 5.0.8.3, and later)	Boolean	`false`	Specifies whether to fail the test if there are Management servers that are identified as dissociated. See Dissociation and your cloud for more information.
`testIsDissociated` (deprecated)	Boolean	`true`	Controls whether the test to check if the server is dissociated from the cloud, is run.

Example responses

The following table shows example responses for the health check APIs.

Table 2. Health check API response information

API type	Response

API type	Response

Simple health check API	200
	`{`
	`    "status" : "success"`
	`}`
	The Management server is up and ready to receive requests.
	409
	`{`
	`    "status" : "failure"`
	`}`
	There is an issue with the Management server, and it is not ready to receive requests.

Detailed health check API	200
	`{`
	`    "isInServers": "true",`
	`    "isInCluster": "unknown",`
	`    "isCouchUp": "true",`
	`    "isWritable": "unknown",`
	`    "isReadable": "true",`
	`    "isActive": "true",`
	`    // Comment: "isCloudDissociated" and "dissociatedServers" are only available in version 5.0.8.3, and later`
	`    "isCloudDissociated": "false",`
	`    "dissociatedServers": [],`
	`    // Comment: "isDissociated" is deprecated as of version 5.0.8.3`
	`    "isDissociated": "false",`
	`    "error": null`
	`}`
	The Management server is up and ready to receive requests.
	409
	`{`
	`    "isInServers": "true",`
	`    "isInCluster": "unknown",`
	`    "isCouchUp": "true",`
	`    "isWritable": "unknown",`
	`    "isReadable": "true",`
	`    "isActive": "true",`
	`    // Comment: "isCloudDissociated" and "dissociatedServers" are only available in version 5.0.8.3, and later`
	`    "isCloudDissociated": "true",`
	`    "dissociatedServers": [`
	`      "hostname_or_IP_address_of_dissociated_server_1",`
	`      "hostname_or_IP_address_of_dissociated_server_2"`
	`    ],`
	`    // Comment: "isDissociated" is deprecated as of version 5.0.8.3`
	`    "isDissociated": "false",`
	`    "error": null`
	` }`
	There is an issue with the Management server, and it is not ready to receive requests.

Related tasks

- V5.0.8 + Obtaining simple health check data of Developer Portal sites by using a REST API call

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring Catalogs and testing the Developer Portal

As an optional postrequisite to the installation of the Developer Portal, you can configure Catalogs and test that configuration before using the Developer Portal.

Before you begin

This task should be performed by an API Provider Organization owner or administrator.

Procedure

To configure Catalogs and test the Developer Portal, complete the following steps:

1. . Configure a Catalog to use the Developer Portal in the API Manager user interface.
   a. Log in to the API Manager user interface with a user that has Catalog edit permissions.
   b. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
   c. In the navigation pane, click Dashboard, then click the Catalog that you want to configure to use the Developer Portal.
   d. Click the Settings > Portal icon , then select IBM Developer Portal.
   e. Enter the full unique Developer Portal URL for this Catalog in the URL field:

   ```
   https://full_developer_portal_URL
   ```

   where *full_developer_portal_URL* must resolve to the IP address of the Developer Portal machine.
   For example:

   ```
   https://myhost.mydomain.com
   ```

   You can also use paths in your URL:

   ```
   https://myhost.mydomain.com/path_x/path_x_x
   ```

   Important:
   - The Developer Portal URL must be unique across all Catalogs.
   - The Developer Portal host name must contain only **lowercase** characters.

   f. Click Save, then log out of API Manager.
   Note: You are sent an email that contains a one-time log in link for the Developer Portal web admin user. The admin user can be used to administer the CMS capabilities of the Developer Portal, and the default email is sent as soon as the Developer Portal is installed.
2. Test the Developer Portal:
   a. To test the Developer Portal configuration, wait for the email and then visit the one-time log in link provided, or visit the URL for the Developer Portal in the browser. The page should display without errors.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

High availability for the Developer Portal

High Availability can be achieved by clustering Developer Portal sites, so that Developer Portal users will always have access to the site.

Note: The principles applied to load balancing in API Manager can be applied to load balancing Developer Portal traffic across multiple Developer Portal nodes. For more information, see Load balancing in IBM® API Connect
To learn about High Availability in the Developer Portal, use the following topic links:

- **High availability configurations for the Developer Portal**
  To sustain high availability for the Developer Portal, there are various conditions that must be met by the Developer Portal, and recommended configurations to apply to the Developer Portal cluster.
- **Configuring an NTP server**
  If the Developer Portal nodes cannot contact the default NTP server, you must configure a different NTP server.
- **Setting up a cluster of machines**
  You need to assign a primary node and set up your machines as a cluster to create Developer Portal high availability.
- **Adding a machine to a cluster**
  You can add machines to an existing cluster to alter the availability of the Developer Portal.

- **Removing a machine from a cluster**
  You can remove machines from a cluster to alter the availability of the Developer Portal.
- **Un-clustering a machine**
  Removing a machine from a cluster involves powering off the machine, whereas un-clustering the machine leaves you with a working machine that is standalone. By un-clustering a machine, you are aiming to reconfigure a machine to be a standalone Developer Portal server.
- **Restoring an inactive clustered machine**
  You can restore a machine that has rebooted or been down when it is part of a cluster.
- **Restoring a whole cluster**
  You can restore entire clusters that have rebooted or been inactive for some time.
- **Cloning a site into the same cluster**
  You can make a copy of an Developer Portal site by cloning it.
- **Cloning a site into a new cluster**
  You can clone an Developer Portal site into a new cluster.
- **Load balancing the requests the Cloud Manager sends to the Developer Portal**
  You can balance the load of requests sent from the Cloud Manager to the Developer Portal
- V5.0.7+ **Obtaining health check data of Developer Portal servers by using a REST API call**
  You can check the status of a Developer Portal cluster by calling a cluster health REST API.
- V5.0.8+ **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.
- **Useful commands for use with a running node**
  By entering the relevant command, you can return information regarding a running node. This information spans from the status of the node, to the platform names that each site is on.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

High availability configurations for the Developer Portal

To sustain high availability for the Developer Portal, there are various conditions that must be met by the Developer Portal, and recommended configurations to apply to the Developer Portal cluster.

If there is any network loss between Developer Portal servers, or if the servers lose power, the priority for the Developer Portal is to prevent database divergence.

To prevent database divergence, every functioning Developer Portal server prioritizes data consistency and calculates the total number of functional servers it can communicate with, divided by the total number of servers in the cluster.
Note: The total number of functional servers includes the server that performs the calculation.
If the result of the calculation is not a value that is greater than 50%, the server that performed the calculation goes into non-Primary state, which means it refuses all database requests and web traffic. This server will automatically rejoin the cluster and accept web traffic only when the result of the calculation is greater than 50%.

Note: If servers are brought down, whether it is the entire server or just the database, by performing a controlled shutdown or running any of the following commands in the CLI:

- `stop_db`
- `sudo service mysql stop`
- `sudo halt -p`
- `sudo reboot`

Then, the calculation does not need to be performed, and the other servers will continue as normal.
To enable effective high availability, you need latency that is less than 50ms between data centers to avoid the risk of performance degradation. Servers with uniform specifications are required as any write actions occur at the speed of the slowest cluster member, as the write actions are synchronous across the cluster.

Therefore, it is recommended that there are a minimum of three servers in each cluster for the high availability configuration. The three nodes 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.

Information about clustering

- All active servers are in a Primary state when you run `status` or `dbstatus` commands

- If network disruption occurs between two sets of equal numbered servers, then all servers go into non-Primary state
- The Developer Portal server databases are synchronous copies of each other. The write performance is constrained by the time taken for all servers to agree to perform a write command ,and commit the updates to maintain database synchronization. Therefore, a larger number of servers in a cluster lowers write performance due to the synchronization of write commands across the cluster.

Note: A three data center configuration provides automatic failover regardless of which data center goes down, while a two data center configuration only provides automatic failover if the Disaster Recovery data center goes down. For more information, see dbstatus and status.

Three data center high availability

Recommended approach for three data center high availability

Place one server in each data center, which ensures that failover is automatic. If a server fails, then the other two can continue functioning without interruption or data loss. You must take the following considerations into account for this approach:

- It requires the use of three data centers
- If you route all your web traffic to a single data center, then you will have only a single server to serve all of the traffic

Alternative approach for three data center high availability

Place two servers in each data center, which ensures that failover is automatic. Two servers can fail, and the other four can carry on without interruption or data loss. There is some extra configuration required for this approach:

- You must run `gmcastsegment` on servers to assign a unique segment ID on the servers in each data center to get the best performance:
  - Run `gmcastsegment 1` on both servers in the first data center
  - Run `gmcastsegment 2` on both servers in the second data center
  - Run `gmcastsegment 3` on both servers in the third data center

You must also take the following considerations into account for this approach:

- It requires the use of three data centers
- You need more hardware or virtual machines
- There is a greater risk for performance degradation

Two data center high availability

Recommended approach for two data center high availability

Place two servers in each data center. By running `pcweight`, one datacenter becomes the Primary data center, while the other becomes the Disaster Recovery data center. Failover is automatic if the Disaster Recovery data center fails, or if a single server fails. The remaining servers resume without interruption or data loss. There is some extra configuration required for this approach:

- You must run `gmcastsegment` on servers to assign a unique segment ID on the servers in each data center to get the best performance:
  - Run `gmcastsegment 1` on both servers in the first data center
  - Run `gmcastsegment 2` on both servers in the second data center
- You must run `pcweight` to assign server weights to three of the four servers:
  - Run `pcweight 2` on both servers in the Primary data center, and on one server in the Disaster Recovery data center

You must also take the following consideration into account for this approach:

- If both servers in the Primary data center fail, you must initiate failover manually. For more information, see Performing manual failover for Developer Portal servers

Alternative approaches for two data center high availability

There are two alternative approaches that you can follow for two data center high availability:

1. Place two servers in the Primary data center, and one server in the Disaster Recovery data center, which ensures that failover is automatic if the Disaster Recovery data center fails or a single server in the Primary data center fails. The remaining servers resume working with no interruption or data loss. There is some extra configuration required for this approach:
   - You must run `gmcastsegment 1` on servers to assign a unique segment ID on the servers in the Primary data center to get the best performance:
     - Run `gmcastsegment 1` on both servers in the Primary data center

   You must also take the following consideration into account for this approach:
   - If both servers in the Primary data center fail, or the entire Primary data center fails, you must initiate failover manually. For more information, see Performing manual failover for Developer Portal servers
2. Alternatively, place one server in each data center. By running `pcweight`, one data center becomes the Primary data center, while the other becomes the Disaster Recovery data center. Failover is automatic if the Disaster Recovery data center fails, you

only need two servers, and all servers can accept traffic. There is some extra configuration required for this approach:

- You must run `pcweight 2` to assign a server weight to the server in the Primary data center:
  - Run `pcweight 2` on the server in the Primary data center

You must also take the following considerations into account for this approach:

- If you route all your web traffic to a single data center, then you will have only a single server to serve all of the traffic
- If the Primary data center fails, you must initiate failover manually. For more information, see Performing manual failover for Developer Portal servers
- If either server fails, there will only be one server to serve all of the traffic

Important: Before you perform manual failover, you must ensure that any other nodes in the Developer Portal cluster are also down. You can either ensure that the database is stopped, or the machine is powered off. If any nodes are still running when you perform manual failover, the contents of the database diverge on the nodes, and the divergent nodes do not cluster together. Then, the nodes must be manually clustered together, which results in data loss.

- **Performing manual failover for Developer Portal servers**
  If you a network disruption occurs between your data centers, and your Primary data center goes down, your Disaster Recovery data center can enter a non-Primary state. To force the Disaster Recovery data center into a Primary state, you can perform manual failover.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Performing manual failover for Developer Portal servers

If you a network disruption occurs between your data centers, and your Primary data center goes down, your Disaster Recovery data center can enter a non-Primary state. To force the Disaster Recovery data center into a Primary state, you can perform manual failover.

Note: You can run `status` or `dbstatus` to determine which servers in a cluster show the non-Primary database status.
If some servers display `STOPPED` as a status, then run `bootastrap_cluster -b`.

If some servers display `UNREACHABLE` as a status, then run `bootastrap_cluster -bi`.

- V5.0.6 + Run the following command on a single server in the data center that you want to become Primary; it will find all reachable non-Primary servers and set them back to Primary state:

  ```
  bootstrap_cluster -pi
  ```

- V5.0.4 + For API Connect Version 5.0.4.0.and 5.0.5.0, run the following command on all servers in your Developer Portal cluster that you want to perform manual failover on and go into Primary state:

  ```
  set_primary
  ```

- V5.0.1 + For API Connect 5.0.1 to 5.0.3, run the following command on a single server in the Datacenter that you want to become Primary:

  ```
  bootstrap_cluster -pi
  ```

- For API Connect Version 5.0.0.1 and earlier, run the following command on all servers in your Developer Portal cluster that you want to perform manual failover on and go into Primary state:

  ```
  sudo mysql -e "SET GLOBAL wsrep_provider_options='pc.bootstrap=true';"
  ```

Important: Before you perform manual failover, you must ensure that any other nodes in the Developer Portal cluster are also down. You can either ensure that the database is stopped, or the machine is powered off. If any nodes are still running when you perform manual failover, the contents of the database diverge on the nodes, and the divergent nodes do not cluster together. Then, the nodes must be manually clustered together, which results in data loss.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring an NTP server

If the Developer Portal nodes cannot contact the default NTP server, you must configure a different NTP server.

About this task

Note: You do not need to configure an NTP server for a stand-alone machine.
The Developer Portal comes preconfigured to interact with one of the following Debian NTP servers:

```
0.debian.pool.ntp.org
1.debian.pool.ntp.org
2.debian.pool.ntp.org
3.debian.pool.ntp.org
```

V5.0.8 + The Developer Portal comes preconfigured to interact with one of the following Ubuntu NTP servers:

```
0.ubuntu.pool.ntp.org
1.ubuntu.pool.ntp.org
2.ubuntu.pool.ntp.org
3.ubuntu.pool.ntp.org
```

If you network does not allow the Developer Portal to reach those servers via UDP on port 123, then you must change the NTP servers.

All cluster members must be synchronized to the same NTP server, or the file synchronization that happens between the nodes does not function.

Important: You must apply the changes to each cluster member.

Procedure

1. To set up the NTP servers that are used, run the following command:

   ```
   set_ntp ntp_server_n
   ```

   Where *ntp_server_n* is the IP address or fully qualified hostname of an NTP server. If you have multiple NTP servers, you can enter them in the same format, within the same command, separated by a single space.

   Alternatively, to reset the servers to their default configurations, run the following command:

   ```
   set_ntp -d
   ```

2. To see the configured NTP servers, run the following command:

   ```
   set_ntp -s
   ```

Results

The NTP server that is being used is changed to allow for successful file synchronization between the nodes.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Setting up a cluster of machines

You need to assign a primary node and set up your machines as a cluster to create Developer Portal high availability.

Before you begin

To enable a high-availability cluster, you need to configure a cluster with a minimum of three nodes; this requirement is due to split brain detection in the database layer. Split brain detection works on a majority voting algorithm, therefore, at least three nodes are required to ensure that the remaining two nodes can detect that they are in the majority at 66% of the cluster. If a node in a cluster of two nodes loses communication with the one other node, then both nodes have only 50% of the cluster and so are not in a majority.

Additionally, due to this algorithm, when a cluster of more than three nodes is required, an odd number of nodes gives you optimum redundancy. Nodes that are not in the majority portion of the cluster do not service database queries, so both nodes in the 50% case would not execute queries.

Note: To check the status of a machine in a cluster (or standalone), run the following command:

```
status
```

If the status check is a success, you will receive a message that ends with the following line:

```
SUCCESS: All services are Up and the cluster timestamps are in sync
```

If the machine is not clustered, the message will be:

```
SUCCESS: All services are Up
```

To check the database status for the entire cluster of machines, run the following the command:

```
bootstrap_cluster -s
```

The **bootstrap_cluster -s** command will check the database status on all defined cluster members. If the check is successful, you receive a message that ends with the following line:

```
SUCCESS: All reachable cluster members reporting Primary database status
```

Procedure

To set up a cluster of machines, complete the following steps, depending on the version of IBM® API Connect you are using:

a. For each node you want to cluster, do a standard setup for a stand-alone node, as in the topic Installing the Developer Portal.
b. On the first node you want to cluster, enter the following command:

```
set_cluster_members -c
```

c. To add additional nodes to the cluster, follow the steps in Adding a machine to a cluster.
d. Optional: If you used the wrong IP address when you first called the `set_cluster_members` command, you can de-cluster a node with the following command:

```
set_cluster_members -d
```

After running the `-d` command, you can attempt the procedure again with the correct IP address so the cluster operation succeeds.

Results

Your cluster is created.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a machine to a cluster

You can add machines to an existing cluster to alter the availability of the Developer Portal.

Before you begin

Before you add your selected machine to the existing cluster, you must configure it to be like a standalone machine by following steps 1 - 6 in Installing the Developer Portal.
V5.0.2+ Before you add your selected machine to the existing cluster, you must configure it to be like a standalone machine by following steps 1 - 3 in Installing the Developer Portal. Steps 4 - 6 are done automatically when the selected machine joins the cluster.

Procedure

To add a machine to a cluster, complete the following steps, depending on the version of IBM® API Connect you are using:

a. Log in to the new machine and enter the following command:

```
set_cluster_members any_available_existing_cluster_member
```

Where *any_available_existing_cluster_member* is the hostname or IP address of an available existing machine in the cluster.
As a result of the command `set_cluster_members`, one of the cluster members will have been selected to be the donor machine for the database transfer. This machine will scan all database files to transfer them to the new machine. This might take some time to complete.
The new machine is successfully added to the cluster if running the command,

```
status
```

returns the following message:

```
SUCCESS: All services are Up and the cluster timestamps are in sync
```

If you do not receive the message indicating success, check /var/log/syslog, and /var/log/devportal/command_line.log for progress and errors.
Note: Processes running on the donor machine might take significantly longer to run while the preceding command is running.
For more information on the `set_cluster_members` command, see set_cluster_members.

Results

You have added or removed a machine to or from an existing cluster.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing a machine from a cluster

You can remove machines from a cluster to alter the availability of the Developer Portal.

Procedure

1. Quiesce the queue processors and stop the database on this node by entering the following command:

   ```
   queue_run -q
   sudo service mysql stop
   ```

2. When the preceding command has completed, enter the following command for all other cluster members to remove this machine from the cluster:

   ```
   set_cluster_members node_A node_C
   ```

   Where *node_B* is the node that has been removed.
3. Power off the node that has been removed from the cluster.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Un-clustering a machine

Removing a machine from a cluster involves powering off the machine, whereas un-clustering the machine leaves you with a working machine that is standalone. By un-clustering a machine, you are aiming to reconfigure a machine to be a standalone Developer Portal server.

About this task

You must be an administrator with access to the Developer Portal OVA CLI.

Procedure

1. On the machine that you want to be standalone, enter the following command:

   ```
   set_cluster_members -d
   ```

2. On the other machines in the cluster, you can either reconfigure them to form a cluster with the other remaining machines by entering the following command:

   ```
   set_cluster_members node_A node_C
   ```

   Where `node_B` is the node that has been removed.
   or shut down the machines by entering the following commands:

   ```
   queue_run -q
   sudo service mysql stop
   sudo halt -p
   ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restoring an inactive clustered machine

You can restore a machine that has rebooted or been down when it is part of a cluster.

About this task

When the machine first starts mysql, the following events can occur, taking up some time:

- Position recovery; the time taken will depend on whether the database has previously shutdown cleanly and how many sites the portal is hosting.
- A synced node will be requested, at random, to become a donor node and send incremental state transfer.
- If the incremental state transfer fails, due to the database logs having wrapped, then a non-incremental state transfer is requested, this causes `xtrabackup` to take a backup of a donor node and tar it across the network using socat on port `4444` to the joining node. It will also be reading the database logs, to ensure any new changes can be sent to the joining node as the donor can still accept writes to the database. Monitor /var/log/syslog for updates or errors on this process. If the SST fails then check /var/lib/innobackup.backup.log on the donor node for details.

Note: Processes running on the donor node might take significantly longer to run while the preceding process is running.
`lsyncd` / `csync2` is configured in a ring, the ring sends from the 1st machine to the 2nd, and then the 2nd to the 3rd, until the last machine sends to the 1st. Each machine maintains a database for itself and for the node it is sending to in /var/lib/csync2.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restoring a whole cluster

You can restore entire clusters that have rebooted or been inactive for some time.

Procedure

If all cluster members are down, restart all of the machines, and then on any single machine complete the following steps:

1. Run the following command to show the current state of the database on all cluster members that are reachable:

   ```
   bootstrap_cluster -s
   ```

If all cluster members show the database status as STOPPED, or are in Primary or non-Primary mode, then run the following command:

```
bootstrap_cluster -b
```

This command will find the machine with the most up-to-date database, and uses the machine to bootstrap the cluster. After this machine has bootstrapped, it will start the database on the other machines consecutively so that they join the cluster.

2. If some of the cluster members show as Starting, you can wait for a minute before running **bootstrap_cluster -s** again and checking whether the cluster members show as STOPPED. However, if you know that the entire cluster is down, you can run the following command:

   ```
   bootstrap_cluster -bf
   ```

   which will stop the database on each member before running **bootstrap_cluster -b**

Results

You have rebooted a whole cluster.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Cloning a site into the same cluster

You can make a copy of an Developer Portal site by cloning it.

Before you begin

You must have an existing site that you want to clone.

About this task

You can either overwrite an existing site with a clone, or specify a URL and UUID.

Procedure

1. To overwrite an existing site with a clone:
   a. Run the following command:

      ```
      clone_site -f existing_source_url existing_destination_url
      ```

2. To specify the UUID and URL of the new clone:
   a. Run the following command:

      ```
      clone_site existing_source_url destination_uuid
          destination_url
      ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Cloning a site into a new cluster

You can clone an Developer Portal site into a new cluster.

Before you begin

You must have an existing site that you want to clone.

About this task

You can either overwrite an existing site with a clone, or specify a URL and UUID.

Procedure

To clone a site into a new cluster:

1. On the source cluster, run the following command:

   ```
   backup_site -u my.url.com
   ```

   A file is produced.
2. Copy the file that is produced from the `backup_site -u my.url.com` command, to the destination cluster by running the following command:

   ```
   scp file destinationhost:
   ```

   If the destination URL and UUID are the same:

   ```
   restore_site -c FILE_FROM_BACKUP_SITE
   ```

   If the destination URL and UUID are different:

   ```
   restore_site -c -i NEW_UUID -u NEW_URL FILE_FROM_BACKUP_SITE
   ```

3. Optional: If the destination site already exists, you must add `-f` onto the `restore_site` commands.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Load balancing the requests the Cloud Manager sends to the Developer Portal

You can balance the load of requests sent from the Cloud Manager to the Developer Portal

Load balancing requests from the CMC

The Cloud Manager accepts only a single IP address for the SSH requests that it sends to the Developer Portal to, so you must use a load balancer in TCP mode to front a cluster of Developer Portal machines if you want to ensure that the request from the Cloud Manager always finds an active cluster member. You can load balance an SSH connection in TCP mode as though you are load balancing an HTTPS pass through connection in TCP mode. The load balancer has no knowledge of the messages being exchanged, but does a basic health check to ensure that each node is healthy. Nodes that do not accept the connection are temporarily removed from the load balancer's target list.

Queue system

Each cluster has a cluster-wide queue, that the following actions go into:

- Add site
- Add language to a site
- Delete site
- Upgrade site
- Update site

The site queue is always processed in order of priority. The priority is any non-add_language tasks in the order they came into the queue, and then any add_language tasks in the order they came into the queue.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 +

Obtaining health check data of Developer Portal servers by using a REST API call

You can check the status of a Developer Portal cluster by calling a cluster health REST API.

About this task

From IBM® API Connect Version 5.0.7.2, you can use the cluster health REST API to dynamically determine the health and status of a Developer Portal cluster. The cluster health API returns details about each server in the cluster, including system uptime, disk usage, web server hits, and CPU usage.

Procedure

To call the cluster health REST API, complete the following steps:

1. Enter the following URL into a web browser:

   `https://portal_node:2443/node/clusterhealth`

   Where *portal_node* is the host name of the Developer Portal node that you want to check.
2. Supply the following user credentials:
   - User Name: `status`
   - Password: `!n0r1t5@C`

   Note: The status user can access only the cluster health API; this user cannot access any other parts of IBM API Connect. If required, the password of the status user, a portal Linux operating system user, can be changed in the Developer Portal CLI by the admin user.
3. Click OK.

   The web browser returns the following health and status information for each node in the cluster (or a standalone node) in JSON format.
   - Node details:
     - `rc` - return code
     - `ip` - IP address of the server
     - `hostname` - host name of the node
     - `version` - Developer Portal system version
     - `installDirectory` - installation directory
     - `os` - operating system and version
     - `uptime` - uptime of the node in seconds since the last reboot
     - `cpus` - number of CPUs
     - `usage` - CPU usage as a percentage of total CPU available
     - `load1` - average CPU load for the last 1 minute
     - `load5` - average CPU load for the last 5 minutes
     - `load15` - average CPU load for the last 15 minutes
   - Disk details:
     - `free` - amount in megabytes of free disk space
     - `used` - amount in megabytes of used disk space
     - `total` - amount in megabytes of total disk space
   - RAM details:
     - `free` - amount in megabytes of free RAM
     - `used` - amount in megabytes of used RAM
     - `total` - amount in megabytes of total RAM
   - Swap space details:
     - `free` - amount in megabytes of free swap space
     - `used` - amount in megabytes of used swap space
     - `total` - amount in megabytes of total swap space
   - Web traffic and database details:
     - `webHits` - number of web traffic hits per minute
     - `transactions` - number of database transactions per minute

Example

The following example shows a cluster of three nodes:

```
{
  "servers" : [
    {
      "rc" : 0,
      "ip" : "1.2.3.4",
      "hostname" : "portalnodename",
      "version" : "5.0.7.2-20170629-1751",
      "installDirectory" : "/",
      "os" : "Ubuntu 16.04",
      "uptime" : 175636,
      "cpu" : {
        "cpus" : 4,
        "usage" : 61,
        "load1" : 2.58,
        "load5" : 2.88,
        "load15" : 2.90
      },
      "disk" : {
        "free" : 81952,
        "used" : 18555,
        "total" : 100507
      },
      "ram" : {
        "free" : 7617,
        "used" : 8431,
        "total" : 16048
      },
      "swap" : {
        "free" : 1000,
        "used" : 1047,
        "total" : 2047
      },
      "webHits" : 7,
      "transactions" : 39560
    },
    {
      "rc" : 0,
      "ip" : "1.2.3.5",
      "hostname" : "portalnodename",
      "version" : "5.0.7.2-20170629-1751",
      "installDirectory" : "/",
      "os" : "Ubuntu 16.04",
      "uptime" : 679078,
      "cpu" : {
        "cpus" : 4,
        "usage" : 64,
        "load1" : 3.71,
        "load5" : 3.54,
        "load15" : 3.16
      },
      "disk" : {
        "free" : 82300,
        "used" : 18207,
        "total" : 100507
      },
      "ram" : {
        "free" : 6474,
        "used" : 9574,
        "total" : 16048
      },
      "swap" : {
        "free" : 1563,
        "used" : 484,
        "total" : 2047
      },
      "webHits" : 8,
      "transactions" : 50376
    },
    {
      "rc" : 0,
      "ip" : "1.2.3.6",
      "hostname" : "portalnodename",
      "version" : "5.0.7.2-20170629-1751",
      "installDirectory" : "/",
      "os" : "Ubuntu 16.04",
      "uptime" : 679160,
      "cpu" : {
        "cpus" : 4,
        "usage" : 66,
        "load1" : 3.26,
        "load5" : 3.07,
        "load15" : 2.96
      },
      "disk" : {
        "free" : 82921,
        "used" : 17586,
        "total" : 100507
      },
      "ram" : {
        "free" : 5607,
        "used" : 10441,
        "total" : 16048
      },
      "swap" : {
        "free" : 1470,
        "used" : 577,
        "total" : 2047
      },
      "webHits" : 3,
      "transactions" : 37069
    }
  ]
}
```

The following example shows the server array entry when a server cannot be contacted:

```
{
      "rc" : 255,
      "ip" : "1.2.3.4",
      "error" : "Host key verification failed.\r"
    }
```

Related tasks

- V5.0.8 + Obtaining simple health check data of Developer Portal sites by using a REST API call

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

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

From IBM® API Connect Version 5.0.8.1, 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 extremely 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
```

Useful commands for use with a running node

By entering the relevant command, you can return information regarding a running node. This information spans from the status of the node, to the platform names that each site is on.

Note: The following commands should be run as an admin user.
To display status information, run the following command:

```
status
```

To check the status of a machine in a cluster (or standalone), run the following command:

```
status
```

If the **status** check is a success, you receive a message that ends with the following:

```
SUCCESS: All services are Up and the cluster timestamps are in sync
```

If the machine is not clustered, the message you receive is:

```
SUCCESS: All services are Up
```

The **status** command displays the following status information:

- System and distribution version
- Cluster members, and whether they are active
- Configuration values
- Database status
- Web server and file synchronization
- Cluster timestamps

To show you a list of the time stamp files that are on a node, run the following command:

```
timestamps
```

Each node writes the current time to a file every minute and these get copied around the cluster by `csync2` and `lsyncd`. If any file is more then 90 seconds behind the current system time, or in front, then the command returns a non-zero return code. The code indicates how many timestamps were bad.
To show you the processes running on this machine that are performing any of the tasks mentioned in the Queue System section of the Load balancing the requests the Cloud Manager sends to the Developer Portal topic, run the following command:

```
show_queue_executors
```

Note: all task types except `add_language` are considered priority tasks and `add_language` tasks are considered non-priority tasks. It also shows the locks that are held across the cluster and if any tasks are waiting on the queue that are currently executable on this node then they will show, also.
To show all tasks that have run, and re-run, on the queue and information about that execution, including the return code, time taken and which host they ran on, run the following command:

```
show_queue_history
```

To show any tasks, runs or re-runs, that have failed to return a zero return code, run the following command:

```
show_queue_history -n
```

To show any tasks that have failed to return a zero return code even after all retry attempts (The number of attempts is controlled by QUEUE_TASK_MAX_RETRIES in config/config.ini), run the following command:

```
show_queue_history -f
```

A value of `Killed` in the rc column means that `run_site_queue` found a queue lock without the process that originally locked it still running so it killed the old queue lock and therefore does not know the rc of the task. This counts as a task failure, so increment the retry count and the task will re-run at the next opportunity.
To show the queue history for just the single site specified, enter the following command:

```
show_queue_history -q orguuid.envuuid
```

To show all the columns, enter the following command in conjunction with any of the preceding commands:

```
show_queue_history -a
```

Some columns are not shown by default. If you find a task that has completely failed and you want to retry it, you can use the preceding command to show all columns and then highlight the contents of the JSON column. Now enter the following command:

```
echo '<paste the json contents here>' | site_action
```

Note: If you re-issue an add action then all the corresponding `add_language` actions will be added to the queue by `site_action` automatically so you do not need to also reissue them.
To delete all traces of a site if it appears to be in a broken state and it cannot be deleted by the CMC, run the following command:

```
delete_site orguuid.envuuid
```

To delete all traces of a site if you cannot find the corresponding UUID for the site, enter the following command:

```
delete_site -u mysite.url.com
```

You should first check the URL you are passing in is not in the list returned by `list_sites`. If the site does appear in `list_sites`, then delete the site by passing the UUID to `delete_site` like `delete_site orguuid.envuuid` as this will then also delete the mapping between the UUID and the URL.
config.ini - changes to values in config.ini are effectively immediately (`run_site_queue` checks the file regularly on the node you edited the file on, and within a few seconds more on all other nodes.

You can find which platform name each site is on by running

```
list_sites -p
```

then you can do the following on both nodes involved:

```
sudo rm -f /var/lib/csync2/*
```

then run

```
sudo service apim_dcluster update-cluster
```

on the sending and receiving nodes.
V5.0.7+ Tip: From API Connect Version 5.0.7.2, you can check the status of a Developer Portal cluster by calling a cluster health REST API. For more information, see Obtaining health check data of Developer Portal servers by using a REST API call.

Failed log in commands

To show the number of failed log in attempts per site and user, and reset the failed log in count per site either for a specific user or all users, run the following command:

```
Usage: reset_locked_user [-s] [-l [sitename]] [-r sitename/-a accountname/-a]
```

Where `-s` lists sites by URL, and `-l` shows all locked accounts per site either for all sites, or for *sitename* if the value is provided.
`-r` resets the failed log in attempts to zero for *sitename* and *accountname*.

`-a` is all sites or all accounts.

To show all locked users on all sites, run the following command:

```
reset_locked_user -l
```

To show locked users on a single site, run:

```
reset_locked_user -l mysite.com
```

Then, you can unlock specific users on specific sites by running one of the following commands:

```
reset_locked_user -r mysite.com admin
```

Or:

```
reset_locked_user -r mysite.com myuser@email.com
```

You can reset the access for all users on a single site by running the following command:

```
reset_locked_user -r mysite.com -a
```

Or, to reset the access for all users on all accounts, run the following command:

```
reset_locked_user -r -a -a
```

Bootstrap commands

To check the database status for the entire cluster of machines, run the following the command:

```
bootstrap_cluster -s
```

The **bootstrap_cluster -s** command will check the database status on all defined cluster members. If the check is successful, you receive a message that ends with the following line:

```
SUCCESS: All reachable cluster members reporting Primary database status
```

To forcibly restart the database on each cluster member, enter the following command:

```
bootstrap_cluster -bf
```

This can be used if some of the cluster members show as `Starting` following the **bootstrap_cluster -s** command, but you know that the entire cluster is down.
If all the cluster members show as `STOPPED`, you can run the following command to find the machine with the most up-to-date database, and use it to bootstrap the cluster:

```
bootstrap_cluster -b
```

After this machine has bootstrapped, it will start the database on the other machines consecutively so that they join the cluster.

- **background_sync_certs**
  By using the `background_sync_certs` command, you can generate or copy SLL certificates that are used for background sync with API Connect.
- **backup_devportal**
  By running the `backup_devportal` command, you can back up your Developer Portal server.
- **backup_site**
  By running the `backup_site` command, you can back up a single Developer Portal site.
- **bootstrap_cluster**
  By running the `bootstrap_cluster` command, you can boostrap the database processes on your cluster.
- **clear_site_cache**
  By running the `clear_site_cache` command, you can clear the site cache for a Drupal site.
- **clone_site**
  By running the `clone_site` command, you can clone your Developer Portal site.
- **cluster_members**
  By running the `cluster_members` command, you can list the IP addresses of the servers that are members of the Developer Portal clusters.
- **dbstatus**
  By running the `dbstatus` command, you can see the status of a Developer Portal database.
- **delete_platform**
  By running the `delete_platform`, you can delete a platform.
- **delete_site**
  By running the `delete_site` command, you can delete a Developer Portal site.
- **dhparam**
  By running the `dhparam` command, you can generate a Diffie-Hellman Parameter that can be used during the connection to the web server.
- **download_apim_cert**
  By running the `download_apim_cert` command, you can download the TLS certificate from the configured IBM® API Connect server.
- **gcache_size**
  By running the `gcache_size` command, you can set the Galera cache size.
- **generate_logs**
  By running the `generate_logs` command, you can generate the logs.
- **get_client_id**
  By running the `get_client_id` command, you can returns the catalog client id for this Developer Portal site.
- **list_platforms**
  By running the `list_platforms` command, you can list the platforms that are associated with your nodes.
- **list_sites**
  By running the `list_sites` command, you can list the Developer Portal sites that are either installed or being installed.
- **mysql_certs**
  By running the `mysql_certs` command, you can generate or copy the TLS certificates that are used to encrypt the database traffic.
- **new_certs**
  By running the `new_certs` command, you can create a TLS certificate and key for the nginx server, or the restservice.

- php_max_memory
  By running the `php_max_memory` command, you can set the maximum memory that a php process can consume to new_memory_limit_in_mb.
- queue_run
  By running the `queue_run` command, you can enable or disable the running of queued `site_action` tasks.
- V5.0.8 + reset_locked_host
  From API Connect Version 5.0.8.5 onwards, you can run the `reset_locked_host` command to show you the IP addresses of all the clients that have made failed login attempts, on a per site basis. You can then clear the failed login attempts from specific or all IP addresses.
- reset_locked_user
  By running the `reset_locked_user` command, you can show the number of failed log in attempts per site and user, and can reset the failed login counter per site for specific or all users.
- V5.0.4 and earlier restore_devportal
  By running the `restore_devportal` command, you can restore your Developer Portal server.
- restore_site
  By running the `restore_site` command, you can restore a single Developer Portal site backup to its original UUID and URL.
- run_bg_sync
  By running the `run_bg_sync` command, you can run background sync for a given Developer Portal site.
- set_apim_cert
  By running the `set_apim_cert` command, you can configure trust between the Developer Portal site and API Manager.
- set_apim_host
  By running the `set_apim_host` command, you can pair the Developer Portal site with API Manager, and create a new default TLS certificate.
- set_cluster_members
  By running the `set_cluster_members` command, you can configure the members in a cluster.
- set_devportal_cert
  By running the `set_devportal_cert` command, you can set the certificate for the Developer Portal.
- set_devportal_key
  By running the `set_devportal_key` command, you can set the key for the Developer Portal.
- set_ntp
  By running the `set_ntp` command, you can configure the ntp server.
- set_smtp
  By running the `set_smtp` command, you can set the smtp server.
- set_timezone
  By running the `set_timezone` command, you can set the php time zone.
- show_queue
  By running the `show_queue` command, you can show the queue.
- show_queue_executors
  By running the `show_queue_executors` command, you can identify the queue of executors.
- show_queue_history
  By running the `show_queue_history` command, you can show the history of the portal.task and portal.lock tables.
- site_action
  By running the `site_action` command, you can see the most recent site actions.
- site_login_link
  By running the `site_login_link` command, you can display a one time login link for the admin account for the site specified by orgid.envid.
- site_maintenance
  By running the `site_maintenance` command, you can enable or disable maintenance mode for the Developer Portal site that is specified.
- site_template
  By running the `site_template` command, you can specify `platform_id_or_dir` to create a Developer Portal site template for that platform.
- V5.0.5 + start_db
  By running the `start_db` command, you can start the database for a Developer Portal server.
- status
  By running the `status` command, you can show the status of the system.
- V5.0.5 + stop_db
  By running the `stop_db` command, you can stop the database for a Developer Portal server.
- timestamps
  By running the `timestamps` command, you can see the timestamp for your machine.
- upgrade_devportal
  By running the `upgrade_devportal` command, you can upgrade your Developer Portal platform.
- upgrade_site
  By running the `upgrade_site` command, you can upgrade your Developer Portal site.
- upgrade_status
  By running the `upgrade_status` command, you can see the status of pending and completed site upgrades.

- **upload_max_size**
  By running the `upload_max_size` command, you can set the maximum file upload size for nginx and php to new_upload_size_in_mb.
- **version**
  By running the `version` command, you can see your system and distribution version.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

background_sync_certs

By using the `background_sync_certs` command, you can generate or copy SLL certificates that are used for background sync with API Connect.

The following help text is displayed when you run the command followed by `-h`:

```
Usage: background_sync_certs [-hgsf] [-c ip/host]

Generates or copies the SSL certs used for background sync with API Connect.

 -g Generate a new CA cert for this cluster and a new server cert for this node.
 -c copy the already generated CA cert for this cluster and generate a new server cert for this node.
 -s Just generate new server certs for this node.
 -f force copy and/or regeneration of certificates.
 -h Show this help information.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

backup_devportal

By running the `backup_devportal` command, you can back up your Developer Portal server.

Running the `backup_devportal` command backs up the following information:

- All of the cluster configuration, including database configuration, the IP addresses of the cluster members, and the certificates that are used to secure communications. It includes the changes made by the following commands:

  ```
  set_apim_host
  set_cluster_members
  download_apim_cert
  set_apim_cert
  gcache_size
  pcweight
  set_devportal_cert
  set_devportal_key
  set_hostname
  ```

- All of your Developer Portal sites (unless the `-e` parameter is used).

The output from this command is a .tgz file (a gzipped tar file) that contains both the server configuration information and all of the backups of the Developer Portal sites. This file is stored in the /home/admin/backups directory. The backups of the Developer Portal sites are also stored separately in the /var/aegir/backups directory.

You can use the .tgz file that is stored in the /home/admin/backups directory to restore a Developer Portal server. For more information, see restore_devportal.

The following help text is displayed when you run the command followed by `-h`:
V5.0.6 and earlier

```
Usage: backup_devportal [-he]
```

```
 -h help
 -e exclude site backups
```

V5.0.7+

```
Usage: backup_devportal [-heiot]

Create a backup of the configuration and sites (unless -e is used) for this portal.
The resulting backup file can be restored to a freshly deployed OVA by running
restore_devportal.

The backup file will be created in /home/admin/backups.

-h help
-e exclude site backups
-o only run if no other cluster member is running backup_devportal
-t test if backup_devportal is already running on this server
-i include existing site backups but do not take new ones
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

backup_site

By running the **backup_site** command, you can back up a single Developer Portal site.

Running the **backup_site** command backs up all of the information that is needed to restore a single Developer Portal site. The output from this command is a .tgz file (a gzipped tar file) that is stored in the /var/aegir/backups directory.

You can use the .tgz file that is stored in the /var/aegir/backups directory to restore a single Developer Portal site. For more information, see restore_site.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: backup_site [-u uuid] uri
          -u use site uuid instead of uri
          -h show this help message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

bootstrap_cluster

By running the **bootstrap_cluster** command, you can boostrap the database processes on your cluster.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: bootstrap_cluster [-bshkapt] [-ilf]

Boostraps the database processes on this cluster.

 commands:
  -b bootstrap the database cluster using either any active cluster members or the most upto date
STOPPED cluster member as the master replica.
  -k kill the database process on any cluster members that are reporting Starting or Stopping.
  -s show the status of the database on each cluster member and exit.
  -a check the cluster database status and automatically bootstrap the cluster if all nodes are STOPPED
and reachable.
  -p Force all members into Primary database state. Use with -i to ignore unreachable members and move a
non-Primary cluster into Primary state.
  -t Stop the database on all cluster members.
  -h show this help information.

 modifiers:
  -i (used with -b, -p, -t or -s) ignore unreachable cluster members. WARNING: Only do this if you are
sure the other cluster members are permanently down. You could end up with a split brain otherwise.
  -l (used with -k and -f) also kill the database process on any cluster members that are reporting SST.
WARNING: Only do this if you are sure the SST has failed. Doing this while an SST is active can destroy
the database on that member, requiring restoration from a back
up.
  -f (used with -b) force a full shutdown and bootstrap of the cluster. This will refuse to run if any
nodes are performing an SST.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

clear_site_cache

By running the **`clear_site_cache`** command, you can clear the site cache for a Drupal site.

The following help text is displayed when you run the command followed by **`-h`**:

```
Clear the drupal site cache for site
    usage: clear_site_cache [-u uuid] uri
         -u use uuid instead of uri
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

clone_site

By running the **`clone_site`** command, you can clone your Developer Portal site.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: clone_site [-hf] original_site_uuid/url new_copy_of_site_uuid/url [new_copy_of_site_url]

Clones a site (referenced by url or uuid) into an new/existing site (referenced by url or uuid).

  When both sites exist: clone_site site1.uuid/url site2.uuid/url
  When the new sites does not exist: clone_site site1.uuid site2.uuid site2.url

 -h help
 -f if site already exists, it will be deleted before restoring it
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

cluster_members

By running the **`cluster_members`** command, you can list the IP addresses of the servers that are members of the Developer Portal clusters.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: cluster_members [-rh]

Lists the IP addresses of the servers which are members of the portal cluster.
```

```
 -r only show servers which are reported as currently active
 -h shows this message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

dbstatus

By running the **dbstatus** command, you can see the status of a Developer Portal database.

The following help text is displayed when you run the command followed by **-h**:

```
Shows database status

 -h shows this message
 -z always return 0 even if the db is down
 -r return an error if the internal revision does not match rev
 -s return the size of the db directory
```

Possible database statuses

The following table describes the possible database statuses that can be returned by the **dbstatus** command:

Table 1. Developer Portal database statuses

Status	Description
Standalone	The database is up and is not in a cluster.
Primary	The database is up and is in sync with the rest of the cluster
non-Primary	The database has temporarily stopped processing user transactions to avoid database divergence, because it lost contact with the rest of the cluster and determined that it was in the minority network segment. It will resume Primary state when it can contact the rest of the cluster.
SST-Starting-Up	The database is down and will get a State Snapshot Transfer (SST) because the database is very out of date.
SST-Applying-Logs	The database is applying the transactions that happened during the SST.
SST-Preparing-Logs	The database is preparing to apply the transactions that happened during the SST.
SST-X%	The database is down and a new copy of the database is being copied from another node. The percentage indicates the progress. When the copy completes, the database will go into Primary state.
Donor	The database is up and in Primary state but is also providing an SST for another node. When the SST has finished, the database will go into Primary state.
Preparing	The database has temporarily stopped processing user transactions while it applies an Incremental State Transfer (IST) to catch up with the rest of the cluster.
Stopping	The database is stopping.
STOPPED	The database is stopped.
Hung	The database is in an error state. It will be restarted shortly.

Glossary of terms

State Snapshot Transfer (SST)
: A copy of all transactions that the rest of the cluster has applied but that are not yet in the local database

Incremental State Transfer (IST)
: An entire copy of the database files, followed by all the transactions that were applied while the copy was occurring

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

delete_platform

By running the **delete_platform**, you can delete a platform.

The following help text is displayed when you run the command followed by **-h**:

```
Usage:  [-h] platform_dir_or_name
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

delete_site

By running the **delete_site** command, you can delete a Developer Portal site.

The following help text is displayed when you run the command followed by **-h**:

```
Usage (by uuid): delete_site [-f] [-h] orgid.envid
Usage (by url): delete_site [-f] [-h] -u my.url.com
```

Important: The recommended method to remove a Developer Portal site that has an associated catalog it to remove it through the API Manager UI. For more information on configuring the availability of the Developer Portal, see Creating and configuring Catalogs.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

dhparam

By running the **dhparam** command, you can generate a Diffie-Hellman Parameter that can be used during the connection to the web server.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: dhparam [-hfo] [-b bits]

Generates a 2048-bit Diffie-Hellman Parameter to use in the HTTPS handshake for
connections to the web server. If the parameter already exists then check to
see if the web server config needs to be updated and the web sever config
reloaded.

  -h display this help message.
  -o do NOT update other cluster members with this Diffie-Hellman parameter.
  -b use bits bits instead of 2048
  -f force regeneration even if the Diffie-Hellman parameter already exists.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

download_apim_cert

By running the **download_apim_cert** command, you can download the TLS certificate from the configured IBM® API Connect server.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: download_apim_cert

Downloads the SSL certificate from the configured IBM APIC server
and executes set_apim_cert to configures trust between this IBM Developer Portal and an IBM API Manager
machine.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

gcache_size

By running the **gcache_size** command, you can set the Galera cache size.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: gcache_size [-h] [new_gcache_size]

Sets the Galera Cache size. Append M or G for megabytes or gigabytes
Without any parameters then it will show the current Gelera cache size
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

generate_logs

By running the **generate_logs** command, you can generate the logs.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: generate_logs [-qh]

 -q Run in quiet mode
 -h Show this message
```

V5.0.5 and earlier In a clustered environment, you must run **generate_logs** on all cluster members and provide all of the resulting files to IBM support when raising a support ticket.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

get_client_id

By running the **get_client_id** command, you can returns the catalog client id for this Developer Portal site.

The following help text is displayed when you run the command followed by **-h**:

```
Get drupal client id for a site
  usage get_client_id [-u uuid] uri
   -u use the uuid instead of the uri
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

list_platforms

By running the **`list_platforms`** command, you can list the platforms that are associated with your nodes.

There is no help text that appears when you run the command followed by **`-h`**, as the command displays the platforms that are present.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

list_sites

By running the **`list_sites`** command, you can list the Developer Portal sites that are either installed or being installed.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: list_sites [-h] [-pin]
```

```
 -h Shows this information.
 -p Also lists the platform that each site is on.
 -n Does not show the current state of the site.
 -i Only shows installed sites.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

mysql_certs

By running the **`mysql_certs`** command, you can generate or copy the TLS certificates that are used to encrypt the database traffic.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: mysql_certs [-hg] [-c ip/host]
```

```
Generates or copies the SSL certs to encrypt the database traffic.
```

```
 -g Generate a new CA cert for this cluster and a new server cert for this node.
 -c copy the already generated CA cert for this cluster and generate a new server cert for this node.
 -s Just generate new server certs for this node.
 -l make all database communication SSL. Default is only replication traffic.
 -f force copy and/or regeneration of certificates.
 -h Show this help information.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

new_certs

By running the **`new_certs`** command, you can create a TLS certificate and key for the nginx server, or the restservice.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: new_certs [-h]

Creates an SSL certificate and key for the nginx server, or the restservice. Default is nginx.

 -r create certificate/key for the restservice instead of nginx.
 -i initial certificate/key creation. Do not overwrite after created for the first time.
 -h shows this message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

php_max_memory

By running the `php_max_memory` command, you can set the maximum memory that a php process can consume to new_memory_limit_in_mb.

The following help text is displayed when you run the command followed by `-h`:

```
Usage: php_memory_limit [-h] [new_memory_limit_in_mb]

Sets the maximum memory that a php process can consume to new_memory_limit_in_mb.
Without any parameters then it will show the current max memory
```

If you have any php memory issues while running background sync command, see the Background Sync section in Troubleshooting the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

queue_run

By running the `queue_run` command, you can enable or disable the running of queued `site_action` tasks.

The following help text is displayed when you run the command followed by `-h`:

```
Usage: queue_run [-hsedrqb]

Enables or disables the runnning of queued site_action tasks

 -s shows the status of the queue, including whether run_site_queue is running
 -b shows just the status of the boolean queue variable QUEUE_RUN_STATUS
 -e enables the queue
 -d disables the queue
 -r restarts the queue daemon, via cron
 -q quiesce the queue. Disable the queue and wait for all tasks to complete befre returning
 -h shows this message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

reset_locked_host

From API Connect Version 5.0.8.5 onwards, you can run the `reset_locked_host` command to show you the IP addresses of all the clients that have made failed login attempts, on a per site basis. You can then clear the failed login attempts from specific or all IP addresses.

The following help text is displayed when you run the command followed by `-h`:

```
Usage: reset_locked_host [-s] [-l [sitename]] [-r sitename hostIP/-a]

Shows the IPs of all clients that have made failed login attempts, on a per site basis.
Allows the clearing of failed login attempts from specific IPs (or all IPs)

  -s get a list of portal sites by url (run this first if you're not sure of the site name to use with -
l/-r options)
  -l show failed login count from all IPs per site, either for all sites, or for sitename if sitename is
provided.
  -r reset the failed login attempts to 0 for sitename and host IP. -a means all host IPs
```

Related reference

- reset_locked_user

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

reset_locked_user

By running the **`reset_locked_user`** command, you can show the number of failed log in attempts per site and user, and can reset the failed login counter per site for specific or all users.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: reset_locked_user [-s] [-l [sitename]] [-r sitenameaccountname/-a]
    Shows the failed login attempts per site and user and can reset the
    failed login countper site for a specific user or all users.

    -s list sites by url
    -l show all locked accounts per site, either for all sites, or for sitename if
    sitename is provided.
    -r reset the failed login attempts to 0 for sitename and
    accountname.
    -a means all sites or all accounts.
```

Related reference

- V5.0.8 + reset_locked_host

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

restore_devportal

By running the **`restore_devportal`** command, you can restore your Developer Portal server.

Before you can restore your Developer Portal server, you must deploy a new Developer Portal OVA file. For information about deploying an OVA file, see Deploying the Developer Portal OVA file.
After a new OVA file is deployed, you can run the **`restore_devportal`** command to restore the Developer Portal server that was backed up by using the **`backup_devportal`** command.

The following help text is displayed when you run the **`restore_devportal`** command followed by **`-h`**:

```
Usage: restore_devportal [-w] [-spcnf] [-u url] path.to.backup.file.tar.gz

The recommended way to execute this script is by using the wizard (-w) as
it gives more information on each step and can be used to restore just
specific sites and even specific dated backups of each site.
```

```
Using the portal restore wizard:
  -w path.to.backup.file.tar.gz launch the portal restore wizard

Restore Developer Portal Instance
  -h display this help information
  -u restore site with url
  -s restore all sites
  -f restore sites even if they already exist
  -c restore cluster configuration. Will re-run set_apim_host to pair with the manager node.
  -n restore node specific configuration. Will set the hostname and IP config.
  -p restore the Drupal platforms in /var/aegir/platforms.

If executed with just the file parameter then the default behaviour is the same as 
running restore_devportal -s -c -p file.tgz
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

restore_site

By running the `restore_site` command, you can restore a single Developer Portal site backup to its original UUID and URL.

You can run the `restore_site` command to restore the single Developer Portal site that was backed up by using the `backup_site` command.
The following help text is displayed when you run the command followed by `-h`:

```
Usage: restore_site [-hf] [-u new_url] [-i new_uuid] site_backup_absolute_path

Restores a site backup to its original UUID and URL. Using -i and/or -u the site
can be restored to a different UUID and/or URL.

 -h help
 -f if site already exists, it will be deleted before restoring it
 -u restore to new_url. Assigns the uuid of the existing site at new_url, unless -i is also used.
 -i restore to new_uuid. Assigns the url of the existing site at new_uuid, unless -u is also used.
 -c clean users (not admin), apis, applications and plans from the database after restoring.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

run_bg_sync

By running the `run_bg_sync` command, you can run background sync for a given Developer Portal site.

The following help text is displayed when you run the command followed by `-h`:

```
Restart background sync for specified site
    run_bg_sync [-u uuid] uri
    -u run on site with uuid
  -h display this message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_apim_cert

By running the **`set_apim_cert`** command, you can configure trust between the Developer Portal site and API Manager.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage (to set a certificate): cat apim_ssl.crt | set_apim_cert -s
Usage (to accept all certificates): set_apim_cert -i
Usage (to delete the current certificate): set_apim_cert -d

Configures trust between this IBM Developer Portal and an IBM API Manager machine.
Pipe the SSL certificate for the IBM API Manager Portal API into the <stdin> of this script to configure
trust.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_apim_host

By running the **`set_apim_host`** command, you can pair the Developer Portal site with API Manager, and create a new default TLS certificate.

CAUTION:
Changing the default Developer Portal APIs can introduce unexpected effects that are not documented.
The following help text is displayed when you run the command followed by **`-h`**:

```
Usage (standalone apim): cat apic_public_key | set_apim_host [-h] devportal_fqdn apic_fqdn[:port]
Usage (cluster of apim): cat apic_public_key | set_apim_host [-h] devportal_fqdn
apic_balancer_fqdn[:port]

Pairs the IBM Developer Portal with an IBM API Manager and creates a new default SSL certificate.

To setup an IBM Developer Portal connected to a single IBM API Manager:

  devportal_fqdn is the fully qualified domain name of this developer portal machine.
  apic_fqdn is the fully qualified domain name or ip address of the IBM API Manager.
    The [:port] defaults to 443 or you can supply one by appending :port_number e.g.  :6871

To setup an IBM Developer Portal connected to IBM API Manager cluster:

  devportal_fqdn is the fully qualified domain name of this developer portal machine.
  apic_balancer_fqdn is the fully qualified domain name or ip address of the load balancer for IBM API
Manager.
    The [:port] defaults to :443 or you can supply one by appending :port_number e.g. :6871

The public ssh key (apic_public_key) generated by IBM API Connect to allow control of this IBM Developer
Portal should be present on <stdin> when executing this script.

-h shows usage information
```

Note: If a **`:port_number`** is entered for this command, it must match the port number that is configured for the Developer Portal APIs in the TLS Profiles section of the Settings page of the Cloud Manager UI.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_cluster_members

By running the **`set_cluster_members`** command, you can configure the members in a cluster.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage (creating a cluster): set_cluster_members -c
Usage (adding a new member): set_cluster_members hostname/ip_of_existing_cluster_member
Usage (runtime change): set_cluster_members -r hostname/ip_1 [hostname/ip_2 hostname/ip_3 ...]
Usage (de-clustering): set_cluster_members -d
```

```
 -c will create a new cluster. This only needs to be run against the 1st member creating the cluster and
not the others
 -r indicates a transient change in cluster membership based on runtime monitoring
 -d de-clusters this node from any others and turns it into a standalone machine.
 -s shortcuts 'creating a cluster' or 'adding a new member' if they have run before.
V5.0.8 + -v re-cluster just with currently visible servers. Prompts for confirmation.
V5.0.8 + -1 modifies the behavior of -v to go standalone if no other servers are visible.
```

Note: If you run the `set_cluster_members` command to configure the clustering, the command will check whether the IP addresses that you have provided are Developer Portal nodes, and that they are at the same version of system code.
If you run the command with a list of IP addresses of other nodes to cluster with, the command checks the system version and checks whether at least one other node on the list has had `set_cluster_members -c` run on it.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_devportal_cert

By running the `set_devportal_cert` command, you can set the certificate for the Developer Portal.

There is no help text when you run the command followed by `-h` because you are setting the certificate.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_devportal_key

By running the `set_devportal_key` command, you can set the key for the Developer Portal.

There is no help text when you run the command followed by `-h` because you are setting the key.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_ntp

By running the `set_ntp` command, you can configure the ntp server.

The following help text is displayed when you run the command followed by `-h`:

```
Usage (show the configured ntp servers): set_ntp -s
Usage (set the ntp servers): set_ntp server1 [server2 server3 ...]
Usage (reset the ntp servers the default): set_ntp -d
```

```
 -h show this help information
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_smtp

By running the **`set_smtp`** command, you can set the smtp server.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage set_smtp: SMTP_SERVER SMTP_PORT [SMTP_USER SMTP_PASSWORD]
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set_timezone

By running the **`set_timezone`** command, you can set the php time zone.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: set_timezone [-h] [timezone]

Sets the php timezone to timezone.
Without any parameters then it will show the current php timezone
```

Warning: Do not change the time zone of the Developer Portal appliance. The Developer Portal appliance must be set to the default time zone of `Etc/UTC`. Do not set the `TZ` environment variable for any users, or run the **`dpkg-reconfigure tzdata`** command to set the global time zone. Setting the time zone to anything other than `Etc/UTC` can lead to unpredictable behavior in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

show_queue

By running the **`show_queue`** command, you can show the queue.

There is no help text when you run the command followed by **`-h`** because you are showing the queue.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

show_queue_executors

By running the **`show_queue_executors`** command, you can identify the queue of executors.

The following help text is displayed when you run the command followed by **`-h`**:

```
QUEUE_RUN_STATUS is true
UID        PID  PPID  C STIME TTY          TIME CMD
admin    17700     1  0 09:41 ?        00:00:29 /bin/bash /home/admin/bin/run_site_queue
admin    17793 17700  0 09:41 ?        00:00:00   /bin/bash /home/admin/bin/run_site_queue

Currently executing priority queue tasks (0):
UID        PID  PPID  C STIME TTY          TIME CMD

Currently executing NON priority queue tasks (0):
UID        PID  PPID  C STIME TTY          TIME CMD

Current locks held (0):

Next executable tasks on the queue (total queue length: 0)
```

```
Current date: current_date_and_time
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

show_queue_history

By running the **show_queue_history** command, you can show the history of the portal.task and portal.lock tables.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: show_queue_history [-ha] [-q qid]

Shows the history of the portal.task and portal.lock tables

 -a show all columns. Default is a columns related to performance.
 -q [qid] show just for this qid
 -f only show tasks that completely failed. Tasks that completed after some retry attempts are NOT
shown.
 -n only show tasks that has a non-zero return code. Tasks that completed after some retry attempts are
shown.
 -h show this usage message.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

site_action

By running the **site_action** command, you can see the most recent site actions.

There is no help text when you run the command followed by **-h** because you are showing the site actions.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

site_login_link

By running the **site_login_link** command, you can display a one time login link for the admin account for the site specified by orgid.envid.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: site_login_link [-f] orgid.envid/url

Displays a one time login link for the admin account for the site specified by orgid.envid or url

 orgid.envid/url is the unique identifier/url for the site.

 -f forces generation of a one time login link even if the site is disabled
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

site_maintenance

By running the **`site_maintenance`** command, you can enable or disable maintenance mode for the Developer Portal site that is specified.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: site_maintenance [-desh] site_uuid/site_url

Enables or disables maintenance mode for the site specified

 -e Enable site maintenance mode.
 -d Disable site maintenance mode.
 -s Show the current site maintenance mode.
 -h Show this help information.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

site_template

By running the **`site_template`** command, you can specify **`platform_id_or_dir`** to create a Developer Portal site template for that platform.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: site_template [-fh] [platform_id_or_dir]

Specify platform_id_or_dir to create a site template for that platform.

 -h show this help.
 -f force recreation of a site template.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

start_db

By running the **`start_db`** command, you can start the database for a Developer Portal server.

Note: If your Developer Portal server is in a cluster, there must be another server in the cluster that is running for the **`start_db`** command to work.
The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: start_db [-febh]

Starts the database on the local host.

 commands:
  -h show this usage info.
  -f Do not start the database if the stop_db flag exists.
  -b Bootstrap a new database cluster instead of starting the database.
  -e Expedite mode, skips the cluster member check
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

status

By running the **status** command, you can show the status of the system.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: status [-h]
```

```
Shows system status
```

```
 -h shows this message
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

stop_db

By running the **stop_db** command, you can stop the database for a Developer Portal server.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: stop_db [-h] [-q] [-u] [-n node]
```

```
Stops the database on a specified node.
```

```
If a node (-n) is not specified the local node is assumed.
```

```
 commands:
  -h show this usage.
  -n Specifies the node (hostname, ip address, or 'localhost') to stop the database on
  -q Skip the quiesce of the site queue
  -u Assume that the database is hung
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

timestamps

By running the **timestamps** command, you can see the timestamp for your machine.

There is no help text when you run the command followed by **-h** because you are seeing the timestamp.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

upgrade_devportal

By running the **upgrade_devportal** command, you can upgrade your Developer Portal platform.

The following help text is displayed when you run the command followed by **-h**:

```
Usage: upgrade_devportal [-dm -ui ibm_apim_platform.tgz] [-p platform] [-s site] [-abant]

 -d Upgrade Drupal to the latest available stable 7.x version
 -m Upgrade all contributed Drupal modules to the latest available stable version
 -i Upgrade the IBM Drupal modules to the versions in file ibm_apim_platform.tgz
 -u Upgrade all. Same as supplying -dmi
 -p Upgrade to the platform identified by platform.
 -s Upgrade just the site identified by site. Useful for testing.
 -b Just build the upgraded platform, do not upgrade any sites to it.
 -a Upgrades all sites to the new platform and configures the platform as the default for new sites.
 -n Sets the platform the default for new sites.
 -t Do not try to patch modules.

Single command if no sites exist:

 1  (If just updating Drupal version and module versions): upgrade_devportal -ndm
 or (If updating Drupal version, module versions and IBM modules): upgrade_devportal -nu 
ibm_apim_devportal-7.x-4.X.X.X.tgz

Recommended workflow when sites exist:

 1  (Build, if just updating Drupal version and module versions): upgrade_devportal -bdm
 or (Build, if updating Drupal version, module versions and IBM modules): upgrade_devportal -bu 
ibm_apim_devportal-7.x-4.X.X.X.tgz
 2  (Create a test site): create_site site_name site_url email
 3  (Test a site upgrade to the new platform): upgrade_devportal -p new_platform_name -s site_name
 4  (Test the test site): Login to https://site_url and check the site functions properly
 5  (Delete the test site): delete_site site_name
 6  (Update all sites to new platform and set as default): upgrade_devportal -p new_platform_name -a
```

Note: For the **`delete_site site_name`** command, *site_name* is the **`site uuid`** or **`site -u site_url`**.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

upgrade_site

By running the **`upgrade_site`** command, you can upgrade your Developer Portal site.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: upgrade_site orgid.envid platform
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

upgrade_status

By running the **`upgrade_status`** command, you can see the status of pending and completed site upgrades.

The following help text is displayed when you run the command followed by **`-h`**:

```
Usage: upgrade_status [-whc] [-p platform] -d [YYYY-MM-DD hh:mm:ss]

Show the status of pending and completed site upgrades.

 -w Wait for all upgrades to complete.
 -p Just show the status for upgrades to this platform.
 -c Just show the count of sites in each state.
 -d Only show tasks that finished after the provided date/time stamp
 -h Show this help information.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

upload_max_size

By running the **upload_max_size** command, you can set the maximum file upload size for nginx and php to new_upload_size_in_mb.

The following help text is displayed when you run the command followed by **-h**:

```
Usage upload_max_size [-h] [new_upload_size_in_mb]
```

```
Sets the maximum file upload size for nginx and php to new_upload_size_in_mb.
Without any parameters then it will show the current max upload size
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

version

By running the **version** command, you can see your system and distribution version.

The following help text is displayed when you run the command followed by **-h**:

```
System version: your_system_version
Distribution version: your_distribution_version
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Enabling database encryption for the Developer Portal

You can enable encryption for the communication data for databases between nodes in your Developer Portal cluster.

About this task

By following the steps that are required to enable database encryption, an outage occurs for any Developer Portal sites that are hosted on the cluster. The outage occurs because all databases must be brought down initially to successfully enable encryption. The system automatically restarts the database, and in turn the Developer Portal sites, after encryption is enabled.

Procedure

1. Run the following command on any node in your cluster:

   ```
   mysql_certs -gl
   ```

   Note: After running the command, you must log into the node again as the group membership has changed for the administrator account.
2. Run the following command on the other nodes in your cluster:

   ```
   mysql_certs -c IP_address_of_first_node
   ```

   Note: After running the command, you must log into the nodes again as the group membership has changed for the administrator account.
3. Run the following command on all of the nodes:

```
queue_run -q && sudo service mysql stop
```

Results

After a few minutes the nodes restart, and the communication channels for their databases are encrypted.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing IBM API Connect

IBM® API Connect can be installed by using VMware, or Citrix XenServer.

Before you begin

Note: These installation instructions are for installing IBM API Connect servers on-premises. If you want to install the IBM API Connect developer toolkit to develop APIs and LoopBack applications, see Installing the toolkit.

- Ensure that you have the minimum hardware and software requirements installed. For more information, see IBM API Connect Version 5.0 requirements.
- Decide how many different servers you want to install. For more information, see Planning your cloud.
- Each server that you configure in IBM API Connect must already be defined in your network and must have a separate IP address.
- Depending on the combination of physical and virtual appliances you are using, ensure that all API Connect management appliances in your on-premises installation are initialized and that all gateway (DataPower®) servers are running. Check that all of the servers are powered on and active in the network. Then, log in to the API Connect console to connect to these servers and define an API Connect provider organization.

About this task

The required steps of installing your server depend on the type of server that you are installing.

At a minimum, you must initialize API Connect on two specific server types. Then, by using the API Connect cloud console, you must add one of each of the following API Connect server types:

- Management (the first Management server is added automatically)
- Gateway
- Developer Portal

Important: Configure Network Time Protocol (NTP) on every Management, Gateway, and Developer Portal server to ensure that there is a consistent time view. Any differences in the clocks across all of the servers can cause problems.
For virtual servers, each server type requires a separate OVA file, which are created during the installation process. Therefore, to install API Connect requires two different OVA files:

- For the Management server, *product_version*-APIConnect-ManagementAppliance-*date_time_unique_id*.ova
- For the Gateway server, refer to the supplied product details

Note: When you obtain an interim fix from the IBM Fix Central support site, the stem of the file name also includes the build identifier.
Depending on the software that you are using to install API Connect within your enterprise, select the relevant installation topics.

Restriction: An API Connect management appliance accepts only Secure Shell (SSH) connections using secure ciphers. Insecure algorithms such as SHA1 and MD5 are not accepted. If a tool that uses insecure ciphers is used to connect to an API Connect management appliance, the appliance refuses the connections and the `Server unexpectedly closed network connection` message is displayed.

Procedure

1. For each server, select a platform (physical appliance or VMware, or Citrix XenServer) and follow the installation instructions from the topics that are provided at the end of this topic.
2. For each server, configure the network to use either Dynamic Host Configuration Protocol (DHCP) or static addressing:
    - Configuring to use DHCP addressing

- Configuring to use static addressing

Important: Using DHCP addressing on a Gateway server (DataPower appliance) is incompatible with using Gateway cluster load balancing.

After API Connect is deployed and configured, the user interfaces are accessed by using the following URL formats:

- Cloud console:

  ```
  https://<management_hostname_or_proxy>.<domain>/cmc
  ```

- API Manager:

  ```
  https://<management_hostname_or_proxy>.<domain>/apim
  ```

- Within an organization, for a specific Catalog Developer Portal, you can configure the following example URL format:

  ```
  https://<management_hostname_or_proxy>.<domain>/<organization_name>/<catalog_name>
  ```

Within an organization, API traffic for a specific Catalog:

- If you are using Dynamic DNS:

  ```
  https://<CatalogName>.<OrganizationName>.<GatewayHostname or ExternalLoadBalancerIPaddress>.
  <domain>
  ```

- If you are not using Dynamic DNS:

  ```
  https://<GatewayHostname or ExternalLoadBalancerIPaddress>.
  <domain>/<OrganizationName>/<CatalogName>
  ```

Note: Dynamic DNS is an option that you can specify when you configure your API Connect on-premises cloud by using the Cloud Console. For more information, see **Selecting the DNS scheme**.

What to do next

To learn how to install and use the Developer Portal, see Deploying the Developer Portal OVA file, Installing the Developer Portal, and Developer Portal: discover and use APIs.

- **Installing your toolkit**
- **Installing the Management virtual server**
  You can create a Management server by deploying the Management OVA template.
- V5.0.7+ **Installing a containerized runtime environment**
  Install a containerized runtime environment to provide a lightweight, efficient deployment location for your APIs and applications.
- **Installing API Connect collective**
  V5.0.6 and earlier Install the API Connect collective Member host and the API Connect collective controller. Add the API Connect collective to the Cloud Manager.
- **Installing IBM HTTP Server**
  Install IBM HTTP Server as a load balancer in front of the Micro Gateway instances.
- **Installing the Micro Gateway for the Professional and Enterprise offering**
  Install the Micro Gateway and add it to the Catalog.
- **Using physical DataPower appliances as Gateway Servers**
  A physical appliance can serve as a Gateway server.
- **Using virtual DataPower appliances as Gateway Servers**
  A virtual DataPower appliance can serve as a Gateway server. To use a virtual appliance as a Gateway server, you must complete a number of configuration tasks. You then add the appliance to your API Connect topology by using the cloud console.
- **Deploying the Developer Portal OVA file**
  You create a Developer Portal node by deploying the Developer Portal OVA template. After you have deployed the Developer Portal OVA template, you can install the Developer Portal.
- **Installing the Developer Portal**
  In order to use the Developer Portal, you need to complete a one-time installation and setup process.
- **Backing up and restoring the Developer Portal**
  You can back up individual Developer Portal sites, or all of the sites and Developer Portal configuration settings. You can then restore backed up sites and portal configurations. You can also take snapshots of your Developer Portal servers.
- V5.0.2+ **Configuring your DataPower Gateway and API Connect collective controller to communicate**
  To be able to use the DataPower Gateway to invoke applications, you must configure TLS certificates on the collective controller and on the DataPower appliance so that they can communicate.
- **Installing API Connect Professional on-premises with the Micro Gateway**
  Install and configure the components for the API Connect Professional offering with the Micro Gateway in an on-premises environment.
- **Configuring IBM API Connect to use a DataPower Tenant**
  You can configure IBM API Connect to work with a tenant on a physical Type 8436 DataPower Gateway.

Installing the toolkit

You can install the toolkit either from npm or from a Management server in your IBM® API Connect cloud.

Before you begin

Features of the toolkit include the default ability for local testing of APIs with a DataPower® Gateway Docker container. When Docker and Docker Compose are present and functioning normally on your system, the DataPower image is downloaded from Docker Hub. For more information, see: Testing APIs with the IBM DataPower Gateway.

The following steps create a software environment that is ready to install the IBM API Connect toolkit.

Note: On Windows, use the Windows command shell run as an administrator to enter Node or npm commands, instead of Powershell or Cygwin (the Windows bash shell emulator).

1. Install Community Node.js distribution version 10, or a higher 10.x.x version. For details of supported Node.js versions, see Detailed System Requirements, then click the Prerequisites tab.
2. Ensure that `node` is in your PATH.
3. Use the `npm -v` command to check the version of npm and ensure that it shows 10.x.x, or higher.

About this task

You install the toolkit by using the `npm` command that is installed as part of Node.js. Installing the toolkit installs:

- API Connect command-line tool, `apic`.
- API Connect API Designer visual tool.
- API Connect Micro Gateway.

You can also uninstall the toolkit and display the version number of the currently installed toolkit.
Use the version of the toolkit that corresponds to the version of your API Connect Management Server. To find the appropriate toolkit version, use the `npm dist-tag ls apiconnect` command. For example:

```
$ npm dist-tag ls apiconnect
1.0.3.0: 1.0.3
apic-v5.0.1.0: 2.0.18
apic-v5.0.2.1: 2.1.19
apic-v5.0.3.0-iFix1: 2.2.12
apic-v5.0.3.0-iFix2: 2.2.14
apic-v5.0.3.0-iFix4: 2.2.17
apic-v5.0.3.0: 2.2.9
apic-v5.0.4.0-iFix2: 2.3.10
apic-v5.0.4.0: 2.3.6
apic-v5.0.5.0: 2.4.11
apic-v5.0.6.0: 2.5.8
apic-v5.0.6.1: 2.5.17
apic-v5.0.6.2: 2.5.21
apic-v5.0.6.3: 2.5.33
apic-v5.0.6.4: 2.5.40
apic-v5.0.6.5: 2.5.52
apic-v5.0.6.6: 2.5.80
apic-v5.0.7.0: 2.6.2
apic-v5.0.7.1: 2.6.55
apic-v5.0.7.2: 2.6.71
apic-v5.0.8.0: 2.7.30
apic-v5.0.8.1: 2.7.62
apic-v5.0.8.2: 2.7.111
apic-v5.0.8.3: 2.7.209
apic-v5.0.8.4-iFix: 2.8.39
apic-v5.0.8.4: 2.8.14
apic-v5.0.8.5: 3.0.17
apic-v5.0.8.6: 4.0.x
latest: 4.0.x
v5.0.0.1: 1.0.3
```

Then, to install a specific version of toolkit use the command `npm install apiconnect@<version>`. For example, if you have API Connect v5.0.6.5, use the command:

```
npm install apiconnect@3.0.17
```

Procedure

Note: Do not use the npm configuration setting `engine-strict` (or use the `--engine-strict` option) since it will prevent installation from completing.
During installation, you may see errors from the `node-gyp` module, but because these errors are from an optional dependency, the installation should finish successfully.

You may also see npm warnings of `Possible EventEmitter memory leak detected`, but these are spurious and do not indicate a memory leak or any other issue.

To install the toolkit, complete the following steps:

- If you are not using trusted certificates, enter the following command:

  ```
  npm config -g set strict-ssl false
  ```

- If you are using a proxy server, enter the following commands:

  ```
  npm config set proxy http://proxy_address:port
  npm config set https-proxy http://proxy_address:port
  ```

  where *proxy_address* is the host name or IP address of the proxy server, and *port* is the port number.
- You are going to install the toolkit globally (**-g**), because it includes a command line interface. You can use **npm config get prefix -g** to learn the location of the global installation directory. Ensure that you are installing with sufficient permissions to write system files.
  Install the toolkit in either of the following ways:
  - To install the toolkit from npm, enter the following command:

    ```
    npm install -g apiconnect
    ```

  - To install the toolkit from a Management server in your IBM API Connect cloud, enter the following command:

    ```
    npm install -g --unsafe-perm https://appliance/packages/apiconnect
    ```

    Where *appliance* is the host name or IP address of the Management server appliance.
  - To install the toolkit locally from the toolkit tarball, download the tarball .tgz file from IBM Fix Central, then enter the following command:

    ```
    npm  i -g apiconnect-version.tgz
    ```

    or if you installing from root, enter the following command:

    ```
    npm  i -g apiconnect-version.tgz --unsafe-perm
    ```

- To display the version number of the currently installed toolkit, enter the following command:

  ```
  apic -v
  ```

What to do next

Obtain an IBM Cloud API Key for authenticating with the toolkit.

1. Obtain the API Key by completing the following steps:
   a. Browse to the IBM Cloud Identity and Access Management page.
   b. In the navigation list, click IBM Cloud API Keys.
   c. On the IBM Cloud API Keys page, click Create an IBM Cloud API key.
   d. In the Create API key dialog box, provide a Name and a Description for your new key, and then click Create.
   e. In the Create API key message box, either Copy or Download the key to make sure you save a copy. If you save the file, it is named apiKey.json.
   Attention: Do not navigate away from the page until you have saved the key. Once you lose this message, you cannot obtain a copy of the key (in that case, delete the key and create a new one).
2. Log in to the toolkit with the new API Key and the following command:

   ```
   apic login --server=Cloud.apiconnect.ibmcloud.com --apikey=Your_new_api_key
   ```

   In the command, ***`Cloud`*** is the hosted instance of API Connect you are connecting to; for example, `us` for US South, or `eu-de` for Frankfurt. In Reserved Instance, this is the host name that you use to access API Manager (begins with `mgr-`).

Uninstalling the toolkit

Before you begin

Before uninstalling the toolkit, stop any apps that are running locally by entering the command:

```
apic stop --all
```

Note: Configuration settings for the toolkit are stored in the `home_dir\.apiconnect` directory, where *home_dir* is the home directory of the user account under which the toolkit was installed. When uninstalling, you have the option to either keep or delete the configuration settings. The default behavior is to delete the `home_dir\.apiconnect` directory. Use the `--no-config-clear` command to preserve the configuration settings.

Procedure

1. Uninstall using npm:

   ```
   npm uninstall -g apiconnect
   ```

   By default, this command removes the toolkit configuration settings, as described above. To keep toolkit configuration settings, instead enter the command:

   ```
   npm uninstall -g apiconnect --no-config-clear
   ```

   To uninstall the toolkit, clear your npm cache, and remove the toolkit configuration, instead enter the command:

   ```
   npm uninstall -g apiconnect
   ```

   To uninstall the toolkit, clear your npm cache, and keep the toolkit configuration, instead enter the command:

   ```
   npm uninstall -g apiconnect --no-config-clear
   ```

   To verify that the cache is consistent after uninstalling the toolkit, enter the following command:

   ```
   npm cache verify
   ```

2. On Windows, delete all files whose names begin with `npm-` in `C:\Users\username\AppData\Local\Temp`

Updating your toolkit installation

Before you begin

Before updating your toolkit installation, be sure to stop any apps that are running locally by entering the command:

```
apic stop --all
```

Procedure

1. Uninstall the toolkit, as described in Uninstalling the toolkit.
2. Reinstall the toolkit by entering this command:

   ```
   npm install -g apiconnect
   ```

What to do next

If you previously used a username and password to authenticate in the API Connect Developer Toolkit, then you must start using API Key authentication instead.

1. Obtain the API Key by completing the following steps:
   a. Browse to the IBM Cloud Identity and Access Management page.
   b. In the navigation list, click IBM Cloud API Keys.
   c. On the IBM Cloud API Keys page, click Create an IBM Cloud API key.
   d. In the Create API key dialog box, provide a Name and a Description for your new key, and then click Create.
   e. In the Create API key message box, either Copy or Download the key to make sure you save a copy. If you save the file, it is named apiKey.json.
      Attention: Do not navigate away from the page until you have saved the key. Once you lose this message, you cannot obtain a copy of the key (in that case, delete the key and create a new one).
2. Log in to the toolkit with the new API Key and the following command:

   ```
   apic login --server=Cloud.apiconnect.ibmcloud.com --apikey=Your_new_api_key
   ```

In the command, ***Cloud*** is the hosted instance of API Connect you are connecting to; for example, **us** for US South, or **eu-de** for Frankfurt. In Reserved Instance, this is the host name that you use to access API Manager (begins with **mgr-**).

Related concepts

- Developer toolkit tutorials

Related information

- Troubleshooting the Micro Gateway

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing the Management virtual server

You can create a Management server by deploying the Management OVA template.

Before you begin

Ensure that you have the minimum hardware and software requirements installed. For more information, see IBM API Connect Version 5.0 requirements.

Plan your IBM® API Connect cloud to ensure you have determined your Management server configuration requirements. For more information, see Planning your cloud.

About this task

You must deploy the IBM API Connect OVA template to create each Management virtual server that you want in your cloud.

There are two separate passwords in an API Connect cloud. The Cloud Manager password is required to administer all Management and Gateway servers. Each server has a separate CLI password required to log in through a Secure Shell (SSH) to perform specific administrative actions for only that server.

Note:

- The following steps apply to VMware only. Depending on the VMware version and features available, that you are using, some of the steps might vary. For example, you might not be able to change the user name and password.
- VMware vCenter 6.5 has a known issue with unzipping compressed distribution files. See https://www.ibm.com/support/docview.wss?uid=ibm10887129.

Important: The Management virtual server is initially configured with a default password of `!n0r1t5@C`. For security reasons, complete one of the following options to change the default password:

- 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 both the Cloud Manager and Management virtual server command line interface (CLI) are modified. Use the new password when logging into either the Cloud Manager or CLI. Any changes you make to the user name during deployment affect only Cloud Manager credentials and not the CLI credentials. You cannot modify the admin user name for the CLI.
- After deployment, log in to the CLI for each virtual server (appliance) and run the command to change the default password for that specific virtual server. The CLI command is **auth set user admin**. This option for changing the CLI password does not change the password for the CMC.

Note: 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, please ensure that you have typed the password correctly.

Procedure

1. For each Management server, configure the network to use either Dynamic Host Configuration Protocol (DHCP) or static addressing:
    - Configuring to use DHCP addressing

- [Configuring to use static addressing]

2. Download a local copy of the IBM API Connect OVA file.
   Important: Download the OVA file to a storage device that is accessible by the virtual infrastructure and where the software is to be installed.
3. In the VMware Infrastructure Client navigation pane, select the virtual machine on which to install the IBM API Connect software.
   Note: If you are deploying multiple clouds on numerous machines, you can save configuration time using the OVF Tool on the command line. For more information, see [Network configuration using the OVF Tool].
4. If your cloud configuration is standard, (for example, utilizing no more than two machines), select File > Deploy OVF Template.
   The Deploy OVF Template wizard is displayed.
5. Select the relevant location to Deploy from a file or URL.
   Choose the file location where you downloaded the OVA file. You can enter a URL if you want to download the OVA file from the Internet.

   Click Next.

   The Verify OVF template details page opens.
6. Verify the template details and click Next.
7. Specify the name and location for the deployed template. Click Next.
8. Select a deployment configuration option.
   You can choose one of the following options.
   - Standard - 4 CPU, 8GB RAM
   - Enhanced - 8 CPU, 16GB RAM

   Note: You can modify your virtual machine settings to increase the number of CPUs and memory size after you complete the installation.
   Increase the number of CPUs and the memory size for improved performance.

   Click Next.

   The Host/Cluster page is displayed.
9. Select the host or cluster on which to run the deployed template. Click Next.
10. Select a resource pool. Click Next.
11. Select the data store to which you want to store the virtual machine files, enter the available disk space in GB and select a disk format option. The minimum supported data disk size for a management server is 50GB. 300GB is recommended for production. Then, click Next.
    Important: The Hard disk 1 is where the IBM API Connect operating system is installed, and must be increased in size when upgrading to IBM API Connect Version 5.0 or later. However, you can increase the size of Hard disk 2 to accommodate the analytics data.
    If you are going to alter the size of Hard disk 2, the size must be modified in the settings of the deployed VM after deployment but before it is powered on for the first time. See [17]

12. Optional: Map the networks that are used in the OVF template to networks in your inventory. Click Next.
13. Optional: Provide the properties to help automate the initial cloud configuration. Any values that are omitted in the Properties panel can be entered later by using the Cloud Manager or the command-line interface (CLI):
    Important: When you create the first Management server, leave the Cloud IP address field blank. When you create another Management server, the process of adding this server to an existing cloud can be automated by entering the IP address of the existing cloud in this configuration panel. Alternatively, Management servers can be manually connected to an existing cloud by using the Cloud Manager for that cloud.
    a. Optional: Specify details for the Username, Password, and Email fields.
       When you specify a password, the password for both the Cloud Manager and individual server CLI is updated to the specified password. Future modifications to the password for the Cloud Manager or an individual server require you to log into each component separately to change either the Cloud Manager password or the CLI password for an individual server.

       Any user name specified here affects only the Cloud Manager user name. The default CLI user name, admin, cannot be modified. The password must consist of at least 8 characters and must contain at least three of the following types of characters:
       - Lowercase letters
       - Uppercase letters
       - Numbers
       - Special characters, for example, exclamation point (!)

       Attention: Take a note of the credentials as user names are case-sensitive.
    b. If you do not specify details for the Username, Password fields, you can log into the Cloud Manager using default credentials: username of `admin` and the default password of `!n0r1t5@C`.
       Note: 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, please ensure that you have typed the password correctly.
       For improved security, change the default password.
    c. Click Next.
14. Configure the network. In the configuration window, enter the Cloud IP address, Domain, Domain Search, Email, Gateway, Hostname, IP address and Netmask information. Then click Next.

15. Verify that the options specified are correct.
16. Click Finish
    The OVF template is deployed to your virtual machine.
17. Optional: To alter the size of Hard disk 2, modify the settings of the deployed VM after deployment but before it is powered on.

What to do next

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 **auth set user admin** to change the password.

Identify the DataPower appliance or appliances to be used as gateway servers in your cloud and obtain the IP addresses.

Define your API Connect configuration by using the Cloud Manager. For more information, see Defining the cloud.

- **Installing virtual appliances on XenServer**
  You can create a virtual server by deploying the relevant IBM API Connect OVA template. Create all of the virtual servers that you want to use in your cloud.
- **Network configuration using the OVF Tool**
  The VMware™ OVF Tool is a command line utility that supports fast and robust hardware configuration and validation. OVF supports efficient, secure distribution of vApps and virtual machine templates. You can create a virtual machine within VMware vSphere and use the OVF Tool to export it into an OVF package for installation, either within your organization or for distribution to other organizations.
- V5.0.1+ **Managing disks on Management appliances**
  Management appliances can use disk encryption to protect data. You can also manage the swap space allocation and size of the code disk for an appliance, and add new data disks to an appliance.
- **Configuring to use DHCP addressing**
  You can configure a virtual machine in a network to use a Dynamic Host Configuration Protocol (DHCP) server for addressing.
- **Configuring to use static addressing**
  You can configure a virtual machine to use static network addressing.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing virtual appliances on XenServer

You can create a virtual server by deploying the relevant IBM® API Connect OVA template. Create all of the virtual servers that you want to use in your cloud.

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. Download a local copy of the API Connect OVA file.
   Important: Download the OVA file to a storage device that is accessible by the virtual infrastructure and where the software is to be installed.
2. From the XenCenter, select File > Import.
   The Import OVF/OVA Package wizard displays.
3. Specify the path to the downloaded OVA file. Click Next.
   The Select the location where the imported VM will be placed page displays.
4. Select the location: pool or stand-alone server to place the imported virtual machine (VM). Click Next.
   The Select target storage page displays.
5. Choose Place all imported virtual disks on this target SR: and select a target from the drop-down. Click Next.
   The Select network to connect VM page displays.
6. Specify the network to which you want to place your VM. Click Next.
   The Select your import security settings page displays.
7. Skip security settings. Selecting Verify manifest content might prevent the import of the OVA file. Click Next.
   The Use Operating System Fixup to ensure hypervisor interoperability page displays.
8. Choose Don't use Operating System Fixup. Click Next.

The Configuring networking options for the Transfer VM page displays.

9. Specify network settings for a transfer VM that runs on the network on which your XenServer is running.
    - To use DHCP to specify network settings, choose Automatically obtain network settings using DHCP. Click Next.
    - If you do not use DHCP, you need to specify an available IP address. This IP address must be dedicated to the API Connect VM and not shared with other virtual machines. Click Next.

    The Review the import settings page displays.
10. Click Finish.
    In the XenCenter navigation pane, the VM displays under the XenServer you specified.
11. Review the vCPUs and memory settings for the VM.
    a. Select the new VM and choose Properties.
    The VM properties dialog displays.
    b. Optional: Change the name of the VM to distinguish between multiple API Connect virtual appliances.
    c. Select the CPU tab and review the number of vCPUs allocated to the VM.
    See the Software Product Compatibility Report for the minimum supported CPU configuration. Click OK to save the settings.
    d. Select the Memory tab.
    The amount of memory that is allocated to the VM displays.
    e. Optional: Update the amount of memory that is allocated to the VM by clicking Edit and entering a fixed amount of memory to be allocated to the VM.
    See the Software Product Compatibility Report for the minimum supported memory configuration. Click OK to save the settings.
12. Log in to the serial console and configure the network. For more information about configuring the VM to use static addressing, see Configuring to use static addressing. This step is optional if the network was configured using DHCP.
13. Start the VM.
    a. In the XenCenter navigation pane, select the VM.
    b. From the XenCenter toolbar, click Start.
14. Log in to the management console with the default username of `admin` and the default password of `!n0r1t5@C`.
    Note: 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, please ensure that you have typed the password correctly.
15. Accept the licensing agreements.
16. Change the username and password.
    The password must consist of at least 8 characters and must contain at least three of the following types of characters:
    - Lowercase letters
    - Uppercase letters
    - Numbers
    - Special characters, for example, exclamation point (!)

    Restriction: Take a note of the credentials because user names are case-sensitive.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Network configuration using the OVF Tool

The VMware™ OVF Tool is a command line utility that supports fast and robust hardware configuration and validation. OVF supports efficient, secure distribution of vApps and virtual machine templates. You can create a virtual machine within VMware vSphere and use the OVF Tool to export it into an OVF package for installation, either within your organization or for distribution to other organizations.

About this task

You can use the OVF Tool on the command line to specify configuration properties, including IP address, gateway, and netmask, saving you from the time consuming tasks of configuring numerous machines individually.

Note: The following information applies to VMware only. Depending on the VMware version and features you are using, some of the information might vary. Specifically, if your VMware server is not licensed to use **`vmtoolsd`**, options specified using the **`--prop`** option are not passed through to deployed virtual machines.

Deploying an IBM API Connect OVA with the OVF Tool

The following example shows how you can use the OVF Tool to deploy an IBM® API Connect OVA:

Example

```
/usr/lib/vmware-ovftool/ovftool --quiet --overwrite --powerOffTarget --powerOn --acceptAllEulas --
noSSLVerify -n=APIM-v2015XXXX-01 -dm=thin -ds="ESX NFS (1)" -nw="VM Network" --
prop:"ConfigNET.ipaddr.1"="1.1.1.1" --prop:"ConfigNET.gateway.1"="1.1.9.1" --
prop:"ConfigNET.netmask.1"="255.255.255.0" --prop:"ConfigNET.hostname.1"="myAPIM01" --
prop:"ConfigNET.domain.1"="test.ibm.com" --prop:"ConfigNET.search.1"="test.ibm.com" --
prop:"ConfigNET.pri_dns.1"="1.1.9.2" --prop:"ConfigNET.sec_dns.1"="1.1.9.3"  --
prop:"ConfigNET.ntp.1"="1.1.9.4" /path/to/APIManagement-ManagementAppliance-2015XXXX.ova 
vi://serverUN:serverPW@myVMserver
```

After the machine in the previous example boots up, you can deploy another machine and join it to the first one, as the next example shows

```
/usr/lib/vmware-ovftool/ovftool --quiet --overwrite --powerOffTarget --powerOn --acceptAllEulas --
noSSLVerify -n=APIM-v2015XXXX-02 -dm=thin -ds="ESX NFS (1)" -nw="VM Network" --
prop:"ConfigNET.ipaddr.1"="1.1.1.2" --prop:"ConfigNET.gateway.1"="1.1.9.1" --
prop:"ConfigNET.netmask.1"="255.255.255.0" --prop:"ConfigNET.hostname.1"="myAPIM02" --
prop:"ConfigNET.domain.1"="test.ibm.com" --prop:"ConfigNET.search.1"="test.ibm.com" --
prop:"ConfigNET.pri_dns.1"="1.1.9.2" --prop:"ConfigNET.sec_dns.1"="1.1.9.3"  --
prop:"ConfigNET.ntp.1"="1.1.9.4" --prop:"ConfigCastIron.mgmt_username.1"="admin" --
prop:"ConfigCastIron.mgmt_password.1"='adminPassword' --prop:"ConfigCastIron.mgmt_ipaddr.1"="1.1.1.1" 
/path/to/APIManagement-ManagementAppliance-2015XXXX.ova vi://serverUN:serverPW@myVMserver
```

With the exception of three additional properties (which join the second machine to the first machine), the commands in the two examples are identical. Here is a breakdown of each option in the second command:

```
/usr/lib/vmware-ovftool/ovftool 
   // required binary, they must also have the correct license for their VM server
--quiet --overwrite --powerOffTarget --powerOn --acceptAllEulas --noSSLVerify  
   // various ovftool deploy options
-n=APIM-v2015XXXX-02 
   // desired VM name
-dm=thin -ds="ESX NFS (1)" -nw="VM Network" 
   // deploy method, server disk to use, and server network to use
--prop:"ConfigNET.ipaddr.1"="1.1.1.2" --prop:"ConfigNET.gateway.1"="1.1.9.1" --
prop:"ConfigNET.netmask.1"="255.255.255.0"         // static network values
--prop:"ConfigNET.hostname.1"="myAPIM02" --prop:"ConfigNET.domain.1"="test.ibm.com" --
prop:"ConfigNET.search.1"="test.ibm.com"    // more static network values
--prop:"ConfigNET.pri_dns.1"="1.1.9.2" --prop:"ConfigNET.sec_dns.1"="1.1.9.3"  --
prop:"ConfigNET.ntp.1"="1.1.9.4" 
   // even more static network values
 --prop:"ConfigCastIron.mgmt_username.1"="admin" --prop:"ConfigCastIron.mgmt_password.1"='adminPassword'
--prop:"ConfigCastIron.mgmt_ipaddr.1"="1.1.1.1" 
   // login information to talk to the first server
/path/to/APIManagement-ManagementAppliance-2015XXXX.ova vi://serverUN:serverPW@myVMserver  
   // URI of OVA and VM server
```

If the deploy command is successful, the second machine's network will appear as follows:

```
myAPIM02/APIManagement> net show all
Hostname (Static): myAPIM02
Domain (Static): test.ibm.com
Domain Search (Static): test.ibm.com
Nameservers (Static): 1.1.9.2 1.1.9.3
Gateway (Static on eth0): 1.1.9.1
NTP (Static): 1.1.9.4

Carrier timeout: 30 seconds

Eth0 (00:50:56:8a:2d:3f): Static
  Address: 1.1.1.1, Netmask: 255.255.255.0, Broadcast: 1.1.1.255
  MTU: 1500
  Link Status is not available

Routes: [None]

myAPIM02/APIManagement> net show status
Last attempt to start networking: Wed Sep 16 19:06:18 2015 GMT
Last time networking started:     Wed Sep 16 19:06:18 2015 GMT
Last time networking stopped:     Wed Sep 16 19:06:06 2015 GMT
Reason for last networking stop:  Normal shutdown
Current activity:                 Monitoring network
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Managing disks on Management appliances

Management appliances can use disk encryption to protect data. You can also manage the swap space allocation and size of the code disk for an appliance, and add new data disks to an appliance.

- V5.0.1+ **Disk encryption**
  You can implement disk encryption to safeguard your data for any Management appliances. You can apply disk encryption to your current appliance by reformatting the hard drive.
- V5.0.1+ **Swap space allocation**
  Management appliances allocate a specific amount of swap space based on the size of the memory that the appliance has, and is restricted by the amount of available disk space.
- V5.0.1+ **Increasing code disk size for appliances**
  The code disk for new Management appliances is larger for IBM® API Connect Version 5.0 than for IBM API Management Version 3.0 and IBM API Management Version 4.0. If you have an existing Management appliance that you want to upgrade to IBM API Connect Version 5.0, you can increase the code disk size for an appliance.
- **Adding a new data disk to a Management appliance**
  IBM API Connect uses the data disk on a Management appliance to store analytics data that is collected from the directory on which Elasticsearch, Informix®, and other processes are mounted. Over time, this disk begins to fill up. You allocate more data disk space by adding a new virtual data disk to the Management appliance, **not** by increasing the size of the existing data disk.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Disk encryption

You can implement disk encryption to safeguard your data for any Management appliances. You can apply disk encryption to your current appliance by reformatting the hard drive.

Disk encryption protects your data by converting it into unreadable code that cannot be deciphered by unauthorized personnel. If you install IBM® API Connect Version 5.0 on new Management appliances, disk encryption is implemented automatically to safeguard your data.

Note: Implementation of disk encryption at the first possible opportunity is enabled by default for new Management appliances. However, the hard drives of existing Management appliances remain unencrypted to preserve data.
It is not possible to retroactively enable disk encryption on a Management appliance. If you want to implement encryption on a disk containing data, you must reformat the hard drive. By reformatting your hard drive, networking is affected and analytics data can be lost. The **system clean all** command resets the network configuration. As a result, you may need access to a VMware console before you can re-configure the network. To view the encryption status, execute the **system show platform** command. The following status is an example of what is displayed when the disks are not encrypted:

```
    slm-01/APIConnect> system show platform
    Serial Number: VMWAA729A73KML9O
    Management MAC: 00:0c:29:c8:40:22
    Encrypted disk partitions: 2 (apim, swap)
    Unencrypted disk partitions: 2 (sysrw, wip)
```

The following status is an example of what is displayed when the disks are encrypted:

```
    slm-02/APIConnect> system show platform
    Serial Number: VMWUY9SN4BAQDX6E
    Management MAC: 00:0c:29:ea:62:c4
    Encrypted disk partitions: 4 (apim, sysrw, wip, swap)
    Unencrypted disk partitions: 0
```

Following are three scenarios for implementing disk encryption on a Management appliance.

- Single management server – This approach preserves configuration data (APIs, Plans, etc.) but not analytics data. This approach also involves downtime as the Management server is effectively being re-installed. To implement disk encryption for a single Management server:
    1. Back up the configuration using the **config save apiconfig** command, then execute the **debug postmortem export** command to export the postmortem archive to an FTP or SFTP server.
    2. Execute the **system clean all** command to reset the appliance to the factory defaults.
       Note: Use this command with care. The **system clean all** command is equivalent to installing a new appliance and creates new, encrypted disks.
    3. Restore from backup. With the exception of analytics, this preserves configuration data (APIs, Plans, etc.).
- Multiple Management servers using current servers – This approach preserves all configuration and analytics data and does not require configuring new servers, which enables you to continue using existing IP addresses, firewall settings and load balancing configurations. However, the Management tier will have reduced capacity during the conversion, as one server will be temporarily offline. To implement disk encryption for multiple Management servers using current servers:
    1. Back up the configuration using the **config save apiconfig** command, then execute the **debug postmortem export** command to export the postmortem archive to an FTP or SFTP server.
    2. Remove one Management server from the cloud. After it has been removed, fully reset the server with **system clean all**.
       Note: Use this command with care. The **system clean all** is equivalent to installing a new appliance and creates new, encrypted disks.
    3. Add the server back into the cloud. Allow the servers time to fully replicate and balance analytics data.
    4. Repeat for other Management servers.
- Multiple Management servers by adding new servers – This approach preserves all configuration and analytics data and preserves runtime capability. However it requires allocating new IP addresses which may impact firewalls, load balancing and other functions. To implement disk encryption for multiple Management servers by adding new servers:
    1. Back up the configuration using the **config save apiconfig** command, then execute the **debug postmortem export** command to export the postmortem archive to an FTP or SFTP server.
    2. Allocate a new Management server and add it to the cloud. Allow the servers time to fully replicate/balance analytics data.
    3. Remove an old Management server from the cloud and discard it.
    4. Repeat as necessary to replace all Management servers.

Related information

- System commands

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Swap space allocation

Management appliances allocate a specific amount of swap space based on the size of the memory that the appliance has, and is restricted by the amount of available disk space.

The amount of swap space that a virtual machine requires is double the size of its RAM. This amount provides enough space for data that is no longer required by RAM to be placed in the swap space. In addition, there is enough space for emergency situations where RAM needs to be cleared due to hypervisor demands.

The Management appliance cannot allocate all of the space from a data disk to swap space, as space is required for databases. The Management appliance follows a specific set of conditions when it allocates swap space:

1. The maximum amount of work in progress partition that can be dedicated to swap space is 25%.
2. When the Management appliance increases the swap space size, the swap size does not increase by more than 20% at a single instance. The maximum increase of 20% in one instances enables Informix and analytics to allocate their disk spaces during the boot process.
3. You can allocate more data disk space by following the procedure in Adding a new data disk to a Management appliance
4. You can increase the size of your code disk by following the procedure in Increasing code disk size for appliances.

Note: Management appliances have less disk space after upgrading to IBM API Connect Version 5.0 because more space is allocated as swap space.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Increasing code disk size for appliances

The code disk for new Management appliances is larger for IBM® API Connect Version 5.0 than for IBM API Management Version 3.0 and IBM API Management Version 4.0. If you have an existing Management appliance that you want to upgrade to IBM API Connect Version 5.0, you can increase the code disk size for an appliance.

About this task

Increasing the size of the code disk is different to increasing the size of the data disk. For more information on increasing the size of your data disk, see Installing the Management virtual server.

Procedure

1. Backup any configurations that you have on your appliance by running the following `config save apiconfig` command. For more information on running the `config save apiconfig` command, see Configuration commands.
2. Upgrade the appliance to a version of API Connect that supports a larger disk size. For more information on upgrading your version of IBM API Management Version 4.0 or IBM API Connect Version 5.0, see Upgrading your API Connect cloud.
3. In your virtual machine software, remove any snapshots from your appliance.
   Note: Virtual machine software does not allow you to modify the disk size of an appliance if it contains snapshots.
4. Select the option to edit the settings of your appliance, identify the setting governs the size of the disk, and increase the code disk size to 10GB.
5. Save your changes and reboot your appliance.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a new data disk to a Management appliance

IBM® API Connect uses the data disk on a Management appliance to store analytics data that is collected from the directory on which Elasticsearch, Informix®, and other processes are mounted. Over time, this disk begins to fill up. You allocate more data disk space by adding a new virtual data disk to the Management appliance, **not** by increasing the size of the existing data disk.

Before you begin

For information on determining how much disk space you need, see IBM API Connect topology.

You can determine the utilization of the data disk by running the **stat show all** CLI command on the Management appliance; the data disk corresponds to the /wip entry in the command output.

V5.0.7+ API Connect monitors the amount of disk space that is used by the analytics data. As the disk usage approaches 70 percent with data from the Elasticsearch, Informix, and other processes, a warning email is sent with recommendations for reducing the amount of information that you are saving to the disk. The email addresses that are specified for the following roles receive the email notifications when the disk space reaches 70% full, 80% full, and 90% full:

- cmc admin
- cmc owner
- topology administrator.

These notifications provide advanced notice when you are running out of disk space, so you can take steps to resolve the issue. The following list highlights some possible methods of managing your disk space:

- Set your log settings to retain less information in the logs. You can change these settings by selecting Settings > Analytics in the IBM API Connect Console.
- Add a virtual disk to your configuration. Complete the procedure later in this topic to add a virtual disk.
- Offload your analytics log information to another product. See Configuring destination targets for API Connect analytics data for information about how to offload your data.

Procedure

1. Power off the virtual machine for your Management appliance.
2. In VMWare, allocate an additional virtual disk of the required size, and attach it to the Management appliance, by completing the following steps:
    a. Right-click your Management appliance name and select Edit settings.
    b. Select the Hardware tab and click Add.
    c. Choose Hard disk from the list of options, then click Next.
    d. Select the Create a new virtual disk radio button, then click Next.
    e. Under the Capacity heading, specify the required virtual disk size.
    Note: The maximum supported size for the new data disk is 2 TB.
    f. Under the Disk Provisioning heading, select the appropriate option for your organization.
    Select thick provisioning to prevent unexpected failures that can result when the host cannot allocate enough space when you use thin provisioning.
    g. Under the Location heading, select Store with the virtual machine, then click Next.
    h. Leave all content under the Virtual device node heading as default, then click Next.
    i. Review your settings, then click Finish.
3. Power on the virtual machine.
4. Run the following command to view the available disk space:

    ```
    stat show all
    ```

5. Optional: If the available disk space is not greater than it was before you added the disk, run a system reboot and repeat step 4.
6. If it did not increase following the restart, capture the logs and contact IBM support.

Results

When the Management appliance reboots, it discovers the new virtual disk and adds it as usable space.

You can repeat this procedure up to six times, but the maximum supported combined size of all disks is 2 TB per Management server. Do not remove or resize data disks after the Management appliance begins using them.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring to use DHCP addressing

You can configure a virtual machine in a network to use a Dynamic Host Configuration Protocol (DHCP) server for addressing.

About this task

Ensure that your DHCP server is configured, as required, for IBM® API Connect.

Procedure

1. Assign the host name and IP address in the Domain Name Server (DNS) for the network interfaces on the appliance.
    The number of network interfaces that are required for API Connect varies depending on the type of server:
    - Management server - one network interface
2. For each network interface, configure the Media Access Control (MAC) address to an IP address mapping. The host system (such as VMware) might assign a unique default MAC address to each network interface. You can use this default address, or, the address can be changed in the configuration of the virtual appliance when it is deployed but before it is powered on.
    For each network interface, there must be a matching entry in the DHCP server configuration that maps the MAC address of the interface to a unique IP address.

Configuring the virtual appliance to use DHCP

Procedure

1. Power on the appliance.
   The first time the appliance starts up, it attempts to discover a DHCP server and request an IP address by using DHCP. If the configuration, in the previous steps, was successful, the appliance starts the networking automatically and connects to the network.
2. Using the virtual machine console, log in to the virtual machine using the default user name and password.
   - Localhost login: `admin`
   - Password: `!n0r1t5@C`

     Note: The keyboard mapping of the vSphere client terminal uses a US layout. If your workstation keyboard uses a different layout, you might need to adapt to the US keyboard layout while you are working in the vSphere client terminal.

   The following prompt displays:

   ```
   none/Standalone>
   ```

3. Monitor the status of the system, by repeatedly entering the following command:

   ```
   none/Standalone> system show status
   ```

   As the network settings are applied, the network status display is updated and shows the following states:
   - Up
   - Stopping
   - Starting
4. Monitor the state of the network:

   ```
   none/Standalone> net show status
   ```

   The network settings are applied when status of the Current activity lists:

   ```
   Monitoring network
   ```

5. Verify the data and management interface network settings, by using the following command:

   ```
   MyHostName/Standalone> net show active
   ```

6. Verify the connection by connecting to the cloud console.
   The first time the appliance is started, it might take several minutes for the appliance to complete its initialization.

Related tasks

- Configuring to use static addressing

Related information

- Network commands
- System commands

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring to use static addressing

You can configure a virtual machine to use static network addressing.

Before you begin

In your DNS server, assign the host name and IP address for the management interface.

About this task

Network settings that you specify are only committed to the Management server when you run the **net restart** command. If a system restart occurs before you commit your network settings, those network settings are lost.

Procedure

1. Using the virtual machine console, log in to the virtual machine with the default user name and password.
   - localhost login: `admin`
   - Password: `!n0r1t5@C`
     Note: 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, please ensure that you have typed the password correctly.

   The following prompt is displayed:

   ```
   none/Standalone>
   ```

   Note: The keyboard mapping of the vSphere Client terminal uses a US layout. If your workstation keyboard uses another layout, your login attempt might fail.
2. Set the host name for the management interface:

   ```
   none/Standalone> net set hostname static hostname
   ```

   After the host machine is restarted at the end of this procedure, the prompt changes to the following setting:

   ```
   hostname>
   ```

3. Set the DNS domain name:

   ```
   none/Standalone> net set domain static domain-name
   ```

4. Set the DNS name server:

   ```
   none/Standalone> net set nameserver static dns-ipaddress
   ```

5. Set the network address for the management interface:
   For a Management server:

   ```
   none/Standalone> net set eth<n> address ipaddress mask netmask [bcast broadcast]
   ```

   Note: The **bcast *broadcast*** parameter is optional.
6. Set the address of the default gateway for your TCP/IP network:
   For a Management server:

   ```
   none/Standalone> net set gateway static ip-address  {eth<n>|none}
   ```

7. Search for hosts by using only the local domain:

   ```
   none/Standalone> net set search none
   ```

8. Set the Network timeserver to the appropriate setting for your environment:
   - If you have a timeserver on your network:

     ```
     none/Standalone> net set ntp static ntp-address
     ```

   - If you do not have a timeserver on your network:

     ```
     none/Standalone> net set ntp none
     ```

9. Review the settings:

   ```
   none/Standalone> net show memory
   ```

10. Apply the network settings:

    ```
    none/Standalone> net restart
    ```

11. Monitor the status of the system, by repetitively issuing the following command:

    ```
    none/Standalone> system show status
    ```

    As the network settings are applied, the network status progresses through the following states:
    - Up
    - Stopping
    - Starting
    - Up

    Note: As network settings are applied, the run time is unavailable.

12. Monitor the state of the network:

    ```
    hostname> net show status
    ```

    When the network settings are applied, the status of the Current activity shows

    ```
    Monitoring network
    ```

13. Verify the settings by entering the following command:

    ```
    hostname> net show active
    ```

14. Reset any address that is incorrect.
    Run the following command to see a list of addresses:

    ```
    net show all
    ```

15. Log out `hostname> exit`
    The session closes.

Related tasks

- Configuring to use DHCP addressing

Related information

- Network commands
- System commands

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Installing a containerized runtime environment

Install a containerized runtime environment to provide a lightweight, efficient deployment location for your APIs and applications.

Before you begin

Before setting up a containerized runtime environment, you must have installed IBM® API Connect.
Additionally, you must have:

- A host server with a supported operating system.
- Administrative privileges, such as root privileges, on the host server.
- The Management server.
- A supported version of Node.js.
- Other requirements depend on the container system being used.

About this task

You can set up the following containerized runtime environments for use with API Connect:

- **Kubernetes**. For more information, see Setting up a Kubernetes runtime environment.
- **Docker Swarm**. For more information, see Setting up a Docker Swarm runtime environment.

- **Setting up a Kubernetes runtime environment**
  You can use Kubernetes containers to run your APIs and applications being managed by API Connect. 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. For the latest information, refer to the Kubernetes documentation at https://kubernetes.io.
- **Build and deploy a sample application to a Kubernetes container**
- **Setting up a Docker Swarm runtime environment**
  You can use Docker containers to run your APIs and applications being managed by API Connect. 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. For the latest information, refer to https://www.docker.com/.

- **Managing microservices in Docker Swarm**
- **Migrating LoopBack applications from collectives to containers**
  Instead of deploying to deprecated Collectives, you can deploy Loopback applications to Docker Swarm or Kubernetes container runtimes. The application's runtime code is packaged and deployed according to the nature of the target runtime. The gateway routes API calls to the internal endpoint hosted by the target runtime using the target URL value that is defined in the assembly section of the API. The assembly defines how the gateway handles calls to the API.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Setting up a Kubernetes runtime environment

You can use Kubernetes containers to run your APIs and applications being managed by API Connect. 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. For the latest information, refer to the Kubernetes documentation at https://kubernetes.io.

Before you begin

System and software requirements:

- Ubuntu version 16.04
- Kubernetes version 1.5.3
- etcd version 3.1.2
- Kubernetes CNI version 0.4.0
- CloudFlare PKI/TLS toolkit (CFSSL) version 1.2

Note: The information in this article is specific to the system and software versions noted above and may not be accurate for other versions.
Networking requirements: All nodes must be interconnected and reachable on the same network. There must be two additional network namespaces that do not overlap with either the service IP addresses or the pod IP addresses.

This document assumes the following network configuration. Your configuration will likely differ and you must modify the configuration files to match your settings.

- Public network CIDR: 10.0.2.0/24
- Public facing NIC: enp0s2
- Private network CIDR: 172.28.127.0/24
- Private NIC: enp0s5
- Pod network: 10.2.0.1/24
- Service network CIDR: 10.3.0.0/24

Hostname	FQDN	IP on private network	IP on public network
master1	master1.k8s.myorg.com	172.28.127.2	10.0.2.15
master2	master2.k8s.myorg.com	172.28.127.3	10.0.2.16
master3	master3.k8s.myorg.com	172.28.127.4	10.0.2.17
worker1	worker1.k8s.myorg.com	172.28.127.5	10.0.2.18
worker2	worker2.k8s.myorg.com	172.28.127.6	10.0.2.19
worker3	worker3.k8s.myorg.com	172.28.127.7	10.0.2.20

Some services also run on known IPs within the service network:

- API Server: 10.3.0.1
- DNS Server: 10.3.0.10

You also must configure a load balancer or RR-DNS for the API servers. The load-balanced FQDN and IP address for the API servers is:

```
apiserver.k8s.myorg.com (10.0.2.45)
```

About this task

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.

Follow this procedure:

- Install Python and required packages
- Prepare certificates
- Create etcd cluster
- Set up nodes
- Set up master nodes
- Start kubelet on all nodes
- Add critical addons

See https://github.com/ibm-apiconnect/kubernetes-setup for the scripts used in this article.

Procedure

1. Prepare machines by installing Python and requisite Python packages:

   ```
   apt-get update
   apt-get install -y python python-simplejson python-pip
   ```

2. Prepare certificates
   This procedure assumes self-signed certificates creating using CloudFlare's PKI toolkit (CFSSL). For more information on CFSSL, see https://cfssl.org/.You can also use `openssl` to generate these certificates. Contact your company's security/compliance team to get the appropriate certificates.

   In general, you create certificates on your local machine and copy them to the remote machines later.

   Note: Copy the certificates you create on one host to the other hosts, except for the peer certificatess.

   a. **Download cfssl and cfssljson**. Download version 1.3 of `cfssl` and `cfssljson` from https://pkg.cfssl.org/ by entering one of the following commands:
      On Linux:

      ```
      curl https://pkg.cfssl.org/R1.2/cfssl_linux-amd64 -o /usr/local/bin/cfssl
      curl https://pkg.cfssl.org/R1.2/cfssljson_linux-amd64 -o /usr/local/bin/cfssljson
      chmod +x /usr/local/bin/cfssl /usr/local/bin/cfssljson
      ```

      On MacOS:

      ```
      curl https://pkg.cfssl.org/R1.2/cfssl_darwin-amd64 -o /usr/local/bin/cfssl
      curl https://pkg.cfssl.org/R1.2/cfssljson_darwin-amd64 -o /usr/local/bin/cfssljson
      chmod +x /usr/local/bin/cfssl /usr/local/bin/cfssljson
      ```

   b. Add `/usr/local/bin/cfssl` and `/usr/local/bin/cfssljson` to your PATH environment variable.

   c. Create CA certificate.
      In a new directory, create your CA certificates and signing profiles.

      Create ca-config.json and add the following to it:

      ```
      {
          "signing": {
              "default": {
                  "expiry": "168h"
              },
              "profiles": {
                  "server": {
                      "expiry": "43800h",
                      "usages": [
                          "signing",
                          "key encipherment",
                          "server auth"
                      ]
                  },
                  "client": {
                      "expiry": "43800h",
                      "usages": [
                          "signing",
                          "key encipherment",
                          "client auth"
                      ]
                  },
                  "peer": {
                      "expiry": "43800h",
      ```

```
                "usages": [
                    "signing",
                    "key encipherment",
                    "server auth",
                    "client auth"
                ]
            }
        }
    }
}
```

Create ca-csr.json and add the following to it:

```
{
    "CN": "Dev CA",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Enter the following command to create the CA certificate and key files:

```
cfssl gencert -initca ca-csr.json | cfssljson -bare ca -
```

d. Create Kubernetes API server certificate.
The Kubernetes API server makes outgoing calls to the Controller, Scheduler, and Kubelets and accepts incoming API calls from many clients. Thus, it uses both `server auth` and `client auth` capabilities.

Customize the file to include host names and IP addresses for your master servers, load balancer, and cluster internal API server IP.

Create apiserver.json and add the following to it:

```
{
    "CN": "apiserver",
    "hosts": [
      "master1.k8s.myorg.com", "master1", "172.28.127.2",
      "master2.k8s.myorg.com", "master2", "172.28.127.3",
      "master3.k8s.myorg.com", "master3", "172.28.127.4",
      "apiserver.k8s.myorg.com", "10.0.2.45",
      "10.3.0.1",
      "localhost", "127.0.0.1"
    ],
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Create the API server certificate and key files by entering the following command:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=peer apiserver.json | cfssljson \
-bare apiserver
```

e. Create Kubernetes Scheduler certificate.
The Kubernetes Scheduler is a client to the API server and requires only `client auth`capabilities.

Create scheduler.json and add the following to it:

```
{
    "CN": "scheduler",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the scheduler certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=client scheduler.json | cfssljson \
-bare scheduler
```

f. Create Kubernetes proxy certificate.
The Kubernetes proxy is a client to the API server and requires only **`client auth`** capabilities.

Create proxy.json and add the following to it:

```
{
    "CN": "proxy",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the proxy certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=client proxy.json | cfssljson \
-bare proxy
```

g. Create Kubernetes Controller certificate.
The Kubernetes Controller is a client to the API server and requires only **`client auth`** capabilities.

Create controller.json and add the following to it:

```
{
    "CN": "controller",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the controller certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=client controller.json | cfssljson \
-bare controller
```

h. Create Kubelet certificate.
The Kubelet service is both a client and server to the Kubernetes API server and thus uses both `server auth` and `client auth` capabilities. You will create a single certificate for all nodes. However, you can create individual certificates if needed.

Create a file named kubelet.json and customize the hostnames and IP addresses for your master and worker nodes (shown in bold in the examle below):

```
{
    "CN": "kubelet",
    "hosts": [
      "master1.k8s.myorg.com", "master1", "172.28.127.2",
      "master2.k8s.myorg.com", "master2", "172.28.127.3",
      "master3.k8s.myorg.com", "master3", "172.28.127.4",
      "worker1.k8s.myorg.com", "worker1", "172.28.127.5",
      "worker2.k8s.myorg.com", "worker2", "172.28.127.6",
      "worker3.k8s.myorg.com", "worker3", "172.28.127.7"
    ],
    "key": {
        "algo": "ecdsa",
        "size": 256
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the Kubelet certificate and key files.

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=peer kubelet.json | cfssljson \
-bare kubelet
```

i. Create Kubernetes admin user certificate, admin.json:

```
{
    "CN": "admin",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "O": "system:masters",
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the admin user certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=client admin.json | cfssljson \
-bare admin
```

j. Create etcd server certificates.

You will create a single certificate for all nodes. However, you can create individual certificates if needed. Create etcd.json and customize the hostnames and IP addresses for your etcd nodes (shown in bold in the example below):

```
{
    "CN": "etcd",
    "hosts": [
      "etcd1.k8s.myorg.com", "etcd1", "172.28.127.8",
      "etcd2.k8s.myorg.com", "etcd2", "172.28.127.9",
      "etcd3.k8s.myorg.com", "etcd3", "172.28.127.10"
    ],
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the etcd certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=server etcd.json | cfssljson \
-bare etcd
```

k. Create etcd peer certificates.
Create peer certificates for each etcd server. Etcd uses the certificates to secure communicate between nodes.

Create a file for each of your etcd nodes with the name etcd-peer-*hostname*.json, where *hostname* is the name of each host. Customize the host name and IP address in each file (shown in bold in the example below):

```
{
    "CN": "etcd",
    "hosts": [
      "etcd1.k8s.myorg.com", "etcd1", "172.28.127.8"
    ],
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the etcd peer certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=peer etcd-peer-HOSTNAME.json | cfssljson \
-bare etcd-peer-HOSTNAME
```

l. Create etcd client certificate
Create a client certificate which API Server will use to communicate with etcd in etcd-client.json:

```
{
    "CN": "etcd-client",
    "key": {
        "algo": "rsa",
        "size": 2048
    },
    "names": [
        {
            "C": "US",
            "L": "CA",
            "ST": "San Francisco"
        }
    ]
}
```

Run the following command to create the etcd client certificate and key files:

```
cfssl gencert -ca=ca.pem \
-ca-key=ca-key.pem \
-config=ca-config.json \
-profile=client etcd-client.json | cfssljson \
-bare etcd-client
```

3. Create etcd cluster
   a. Create SSL certificate directory.
      Enter this command on each etcd server:

      ```
      useradd -r -s /sbin/nologin etcd
      mkdir -p /etc/etcd/ssl
      ```

   b. Copy certificates.
      Copy the following certificates onto each etcd server in directory **/etc/etcd/ssl**:

      - ca.pem
      - etcd-peer-*hostname*.pem
      - etcd-peer-*hostname*-key.pem
      - etcd.pem
      - etcd-key.pem
      - etcd-client.pem
      - etcd-client-key.pem

      Set *hostname* above to be the name of each of the etcd nodes.

      Enter this command to give the etcd user the required directory and file permissions:

      ```
      chown root:etcd -R /etc/etcd/ssl
      ```

   c. Install etcd binaries.
      Download etcd binaries from https://github.com/coreos/etcd/releases/download/v3.1.2/etcd-v3.1.2-linux-amd64.tar.gz and extract **etcd** and **etcdctl** to /usr/bin on each etcd server.

      ```
      curl -L https://github.com/coreos/etcd/releases/download/v3.1.2/etcd-v3.1.2-linux-amd64.tar.gz
      -o \
       /tmp/etcd-v3.1.2-linux-amd64.tar.gz
      cd /tmp
      tar zxf etcd-v3.1.2-linux-amd64.tar.gz
      cp /tmp/etcd-v3.1.2-linux-amd64/etcd /usr/bin/
      cp /tmp/etcd-v3.1.2-linux-amd64/etcdcctl /usr/bin/
      ```

   d. Create etcd data directory.
      Create the data directory by entering this command on each etcd server:

      ```
      mkdir -p /var/lib/etcd/
      chown etcd:etcd /var/lib/etcd/
      ```

   e. Create and start etcd service.
      Customize the following template for each etcd node and copy it to **/etc/systemd/system/etcd.service**.

      The initial cluster is a comma-separated list of **HOSTNAME=IP** for each of the etcd nodes. Customize the rest of the file with the IP address of each etcd node. Values you need to set are shown in bold in the example below:

      ```
      [Unit]
      Description=Etcd Server
      After=network.target

      [Service]
      Type=simple
      WorkingDirectory=/var/lib/etcd/
      EnvironmentFile=-/etc/etcd/etcd.conf
      ```

```
EnvironmentFile=-/etc/default/etcd
User=etcd
ExecStart=/usr/bin/etcd \
    --name="HOSTNAME" \
    --initial-cluster="https://HOST_IP:2380,https://HOST_IP:2380" \
    --listen-peer-urls="https://HOST_IP:2380" \
    --initial-advertise-peer-urls="https://HOST_IP:2380" \
    --advertise-client-urls="https://HOST_IP:2379" \
    --listen-client-urls="https://HOST_IP:2379,https://127.0.0.1:2379" \
    --data-dir=/var/lib/etcd/ \
    --trusted-ca-file="/etc/etcd/ssl/ca.pem" \
    --cert-file="/etc/etcd/ssl/etcd.pem" \
    --key-file="/etc/etcd/ssl/etcd-key.pem" \
    --peer-cert-file="/etc/etcd/ssl/etcd-peer-hostname.pem" \
    --peer-key-file="/etc/etcd/ssl/etcd-peer-hostname-key.pem" \
    --peer-trusted-ca-file="/etc/etcd/ssl/ca.pem" \
    --client-cert-auth \
    --peer-client-cert-auth \
    --initial-cluster-state=new
Restart=on-failure
LimitNOFILE=65536

[Install]
WantedBy=multi-user.target
```

Note: You created the files etcd-peer-hostname.pem and etcd-peer-hostname-key.pem in step 2.
Start etcd on each node:

```
systemctl enable etcd.service
systemctl start etcd.service
```

4. Set up nodes
   This is the common setup for both master and worker Kubernetes nodes.
   a. Create the SSL certificate directory.
      On each master and worker node enter this command:

      ```
      mkdir -p /etc/kubernetes/ssl
      ```

   b. Copy certificates.
      Copy the following certificates onto each master and worker node in directory **/etc/kubernetes/ssl/**:

      - ca.pem
      - etcd-client.pem
      - etcd-client-key.pem
      - kubelet.pem
      - kubelet-key.pem
      - proxy.pem
      - proxy-key.pem

   c. Install flannel, a virtual network that gives a subnet to each host for use with Kubernetes.
      Enter the following command to download flannel binaries from https://github.com/coreos/flannel/releases/download/v0.7.0/flanneld-amd64 and extract them to **/usr/bin** on each Kubernetes server.

      ```
      curl -L -o /usr/bin/flanneld 
      https://github.com/coreos/flannel/releases/download/v0.7.0/flanneld-amd64
      chmod +x /usr/bin/flanneld
      ```

   d. Install flannel CNI binaries.
      Download https://github.com/containernetworking/cni/releases/download/v0.4.0/cni-amd64-v0.4.0.tgz and extract to /opt/cni/bin on each Kubernetes node.

      ```
      curl -L https://github.com/containernetworking/cni/releases/download/v0.4.0/cni-amd64-
      v0.4.0.tgz -o /tmp/cni-amd64-v0.4.0.tgz
      cd /tmp
      mkdir cni-amd64-v0.4.0
      tar zxf cni-amd64-v0.4.0.tgz --directory cni-amd64-v0.4.0
      mkdir -p /opt/cni/bin
      cp -ar /tmp/cni-amd64-v0.4.0/* /opt/cni/bin/
      ```

   e. Create the flannel service.
      Copy the following to **/etc/systemd/system/flannel.service**:

```
[Unit]
Description=flannel is an etcd backed overlay network for containers
After=network-online.target
Wants=network-online.target
After=etcd.service
Before=docker.service

[Service]
Type=notify
EnvironmentFile=-/etc/default/flanneld
ExecStart=/usr/bin/flanneld $FLANNEL_OPTIONS -logtostderr
Restart=on-failure

[Install]
WantedBy=multi-user.target
RequiredBy=docker.service
```

Customize the following template with information about your etcd node (shown in bold below) and copy it to **/etc/default/flanneld**:

```
FLANNELD_ETCD_ENDPOINTS="HOST=IP:2379,HOST=IP:2379"
FLANNELD_ETCD_PREFIX="coreos.com/network/"
FLANNELD_ETCD_CERTFILE="/etc/kubernetes/ssl/etcd-client.pem"
FLANNELD_ETCD_KEYFILE="/etc/kubernetes/ssl/etcd-client-key.pem"
FLANNELD_ETCD_CAFILE="/etc/kubernetes/ssl/ca.pem"
FLANNELD_IFACE="INTERNAL_NETWORK_NIC"
FLANNEL_OPTIONS="-ip-masq"
```

Change the value for **FLANNELD_IFACE** to be the NIC for your internal traffic.

f. Initialize flannel configuration in etcd.
Customize the following command with the HOST and IP address of your etcd nodes and the POD Network CIDR.

Run the following command on one of the etcd servers.

```
etcdctl \
    --cert-file /etc/kubernetes/ssl/etcd-client.pem \
    --key-file=/etc/kubernetes/ssl/etcd-client-key.pem \
    --ca-file /etc/kubernetes/ssl/ca.pem \
    --endpoints "https://HOST_IP:2379,https://HOST_IP:2379" \
    set \
    -- coreos.com/network/config '{"Network":"POD_NETWORK","Backend":{"Type":"vxlan"}}'
```

g. Enable and start the flannel service.
On each of the master and worker nodes, run the following commands to enable and run the flannel overlay network.

```
systemctl enable flannel.service
systemctl start flannel.service
```

h. Install Docker.
On each of the master and worker nodes, run the following commands to install Docker and stop the service.

```
apt-get install -y docker.io
systemctl stop docker
```

i. Configure Docker to use flannel
Add the following lines to the /etc/default/docker file.

```
DOCKER_NOFILE=1000000
DOCKER_OPT_BIP=""
DOCKER_OPT_IPMASQ=""
```

Run the following command:

```
mkdir -p /etc/cni/net.d/
```

Copy the following lines to /etc/cni/net.d/10-flannel.conf

```
{
  "name": "kubenet",
  "type": "flannel",
  "delegate": {
    "isDefaultGateway": true,
    "ipMasq": true
  }
}
```

Bring down the docker0 bridge network.

```
ip link set dev docker0 down
brctl delbr docker0
iptables -t nat -F
```

j. Install the Kubernetes binaries
On each of the master and worker nodes, enter these commands to download **kubectl** and **kubelet** to **/usr/bin/**:

```
curl -o /usr/bin/kubelet http://storage.googleapis.com/kubernetes-
release/release/v1.5.3/bin/linux/amd64/kubelet
curl -o /usr/bin/kubectl http://storage.googleapis.com/kubernetes-
release/release/v1.5.3/bin/linux/amd64/kubectl
chmod +x /usr/bin/kubelet /usr/bin/kubectl
```

k. Create the Kubernetes service.
On each of the master and worker nodes, customize the template below and copy to **/lib/systemd/system/kubelet.service**.

```
[Unit]
Description=Kubernetes Kubelet Server
Documentation=https://github.com/kubernetes/kubernetes

[Service]
EnvironmentFile=-/etc/sysconfig/kubelet
ExecStart=/usr/bin/kubelet \
  --client-ca-file=/etc/kubernetes/ssl/ca.pem \
  --tls-cert-file=/etc/kubernetes/ssl/kubelet.pem \
  --tls-private-key-file=/etc/kubernetes/ssl/kubelet-key.pem \
  --config=/etc/kubernetes/manifests \
  --register-node=true \
  --api-servers="https://HOST:PORT" \
  --cluster_dns=DNS_SERVICE_IP \
  --cluster_domain=cluster.local \
  --allow-privileged=true \
  --enable-debugging-handlers=true \
  --port=10250 \
  --network-plugin=cni \
  --cni-conf-dir=/etc/cni/net.d \
  --hostname-override=PUBLIC_IP
Restart=always
RestartSec=2s
StartLimitInterval=0
KillMode=process

[Install]
WantedBy=multi-user.target
```

Make the following changes:

- Set **api-servers** to either the load-balanced host/port or a comma-separated list of all the API servers.
- Set **cluster_dns** to the IP address of the Kubernetes DNS service.
- For **hostname-override**, specify the public IP address.
- Set **server: https://HOST:PORT** to the host name and port number of the load balancer.

l. Set up kubeconfig for kubelet.
On each of the master and worker nodes, customize the template below and copy to **/var/lib/kubelet/kubeconfig**.

Set **server: https://HOST:PORT** to be the host name and port number of the load-balancer.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/ssl/ca.pem
    server: https://HOST:PORT
  name: mycluster
contexts:
- context:
    cluster: mycluster
    user: kubelet
  name: mycontext
current-context: mycontext
kind: Config
preferences: {}
users:
- name: kubelet
  user:
    client-certificate: /etc/kubernetes/ssl/kubelet.pem
    client-key: /etc/kubernetes/ssl/kubelet-key.pem
```

m. Set up kubeconfig for kube-proxy.
On each of the master and worker nodes, customize the template below and copy to **/var/lib/kube-proxy/kubeconfig**.

Set **server: https://HOST:PORT** to the host name and port number of the load-balancer.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/ssl/ca.pem
    server: https://HOST:PORT

name: mycluster
contexts:
- context:
    cluster: mycluster
    user: proxy
  name: mycontext
current-context: mycontext
kind: Config
preferences: {}
users:
- name: proxy
  user:
    client-certificate: /etc/kubernetes/ssl/proxy.pem
    client-key: /etc/kubernetes/ssl/proxy-key.pem
```

n. Set up manifest for kube-proxy.
On each of the master and worker nodes, customize the template below and copy to **/etc/kubernetes/manifests/proxy.manifest**.

Set **server: https://HOST:PORT** to the host name and port number of the load-balancer.

```
apiVersion: v1
kind: Pod
metadata:
  name: kube-proxy
  namespace: kube-system
  # This annotation ensures that kube-proxy does not get evicted if the node
  # supports critical pod annotation based priority scheme.
  # Note that kube-proxy runs as a static pod so this annotation does NOT have
  # any effect on rescheduler (default scheduler and rescheduler are not
  # involved in scheduling kube-proxy).
  annotations:
    scheduler.alpha.kubernetes.io/critical-pod: ''
  labels:
    tier: node
    component: kube-proxy
spec:
  hostNetwork: true
  containers:
  - name: kube-proxy
    image: "gcr.io/google_containers/hyperkube:v1.5.3"
    command:
    - /hyperkube
    - proxy
    - "--master=https://HOST:PORT"
    - "--kubeconfig=/var/lib/kube-proxy/kubeconfig"
    - "--cluster-cidr=POD_NETWORK_CIDR"
    - --proxy-mode=iptables
    - --masquerade-all
    securityContext:
      privileged: true
    volumeMounts:
    - mountPath: /etc/ssl/certs
      name: etc-ssl-certs
      readOnly: true
    - mountPath: /etc/kubernetes/ssl
```

```
      name: kubecerts
      readOnly: true
    - mountPath: /var/lib/kube-proxy/kubeconfig
      name: kubeconfig
      readOnly: false
  volumes:
  - hostPath:
      path: /etc/kubernetes/ssl
    name: kubecerts
  - hostPath:
      path: /etc/ssl/certs
    name: etc-ssl-certs
  - hostPath:
      path: /var/lib/kube-proxy/kubeconfig
    name: kubeconfig
```

5. Set up master nodes
   Run the following steps only on master nodes.
   a. Copy certificates.
      Copy the following certificates onto each master server under **`/etc/kubernetes/ssl/`**:

      - apiserver.pem
      - apiserver-key.pem
      - scheduler.pem
      - scheduler-key.pem
      - controller.pem
      - controller-key.pem
      - proxy.pem
      - proxy-key.pem
      - ca-key.pem

      Copy the following certificates onto each master server under **`/root/.kube/`**:

      - ca.pem
      - admin.pem
      - admin-key.pem

   b. Set up the admin user kubeconfig.
      On each of the master nodes, customize the template below and copy to **`/root/.kube/config`**.

      Set **`server: https://HOST:PORT`** to the host name and port number of the load-balancer.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.pem
    server: https://HOST:PORT
  name: mycluster
contexts:
- context:
    cluster: mycluster
    user: admin
  name: mycontext
current-context: mycontext
kind: Config
preferences: {}
users:
- name: admin
  user:
    client-certificate: admin.pem
    client-key: admin-key.pem
```

   c. Set up API Server ABAC permissions.
      On each of the master nodes, copy the ABAC file to **`/etc/kubernetes/abac-auth.jsonl`**:

```
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"group":"system:authenticated", "nonResourcePath": "*", "readonly": true } }
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"system:unauthenticated", "nonResourcePath": "*", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"admin", "namespace": "*", "resource": "*", "apiGroup": "*" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"controller","namespace": "*", "resource": "*", "apiGroup": "*" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"system:serviceaccount:kube-system:default", "namespace":"*", "resource":"*",
"apiGroup":"*"}}
```

```
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"system:serviceaccount:kube-system:readonly-addon", "namespace":"*", "resource":"*",
"apiGroup":"*", "readonly": true}}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "services", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "endpoints", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "secrets", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "healthz", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "configmaps", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "persistentvolumes", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "persistentvolumeclaims", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "events" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "nodes" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"kubelet", "namespace": "*", "resource": "pods" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "nodes", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "pods", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "persistentvolumeclaims", "readonly": true
}}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "persistentvolumes", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "replicationcontrollers", "readonly": true
}}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "services", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "replicasets",  "apiGroup": "*",
"readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "endpoints" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "bindings" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"scheduler", "namespace": "*", "resource": "events" }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"proxy", "namespace": "*", "resource": "services", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"proxy", "namespace": "*", "resource": "endpoints", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"proxy", "namespace": "*", "resource": "nodes", "readonly": true }}
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy", "spec":
{"user":"proxy", "namespace": "*", "resource": "events" }}
```

d. Create API Server kubelet.
On each of the master nodes, customize the template below and copy to **/etc/kubernetes/manifests/apiserver.manifest**.

Set **service-cluster-ip-range** to the service CIDR. For **etcd-servers**, provide a comma-separated list of **HOST:PORT** of the etcd servers.

```
---
kind: Pod
apiVersion: v1
metadata:
  name: kube-apiserver
  namespace: kube-system
  labels:
    tier: control-plane
    component: kube-apiserver
spec:
  hostNetwork: true
  containers:
    - name: apiserver
      image: "gcr.io/google_containers/hyperkube:v1.5.3"
      resources:
        requests:
          cpu: 250m
      command:
        - /hyperkube
        - apiserver
        - "--advertise-address=INTERNAL_IP"
        - "--secure-port=6443"
        - "--insecure-port=0"
        - "--service-cluster-ip-range=SERVICE_CIDR"
        - "--etcd-servers=https://HOST_IP:2379,https://HOST_IP:2379"
        - "--etcd-quorum-read"
        - "--cert-dir=/etc/kubernetes/ssl"
        - "--allow-privileged=true"
        - "--anonymous-auth=false"

        - "--tls-ca-file=/etc/kubernetes/ssl/ca.pem"
        - "--tls-cert-file=/etc/kubernetes/ssl/apiserver.pem"
        - "--tls-private-key-file=/etc/kubernetes/ssl/apiserver-key.pem"

        - "--etcd-cafile=/etc/kubernetes/ssl/ca.pem"
        - "--etcd-certfile=/etc/kubernetes/ssl/etcd-client.pem"
        - "--etcd-keyfile=/etc/kubernetes/ssl/etcd-client-key.pem"

        - "--client-ca-file=/etc/kubernetes/ssl/ca.pem"

        - "--kubelet-certificate-authority=/etc/kubernetes/ssl/ca.pem"
        - "--kubelet-client-certificate=/etc/kubernetes/ssl/apiserver.pem"
        - "--kubelet-client-key=/etc/kubernetes/ssl/apiserver-key.pem"
        - "--kubelet-https"

        - "--service-account-key-file=/etc/kubernetes/ssl/apiserver.pem"

        - "--admission-
control=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,ResourceQuota"
        - "--runtime-
config=extensions/v1beta1=true,batch/v1=true,rbac.authorization.k8s.io/v1alpha1=true"
        - --authorization-mode=ABAC,RBAC

        - --authorization-policy-file=/etc/kubernetes/abac-auth.jsonl
        - -v=6
      ports:
        - name: https
          hostPort: 6443
          containerPort: 6443
      volumeMounts:
        - name: etckubernetes
          mountPath: /etc/kubernetes
          readOnly: true
  volumes:
    - name: etckubernetes
      hostPath:
        path: /etc/kubernetes
```

e. Set up kubeconfig for scheduler.
On each of the master and worker nodes, customize the template below and copy to **/var/lib/kube-scheduler/kubeconfig**.

Set **server: https://HOST:PORT** to the host name and port number of the load-balancer.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/ssl/ca.pem
    server: https://HOST:PORT
  name: mycluster
contexts:
- context:
    cluster: mycluster
    user: scheduler
  name: mycontext
current-context: mycontext
kind: Config
preferences: {}
users:
- name: scheduler
  user:
    client-certificate: /etc/kubernetes/ssl/scheduler.pem
    client-key: /etc/kubernetes/ssl/scheduler-key.pem
```

f. Set up manifest for scheduler.
On each of the master and worker nodes, customize the template below and copy to **/etc/kubernetes/manifests/scheduler.manifest**.

Set **master=https://HOST:PORT** to the host name and port number of the load-balancer.

```
---
kind: Pod
apiVersion: v1
metadata:
  name: kube-scheduler
  namespace: kube-system
  labels:
    tier: control-plane
    component: kube-scheduler
spec:
  hostNetwork: true
  containers:
    - name: kube-scheduler
      image: "gcr.io/google_containers/hyperkube:v1.5.3"
      command:
        - /hyperkube
        - scheduler
        - "--algorithm-provider=ClusterAutoscalerProvider"
        - "--kubeconfig=/var/lib/kube-scheduler/kubeconfig"
        - --master=https://HOST:PORT
        - "--leader-elect=true"
      livenessProbe:
        httpGet:
          scheme: HTTP
          host: 127.0.0.1
          port: 10251
          path: /healthz
        initialDelaySeconds: 15
        timeoutSeconds: 15
      volumeMounts:
        - name: kubeconfig
          mountPath: /var/lib/kube-scheduler/kubeconfig
          readOnly: true
        - name: etckubernetesssl
          mountPath: /etc/kubernetes/ssl
          readOnly: true
  volumes:
    - name: kubeconfig
      hostPath:
        path: /var/lib/kube-scheduler/kubeconfig
    - name: etckubernetesssl
      hostPath:
        path: /etc/kubernetes/ssl
```

g. Set up kubeconfig for controller.
On each of the master and worker nodes, customize the template below and copy to **/var/lib/kube-controller/kubeconfig**.

Set **server: https://HOST:PORT** to the host name and port number of the load balancer.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/ssl/ca.pem
    server: https://HOST:PORT
  name: mycluster
contexts:
- context:
    cluster: mycluster
    user: controller
  name: mycontext
current-context: mycontext
kind: Config
preferences: {}
users:
- name: controller
  user:
    client-certificate: /etc/kubernetes/ssl/controller.pem
    client-key: /etc/kubernetes/ssl/controller-key.pem
```

h. Set up manifest for controller.
On each of the master and worker nodes, customize the template below and copy to **/etc/kubernetes/manifests/controller.manifest**.

```
---
kind: Pod
apiVersion: v1
metadata:
  name: kube-controller-manager
  namespace: kube-system
  labels:
    tier: control-plane
    component: kube-controller-manager
spec:
  hostNetwork: true
  containers:
    - name: kube-controller-manager
      image: "gcr.io/google_containers/hyperkube:v1.5.3"
      resources:
        requests:
          cpu: 200m
      command:
        - /hyperkube
        - controller-manager
        - "--cluster-name=CLUSTER_NAME"
        - "--cluster-signing-cert-file=/etc/kubernetes/ssl/ca.pem"
        - "--cluster-signing-key-file=/etc/kubernetes/ssl/ca-key.pem"

        # to add ca.crt to service accounts
        - "--root-ca-file=/etc/kubernetes/ssl/ca.pem"

        # to sign service account token
        - "--service-account-private-key-file=/etc/kubernetes/ssl/apiserver-key.pem"
        - "--kubeconfig=/var/lib/kube-controller/kubeconfig"
        - --leader-elect=true
        - "--cluster-cidr=POD_NETWORK_CIDR"
        - "--node-cidr-mask-size=24"
      volumeMounts:
        - name: etckubernetes
          mountPath: /etc/kubernetes
          readOnly: true
        - name: kubeconfig
          mountPath: /var/lib/kube-controller/kubeconfig
          readOnly: true
      livenessProbe:
        httpGet:
          host: 127.0.0.1
          port: 10252
          path: /healthz
        initialDelaySeconds: 15
        timeoutSeconds: 15
  volumes:
    - name: kubeconfig
      hostPath:
        path: /var/lib/kube-controller/kubeconfig
    - name: etckubernetes
      hostPath:
        path: /etc/kubernetes
```

6. Start kubelet on all nodes
   a. Enter these commands to start kubelet:

```
systemctl start docker
systemctl enable kubelet.service
systemctl start kubelet.service
```

   b. Check cluster status.
   On one of the master nodes, enter the following command:

```
kubectl get nodes
```

7. Add critical addons
   a. On each master node, create the addons directory.

```
mkdir -p /etc/kubernetes/addons
```

b. Create Addon-manager manifest.
On each of the master, customize the template below and copy to **/etc/kubernetes/addons/addon-manager.yaml**:

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  namespace: kube-system
  name: readonly-addon
---
apiVersion: v1
kind: Pod
metadata:
  name: kube-addon-manager
  namespace: kube-system
  labels:
    component: kube-addon-manager
spec:
  hostNetwork: true
  containers:
  - name: kube-addon-manager
    image: gcr.io/google-containers/kube-addon-manager:v6.4-beta.1
    command:
    - /bin/bash
    - -c
    - /opt/kube-addons.sh 1>>/var/log/kube-addon-manager.log 2>&1
    resources:
      requests:
        cpu: 5m
        memory: 50Mi
    volumeMounts:
    - mountPath: /etc/kubernetes/
      name: addons
      readOnly: true
    - mountPath: /var/log
      name: varlog
      readOnly: false
  volumes:
  - hostPath:
      path: /etc/kubernetes/
    name: addons
  - hostPath:
      path: /var/log
    name: varlog
```

c. Create DNS addon manifest.
On each of the master nodes, customize the template below and copy to **/etc/kubernetes/addons/dns-addon.yaml**.

Set **DNS_SERVICE_IP** to the IP address of the DNS service. This should match the **cluster_dns** setting on the Kubelet services.

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: kube-dns
  namespace: kube-system
  labels:
    addonmanager.kubernetes.io/mode: EnsureExists
---
apiVersion: v1
kind: Service
metadata:
  name: kube-dns
  namespace: kube-system
  labels:
    k8s-app: kube-dns
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
    kubernetes.io/name: "KubeDNS"
spec:
  selector:
    k8s-app: kube-dns
  clusterIP: DNS_SERVICE_IP
  ports:
  - name: dns
    port: 53
    protocol: UDP
  - name: dns-tcp
    port: 53
    protocol: TCP
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: kube-dns
  namespace: kube-system
  labels:
    k8s-app: kube-dns
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
spec:
  # replicas: not specified here:
  # 1. In order to make Addon Manager do not reconcile this replicas parameter.
  # 2. Default is 1.
  # 3. Will be tuned in real time if DNS horizontal auto-scaling is turned on.
  strategy:
    rollingUpdate:
      maxSurge: 10%
      maxUnavailable: 0
  selector:
    matchLabels:
      k8s-app: kube-dns
  template:
    metadata:
      labels:
        k8s-app: kube-dns
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
        scheduler.alpha.kubernetes.io/tolerations: '[{"key":"CriticalAddonsOnly", 
"operator":"Exists"}]'
    spec:
      volumes:
      - name: kube-dns-config
        configMap:
          name: kube-dns
      containers:
      - name: kubedns
        image: gcr.io/google_containers/k8s-dns-kube-dns-amd64:1.14.1
        resources:
          # TODO: Set memory limits when we've profiled the container for large
          # clusters, then set request = limit to keep this container in
          # guaranteed class. Currently, this container falls into the
          # "burstable" category so the kubelet doesn't backoff from restarting it.
          limits:
            memory: 170Mi
          requests:
            cpu: 100m
            memory: 70Mi
        livenessProbe:
          httpGet:
            path: /healthcheck/kubedns
            port: 10054
            scheme: HTTP
          initialDelaySeconds: 60
          timeoutSeconds: 5
          successThreshold: 1
          failureThreshold: 5
        readinessProbe:
          httpGet:
            path: /readiness
            port: 8081
            scheme: HTTP
          # we poll on pod startup for the Kubernetes master service and
          # only setup the /readiness HTTP server once that's available.
          initialDelaySeconds: 3
          timeoutSeconds: 5
        args:
        - --domain=cluster.local.
        - --dns-port=10053
        - --config-dir=/kube-dns-config
        - --v=2
        env:
        - name: PROMETHEUS_PORT
          value: "10055"
        ports:
        - containerPort: 10053
```

```
          name: dns-local
          protocol: UDP
        - containerPort: 10053
          name: dns-tcp-local
          protocol: TCP
        - containerPort: 10055
          name: metrics
          protocol: TCP
        volumeMounts:
        - name: kube-dns-config
          mountPath: /kube-dns-config
      - name: dnsmasq
        image: gcr.io/google_containers/k8s-dns-dnsmasq-nanny-amd64:1.14.1
        livenessProbe:
          httpGet:
            path: /healthcheck/dnsmasq
            port: 10054
            scheme: HTTP
          initialDelaySeconds: 60
          timeoutSeconds: 5
          successThreshold: 1
          failureThreshold: 5
        args:
        - -v=2
        - -logtostderr
        - -configDir=/etc/k8s/dns/dnsmasq-nanny
        - -restartDnsmasq=true
        - --
        - -k
        - --cache-size=1000
        - --log-facility=-
        - --server=/cluster.local/127.0.0.1#10053
        - --server=/in-addr.arpa/127.0.0.1#10053
        - --server=/ip6.arpa/127.0.0.1#10053
        ports:
        - containerPort: 53
          name: dns
          protocol: UDP
        - containerPort: 53
          name: dns-tcp
          protocol: TCP
        # see: https://github.com/kubernetes/kubernetes/issues/29055 for details
        resources:
          requests:
            cpu: 150m
            memory: 20Mi
        volumeMounts:
        - name: kube-dns-config
          mountPath: /etc/k8s/dns/dnsmasq-nanny
      - name: sidecar
        image: gcr.io/google_containers/k8s-dns-sidecar-amd64:1.14.1
        livenessProbe:
          httpGet:
            path: /metrics
            port: 10054
            scheme: HTTP
          initialDelaySeconds: 60
          timeoutSeconds: 5
          successThreshold: 1
          failureThreshold: 5
        args:
        - --v=2
        - --logtostderr
        - --probe=kubedns,127.0.0.1:10053,kubernetes.default.svc.cluster.local,5,A
        - --probe=dnsmasq,127.0.0.1:53,kubernetes.default.svc.cluster.local,5,A
        ports:
        - containerPort: 10054
          name: metrics
          protocol: TCP
        resources:
          requests:
            memory: 20Mi
            cpu: 10m
      dnsPolicy: Default  # Don't use cluster DNS.
      serviceAccountName: readonly-addon
```

d. Create Dashboard addon manifest.

On each of the master nodes, customize the template below and copy to **/etc/kubernetes/addons/dashboard.yaml**:

```
---
apiVersion: v1
kind: Service
metadata:
  name: kubernetes-dashboard
  namespace: kube-system
  labels:
    k8s-app: kubernetes-dashboard
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
spec:
  selector:
    k8s-app: kubernetes-dashboard
  ports:
  - port: 80
    targetPort: 9090
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: kubernetes-dashboard
  namespace: kube-system
  labels:
    k8s-app: kubernetes-dashboard
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
spec:
  selector:
    matchLabels:
      k8s-app: kubernetes-dashboard
  template:
    metadata:
      labels:
        k8s-app: kubernetes-dashboard
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
        scheduler.alpha.kubernetes.io/tolerations: '[{"key":"CriticalAddonsOnly", 
"operator":"Exists"}]'
    spec:
      containers:
      - name: kubernetes-dashboard
        image: gcr.io/google_containers/kubernetes-dashboard-amd64:v1.5.1
        resources:
          # keep request = limit to keep this container in guaranteed class
          limits:
            cpu: 100m
            memory: 50Mi
          requests:
            cpu: 100m
            memory: 50Mi
        ports:
        - containerPort: 9090
        livenessProbe:
          httpGet:
            path: /
            port: 9090
          initialDelaySeconds: 30
          timeoutSeconds: 30
```

e. Add Grafana monitoring addon manifest.
   On each of the master nodes, customize the template below and copy to **/etc/kubernetes/addons/grafana-influx-monitoring.yaml**.

```
---
apiVersion: v1
kind: Service
metadata:
  name: monitoring-grafana
  namespace: kube-system
  labels:
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
    kubernetes.io/name: "Grafana"
spec:
  # On production clusters, consider setting up auth for grafana, and
  # exposing Grafana either using a LoadBalancer or a public IP.
  # type: LoadBalancer
  ports:
    - port: 80
      targetPort: 3000
  selector:
    k8s-app: influxGrafana
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: heapster-v1.3.0-beta.1
  namespace: kube-system
  labels:
    k8s-app: heapster
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
    version: v1.3.0-beta.1
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: heapster
      version: v1.3.0-beta.1
  template:
    metadata:
      labels:
        k8s-app: heapster
        version: v1.3.0-beta.1
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
        scheduler.alpha.kubernetes.io/tolerations: '[{"key":"CriticalAddonsOnly", 
"operator":"Exists"}]'
    spec:
      containers:
        - image: gcr.io/google_containers/heapster-amd64:v1.3.0-beta.1
          name: heapster
          livenessProbe:
            httpGet:
              path: /healthz
              port: 8082
              scheme: HTTP
            initialDelaySeconds: 180
            timeoutSeconds: 5
          command:
            - /heapster
            - --source=kubernetes.summary_api:''
            - --sink=influxdb:http://monitoring-influxdb:8086
        - image: gcr.io/google_containers/heapster-amd64:v1.3.0-beta.1
          name: eventer
          command:
            - /eventer
            - --source=kubernetes:''
            - --sink=influxdb:http://monitoring-influxdb:8086
        - image: gcr.io/google_containers/addon-resizer:1.7
          name: heapster-nanny
          resources:
            limits:
              cpu: 50m
              memory: 90Mi
            requests:
              cpu: 50m
              memory: 90Mi
          env:
            - name: MY_POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: MY_POD_NAMESPACE
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
          command:
            - /pod_nanny
            - --cpu=80m
            - --extra-cpu=0.5m
            - --memory=140Mi
            - --extra-memory=4Mi
            - --threshold=5
            - --deployment=heapster-v1.3.0-beta.1
            - --container=heapster
            - --poll-period=300000
            - --estimator=exponential
        - image: gcr.io/google_containers/addon-resizer:1.7
          name: eventer-nanny
          resources:
            limits:
              cpu: 50m
              memory: 90Mi
            requests:
              cpu: 50m
              memory: 90Mi
          env:
            - name: MY_POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: MY_POD_NAMESPACE
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
          command:
            - /pod_nanny
            - --cpu=100m
            - --extra-cpu=0m
            - --memory=190Mi
            - --extra-memory=500Ki
            - --threshold=5
            - --deployment=heapster-v1.3.0-beta.1
            - --container=eventer
            - --poll-period=300000
            - --estimator=exponential
---
kind: Service
apiVersion: v1
metadata:
  name: heapster
  namespace: kube-system
  labels:
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
    kubernetes.io/name: "Heapster"
spec:
  ports:
    - port: 80
      targetPort: 8082
  selector:
    k8s-app: heapster
---
apiVersion: v1
kind: ReplicationController
metadata:
  name: monitoring-influxdb-grafana-v4
  namespace: kube-system
  labels:
    k8s-app: influxGrafana
    version: v4
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
spec:
  replicas: 1
  selector:
    k8s-app: influxGrafana
    version: v4
  template:
    metadata:
      labels:
        k8s-app: influxGrafana
        version: v4
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
        - image: gcr.io/google_containers/heapster-influxdb-amd64:v1.1.1
          name: influxdb
          resources:
            # keep request = limit to keep this container in guaranteed class
            limits:
              cpu: 100m
              memory: 500Mi
            requests:
              cpu: 100m
              memory: 500Mi
          ports:
            - containerPort: 8083
            - containerPort: 8086
          volumeMounts:
          - name: influxdb-persistent-storage
            mountPath: /data
        - image: gcr.io/google_containers/heapster-grafana-amd64:v4.0.2
          name: grafana
          env:
          resources:
            # keep request = limit to keep this container in guaranteed class
            limits:
              cpu: 100m
              memory: 100Mi
            requests:
              cpu: 100m
              memory: 100Mi
          env:
            # This variable is required to setup templates in Grafana.
            - name: INFLUXDB_SERVICE_URL
              value: http://monitoring-influxdb:8086
              # The following env variables are required to make Grafana accessible via
              # the kubernetes api-server proxy. On production clusters, we recommend
              # removing these env variables, setup auth for grafana, and expose the grafana
              # service using a LoadBalancer or a public IP.
            - name: GF_AUTH_BASIC_ENABLED
              value: "false"
            - name: GF_AUTH_ANONYMOUS_ENABLED
              value: "true"
            - name: GF_AUTH_ANONYMOUS_ORG_ROLE
              value: Admin
            - name: GF_SERVER_ROOT_URL
              value: /api/v1/proxy/namespaces/kube-system/services/monitoring-grafana/
          volumeMounts:
          - name: grafana-persistent-storage
            mountPath: /var
      volumes:
      - name: influxdb-persistent-storage
        emptyDir: {}
      - name: grafana-persistent-storage
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: monitoring-influxdb
  namespace: kube-system
  labels:
    kubernetes.io/cluster-service: "true"
    addonmanager.kubernetes.io/mode: Reconcile
    kubernetes.io/name: "InfluxDB"
spec:
  ports:
    - name: http
      port: 8083
      targetPort: 8083
    - name: api
      port: 8086
      targetPort: 8086
  selector:
    k8s-app: influxGrafana
```

f. Add Docker registry addon manifest.
On each of the master nodes, customize the template below and copy to **/etc/kubernetes/addons/registry.yaml**. This creates a Docker registry where you can push images.

Note: If you already have a Docker registry, you can skip this step.
This registry is set up with an **empty-dir** mount which means that data will be lost if the pod is restarted or moved. To persist data, you will need to back the **image-store** volume with a clustered filesystem.

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: kube-registry-v0
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: kube-registry
      version: v0
  template:
    metadata:
      labels:
        k8s-app: kube-registry
        version: v0
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: registry
        image: registry:2
        resources:
          limits:
            cpu: 100m
            memory: 100Mi
        env:
        - name: REGISTRY_HTTP_ADDR
          value: :5000
        - name: REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY
          value: /var/lib/registry
        volumeMounts:
        - name: image-store
          mountPath: /var/lib/registry
        ports:
        - containerPort: 5000
          name: registry
          protocol: TCP
      volumes:
      - name: image-store
        # Update to volume claim
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: kube-registry
  namespace: kube-system
  labels:
    k8s-app: kube-registry
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: "KubeRegistry"
spec:
  selector:
    k8s-app: kube-registry
  type: NodePort
  ports:
  - name: registry
    port: 5000
    nodePort: 30000
    protocol: TCP
```

g. Add an ingress controller manifest
On each of the master nodes, customize the template below and copy to **`/etc/kubernetes/addons/nginx-ingress.yaml`**:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: default-http-backend
  labels:
    k8s-app: default-http-backend
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        k8s-app: default-http-backend
    spec:
      terminationGracePeriodSeconds: 60
      containers:
      - name: default-http-backend
        # Any image is permissable as long as:
        # 1. It serves a 404 page at /
        # 2. It serves 200 on a /healthz endpoint
        image: gcr.io/google_containers/defaultbackend:1.0
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
        ports:
        - containerPort: 8080
        resources:
          limits:
            cpu: 10m
            memory: 20Mi
          requests:
            cpu: 10m
            memory: 20Mi
---
apiVersion: v1
kind: Service
metadata:
  name: default-http-backend
  namespace: kube-system
  labels:
    k8s-app: default-http-backend
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    k8s-app: default-http-backend
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: nginx-ingress-controller
  labels:
    k8s-app: nginx-ingress-controller
  namespace: kube-system
spec:
  template:
    metadata:
      labels:
        k8s-app: nginx-ingress-controller
    spec:
      # hostNetwork makes it possible to use ipv6 and to preserve the source IP correctly 
regardless of docker configuration
      # however, it is not a hard dependency of the nginx-ingress-controller itself and it may 
cause issues if port 10254 already is taken on the host
      # that said, since hostPort is broken on CNI 
(https://github.com/kubernetes/kubernetes/issues/31307) we have to use hostNetwork where CNI 
is used
      # like with kubeadm
      hostNetwork: true
      terminationGracePeriodSeconds: 60
      containers:
      - image: gcr.io/google_containers/nginx-ingress-controller:0.9.0-beta.2
        name: nginx-ingress-controller
        readinessProbe:
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
        livenessProbe:
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
          initialDelaySeconds: 10
          timeoutSeconds: 1
        ports:
        - containerPort: 80
          hostPort: 80
        - containerPort: 443
          hostPort: 443
        env:
          - name: POD_NAME
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: POD_NAMESPACE
      valueFrom:
        fieldRef:
          fieldPath: metadata.namespace
    args:
    - /nginx-ingress-controller
    - --default-backend-service=$(POD_NAMESPACE)/default-http-backend
```

h. Create the addons.
On one of the master nodes, use kubectl to create the addons by entering the following commands:

```
/usr/bin/kubectl apply -f /etc/kubernetes/addons/addon-manager.yaml
/usr/bin/kubectl apply -f /etc/kubernetes/addons/dns-addon.yaml
/usr/bin/kubectl apply -f /etc/kubernetes/addons/dashboard.yaml
/usr/bin/kubectl apply -f /etc/kubernetes/addons/grafana-influx-monitoring.yaml
/usr/bin/kubectl apply -f /etc/kubernetes/addons/registry.yaml
/usr/bin/kubectl apply -f /etc/kubernetes/addons/nginx-ingress.yaml
```

i. Get cluster status.
Run the following command on a master node to see all the addons have been created:

```
kubectl cluster-info
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Build and deploy a sample application to a Kubernetes container

About this task

Once you have set up a Kubernetes container runtime environment, build and deploy a small LoopBack application to confirm that it works properly.

Procedure

1. Get a simple LoopBack project.
   If you don't have a simple "Hello World" LoopBack project handy, get the LoopBack "Getting Started" app. On one of the master nodes, enter the following commands:

   ```
   git clone https://github.com/strongloop/loopback-getting-started
   cd loopback-getting-started
   ```

   Alternatively, you can create a new LoopBack project from scratch. For more information, see Tutorial: Creating a LoopBack® project from the command line.

2. Create a Dockerfile.
   Create a file name `Dockerfile` in the Loopback application directory:

   ```
   FROM node:slim
   ADD . /app
   WORKDIR /app
   RUN npm install
   CMD npm start
   ```

3. Build and publish the image.
   Build the Docker image and tag it to publish to your Docker registry by entering this command:

   ```
   docker build -t 127.0.0.1:30000/apps/testapp .
   ```

   Once the image is built, publish it to the registry by entering this command:

   ```
   docker push 127.0.0.1:30000/apps/testapp
   ```

4. Create a deployment manifest for the application named app.yml in your home directory:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: testapp-deployment
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: testapp
    spec:
      containers:
      - name: testapp
        image: 127.0.0.1:30000/apps/testapp
        ports:
        - containerPort: 3000
---
kind: Service
apiVersion: v1
metadata:
  name: testapp-service
spec:
  selector:
    app: testapp
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: testapp-ingress
  annotations:
    ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testapp
        backend:
          serviceName: testapp-service
          servicePort: 3000
```

5. Test the deployed application.
   Once deployed, the application will be available at https://127.0.0.1/testapp. Enter the following command to retrieve data from the deployed app:

   ```
   curl -k https://127.0.0.1/testapp/api/CoffeeShops
   ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Setting up a Docker Swarm runtime environment

You can use Docker containers to run your APIs and applications being managed by API Connect. 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. For the latest information, refer to https://www.docker.com/.

Before you begin

Before setting up a Docker Swarm, you need to determine its topology. A topology includes information about how many manager nodes and how many worker nodes the developer will have in the Swarm cluster. A topology also needs to determine the number of Docker registries and how these nodes are wired up with a virtual network. A cluster may be divided by multiple overlay networks. Each overlay network is isolated from other networks by default.

Note: The procedures in this article are for Docker version 1.13 and Docker Machine version 0.9.0. They may not work with other versions.

About this task

Docker automates the deployment of applications inside software containers. Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run. Docker Swarm mode provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine. For more information, see https://www.docker.com. You can use Docker containers to run your APIs and applications being managed by API Connect.

Docker Swarm provides high availability for worker nodes as well as manager nodes. Swarm dispatches the containers among all nodes registered to a Swarm cluster with a fair algorithm. However, the number of containers usually depends on the traffic and the system’s degree of high availability.

Docker Swarm provides high availability for manager nodes using a Raft algorithm, which needs an odd number of nodes registered as manager (3, 5, 7 and so on). This algorithm selects a leader among different manager nodes. Having three managers in the cluster provides a fault tolerance of one node, and having five nodes provides a tolerance of up to two nodes.

See https://github.com/ibm-apiconnect/docker-swarm-setup for the scripts used in this article

Setting up Docker Swarm

Procedure

1. Install the Docker machine on all of the nodes (both manager and worker nodes).
    a. Update the application manager list.
    On Ubuntu, enter this command:

    ```
    $ sudo apt-get update
    ```

    On Red Hat/Centos, enter this command

    ```
    $ sudo yum update
    ```

    b. Install and run docker-engine by entering the following command:

    ```
    $ sudo curl -sSL https://get.docker.com/ | sh
    ```

2. Determine if the Docker daemon is running by entering this command:

    ```
    $ sudo docker info
    ```

    If the Docker daemon is not running, enter the following command to configure and start the Docker engine to listen for Swarm nodes on port 2377:

    ```
    $ sudo docker daemon -H tcp://0.0.0.0:2377 -H unix:///var/run/docker.sock
    ```

3. Initialize the Swarm cluster on the first manager Docker host and make this machine a manager, by entering the following command:

    ```
    $ sudo docker swarm init --advertise-addr <IP-address-of-manager>
    ```

4. Note: The initialize command returns different tokens for joining as a manager or a worker node.
    Now repeat steps 1 and 2 on all of the nodes, and then join them to the Swarm using the following commands.
    a. Add a worker to the Swarm by entering the following command:

    ```
    $ sudo docker swarm join-token worker
    ```

    This command returns the command to execute on a worker host; for example, on Ubuntu (the command may be different on other systems):

    ```
    $ sudo docker swarm join \
    --token SWMTKN-1-343oklx5y9zqwc129p4ttvgxeeexh49vp
    gslavcnqzipjmp2n4-0375hm0z4nto0218jbt08ga1n \
    Manager-IP-address:2377
    ```

    Where *Manager-IP-address* is the IP address of the Docker Swarm manager node.
    b. Add a manager to this Swarm, by entering the following command:

    ```
    $ sudo docker swarm join-token manager
    ```

    This command returns the command to execute on a manager host; for example:

    ```
    sudo docker swarm join \
    --token SWMTKN-1-343oklx5y9zqwc129p4ttvgxeeexh49
    vpgslavcnqzipjmp2n4-bnfuijeolp487mllvj719sjil \
    Manager-IP-address:2377
    ```

Where *Manager-IP-address* is the IP address of the Docker Swarm manager node.

5. Create an overlay network on the first manager by entering the following command:

   ```
   $ sudo docker network create -d overlay mynet
   ```

   Your Swarm is now ready to use. Now it is time to create an image and push it to the Docker registry.

Setting up the registry using self-signed certificates

About this task

The server that hosts your registry should be separate from the manager or worker node servers.

Procedure

1. Create a directory on the registry host:

   ```
   $ mkdir -p /etc/docker/certs.d/registry-host-FQDN:5000/
   ```

   Where *registry-host-FQDN* is the fully-qualified domain name of the registry host.
2. Make the directory you just created current working directory:

   ```
   $ cd /etc/docker/certs.d/registry-host-FQDN:5000/
   ```

   Where *registry-host-FQDN* is the fully-qualified domain name of the registry host.
3. Generate the certificate and key in the `certs` folder:

   ```
   $ openssl req -newkey rsa:4096 -nodes -sha256 -keyout domain.key -x509 -days 365 -out domain.crt
   ```

   Note: Make sure you use a fully-qualified domain name as common name (CN) when generating the certificate and key.
4. Start the Docker registry container on the server that will host your registry by entering this command:

   ```
   $ sudo docker run -d -p 5000:5000 \
   --restart=always --name registry \
   -v /etc/docker/certs.d/registry-host-FQDN:5000/:/certs \
   -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
   -e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
   registry:2
   ```

   Where *registry-host-FQDN* is the fully-qualified domain name of the registry host.
5. Instruct all Docker Swarm hosts to trust the certificate by copying the `domain.crt`file to the following location on all manager and worker hosts: `/etc/docker/certs.d/registry-host-FQDN:5000/ca.crt`, where *registry-host-FQDN* is the fully-qualified domain name of the registry host.

Building a Docker image for a LoopBack project

About this task

Once your microservice is ready, follow steps one through five below to package your API using `npm` and create a deployable image to run in a Docker Swarm runtime environment.
Note: You must perform these steps on the local machine where you develop LoopBack applications. It needs to have Docker Host and Docker Machine installed and connectivity to your Docker Registry.
Then follow steps six and seven to tag and push your image to an internal Docker registry. You can replace the last two steps with publishing your image to another public or private Docker registry such as Docker hub.

Procedure

1. Go into the directory containing your LoopBack app, for example, assuming it's called “microservice”:

   ```
   $ cd microservice
   ```

2. Package your LoopBack app:

   ```
   $ npm pack microservice
   ```

   This command creates a tar file called `microservice-1.0.0.tgz`.
3. Copy the `microservice-1.0.0.tgz` to the `microservice` directory.

   ```
   $ cp microservice-1.0.0.tgz microservice
   ```

   Note: The number at the end of the file name is based on the `version` property in the application's `package.json` file. Depending on the version of your application, you may see a different number.

4. Create a file named `Dockerfile` and add the following text to it:

```
FROM node:argon
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app/package
ADD microservice1-1.0.0.tgz /usr/src/app/
RUN npm install
EXPOSE 3000
ENV NODE_ENV production
CMD [ "npm", "start" ]
```

5. Build your image by entering the following command:

```
$ docker build -t microservice1 .
```

6. Tag your image by entering the following command:

```
$ docker tag microservice1 registry-host-FQDN:5000/ microservice1
```

   Where *registry-host-FQDN* is the fully-qualified domain name of the registry host.
7. Push the image to the registry by entering the following command:

```
$ docker push registry-host-FQDN:5000/microservice1
```

Building a Docker image for a Java application

About this task

The following instructions show how to create a Docker image for your Java API if you already have a JAR package. Alternatively, you could use a Maven plug-in to create and publish your image to a Docker registry.

Procedure

1. Go into the root directory of your Java application (the directory that contains the `pom.xml` file), for example, assuming it's called “microservice”:

```
$ cd microservice
```

2. Package your Java application:

```
mvn package
```

   This creates a `microservice-1.0.0.jar` file in the `target` subdirectory by default.
3. Copy `microservice1-1.0.0.jar` to the `microservice` directory.
   Note: The numeric suffix in the JAR file name is determined by the version specified in the application `pom.xml` file. This example assumes that it is "1.0.0".

```
$ cp microservice1-1.0.0.jar microservice
```

4. Create Dockerfile and add the following text to it:

```
FROM frolvlad/alpine-oraclejdk8:slim
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app/
ADD microservice1-1.0.0.jar /usr/src/app/
EXPOSE 8080
ENV ENV production
ENTRYPOINT ["java","-Djava.security.egd=file:/
dev/./urandom","-jar","/usr/src/app/microservice1-0.0.0.jar"]
```

5. Follow the steps five to seven in the previous section to build your image and push it to the image registry.

Adding Docker Swarm visualizer tool

About this task

Docker Swarm provides a visualization tool to show the topology of the cluster as well as information about containers. The visualization tool runs as an extra container that you can start as a new service in Docker Swarm.

Procedure

Start the service for the visualization tool by entering the following command in the first manager:

```
docker service create \
--name=viz \
--publish=8080:8080/tcp \
--constraint=node.role==manager \
--mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \manomarks/visualizer
```

Adding a health check for LoopBack application

About this task

Docker Swarm provides a mechanism for a health check of each container. If a container is terminated after a health check, Swarm starts a new container to replace the terminated one. Since a microservice runs in an isolated process, and Docker Swarm does not communicate with the process running inside a container, you can implement service health check by adding an additional endpoint in the API without exposing it to the API consumers in the Swagger file.
You can implement this endpoint in a LoopBack app by adding a new component as detailed below. Note that this REST endpoint is available only to the cluster. No API consumer can hit this endpoint since it is not listed in the Swagger file and the API is protected by DataPower.

Procedure

1. In your LoopBack project, edit `server/component-config.json` and add the following:

   ```
   {
   ...
     "node-docker-health": { }
   ...
   }
   ```

2. Install the `node-docker-health` package:

   ```
   npm install --save node-docker-health
   ```

   This command installs the package and updates the application dependencies.
3. Edit the Dockerfile and add the following line before building an image:

   ```
   HEALTHCHECK CMD curl --fail http://localhost:<APIPORT>/<RestApiRoot>/vitals/node-docker-health ||
   exit 1
   ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Managing microservices in Docker Swarm

About this task

Every microservice in API Connect is defined by an OpenAPI YAML file. Developer toolkit generates the API definition file when you create a new API.

Deploy microservices

Procedure

Create containers from the microservice image that you published to a Docker registry in Setting up a Docker Swarm runtime environment by using this command in the first manager:

```
$ docker service create --name microservice1-v1 \
--replicas=number-of-instances \
--network overlaynet-name -p3000:3000 registry-host-FQDN:5000/microservice1-v1
```

Where:

- *number-of-instances* is a positive integer that specifies the number of container instances you want.
- *overlaynet-name* is the name of the overlay network you created (`mynet` in the previous article).
- *registry-host-FQDN* is the fully-qualified domain name of the registry host.

When you deploy a service, Docker Swarm automatically deploys it in a way that can shift around which nodes are running the microservice. It may do this to avoid non-responsive Docker hosts, ease load on hosts, or for variety of other reasons. You can, however, enforce some rules about how a Docker Swarm deploys service containers to nodes. See Docker documentation for examples.

Publish API definitions to the API Connect management server

Before you begin

1. Create a LoopBack® project. For more information, see Tutorial: Creating a LoopBack project from the command line.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM® Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.

Procedure

1. In the APIs tab in the API Designer, click your API definition, go to the assemble tab and then click the invoke icon. A window appears on the right.
2. Replace the URL shown with the following URL:

   ```
   http://Manager-IP:3000$(request.path)$(request.search)
   ```

   Where *Manager-IP* is the Docker Swarm manager IP address
3. Set your catalog by copying and pasting the link you are given by the API management UI dashboard. The command looks like this:

   ```
   apic config:set catalog=apic-catalog://API-management-server/orgs/ibmcom-dev/catalogs/sb
   ```

   Where *API-management-server* is the hostname or IP address of the API management server.
4. Go to the command prompt and run the following command:

   ```
   apic products:publish definition/microservice1-product.yaml
   ```

Un-deploy a microservice

Procedure

Un-deploy your microservice by using the following command in the first manager to remove all containers belonging to a specific service in Docker Swarm:

```
$ docker service remove microservice1-v1
```

Upgrade a microservice

Procedure

1. Execute the following command to create the new version of the service:

   ```
   $ docker service create --name microservice1-v2 --replicas=4 
   --network mynet -p <new-port>:<newport> 
   registry-host-FQDN:5000/microservice1-v2
   ```

   registry-host-FQDN is the fully-qualified domain name of the registry host.

2. Change the Swagger invoke URL so that it routes to the new port specified in the above command.

Scale/rescale a microservice

Procedure

Schedule and create new containers for a given service, with the following command (for example, here scaling to five instances) :

```
$ docker service scale microservice1-v1=5
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Migrating LoopBack applications from collectives to containers

Instead of deploying to deprecated Collectives, you can deploy Loopback applications to Docker Swarm or Kubernetes container runtimes. The application's runtime code is packaged and deployed according to the nature of the target runtime. The gateway routes API calls to the internal endpoint hosted by the target runtime using the target URL value that is defined in the assembly section of the API. The assembly defines how the gateway handles calls to the API.

Procedure

1. Set up Containerized runtime environment, either Docker Swarm or Kubernetes.
   For more information, see Setting up a Docker Swarm runtime environment or Setting up a Kubernetes runtime environment.
2. Configure the API definition of the LoopBack application project as described below.
3. Deploy application to the containerized runtime.
4. Migrate API subscriptions, as described in Migrating subscribed users to a new Product version and Migrating application subscriptions to another Plan.
5. Deprecate or remove the application from API Connect Collectives.

Configuring API definitions for container runtimes

About this task

Loopback applications that are created by the `apic loopback` command have an OpenAPI definition YAML file in the definitions directory. The API definition contains an `invoke` policy that routes requests to the runtime. The configuration of the `Invoke` policy is in the `x-ibm-configuration` section of the YAML file.

Procedure

Set the `runtime-url` value in the application project's API definition to refer to endpoint at which the application is exposed in the runtime.
Here's an example of x-ibm-configuration taken from a Loopback application API definition YAML file:

```
x-ibm-configuration:
  testable: true
  enforced: true
  cors:
    enabled: true
  catalogs:
    apic-dev:
      properties:
        runtime-url: $(TARGET_URL)
    sb:
      properties:
        runtime-url: 'http://localhost:4001'
  assembly:
    execute:
      - invoke:
          target-url: $(runtime-url)$(request.path)$(request.search)
```

Here, the `invoke` appears on the last two lines in the example text, and is configured initially with this target URL value:

```
$(runtime-url)$(request.path)$(request.search)
```

The API gateway resolves the `runtime-url`, `request.path`, and `request.search` values when it processes API calls. The `runtime-url` value determines the host endpoint to which the gateway forwards API calls. This value is itself declared with an API property, that can be assigned a value for each of the catalogs in which the API is published.
In the example, the configuration assigns a value of `$(TARGET_URL)` in the `apic-dev` catalog and `http://localhost:4001` to the `sb` catalog.
Note: The `apic-dev` catalog name is special; it represents the catalog that is provided by the API Connect developer toolkit for local testing.
When you deploy the Loopback application code to a container runtime, you must package the Loopback application code and deploy it to the runtime, and set the `runtime-url` value in the API to refer to endpoint at which the application is exposed in the runtime. The API is then published to the API catalog as before.

Configuring Loopback applications for Docker Swarm

About this task

To configure a Loopback application for deployment in a Docker or Docker Swarm environment, you must package the Loopback application code as a Docker container, and set the API's target URL to the endpoint of the deployed Docker container or Swarm service. This URL is composed of the Docker host or load balancer address through which the container or service is exposed, with the port value corresponding with the Dockerfile or service definition.

Example

For a Docker service created with this command:

```
$ docker service create --name microservice1-v1 \
--replicas=number-of-instances \
--network overlaynet-name -p3000:3000 registry-host-FQDN:5000/microservice1-v1
```

Where:

- *number-of-instances* is a positive integer that specifies the number of container instances you want.
- *overlaynet-name* is the name of the overlay network you created.
- *registry-host-FQDN* is the fully-qualified domain name of the registry host.

The target URL value is:

```
http://:3000$(request.path)$(request.search)
```

Configuring Loopback applications for Kubernetes

About this task

To configure a Loopback application for deployment in a Docker or Docker Swarm environment, the Loopback application code must be packaged as a Docker container, and the API's target URL set to the exposed endpoint of the deployed Kubernetes service. This URL is composed of the external IP or load balancer IP address for the service, with the port value corresponding with the port described in the ports section of the service spec.

Example

For a Kubernetes service with the following spec:

```
"spec": {
    "selector": {
        "app": "MyApp"
    },
    "ports": [
        {
            "protocol": "TCP",
            "port": 3001,
            "targetPort": 9376,
            "nodePort": 30061
        }
    ],
    "clusterIP": "10.0.171.239",
    "loadBalancerIP": "78.11.24.19",
    "type": "LoadBalancer"
},
"status": {
    "loadBalancer": {
        "ingress": [
            {
                "ip": "146.148.47.155"
            }
        ]
    }
}
```

The target URL value is:

```
http://146.148.47.155:3001$(request.path)$(request.search)
```

Building a docker image for a Loopback application

About this task

When your microservice is ready, it is time to package your API by using a node package manager (npm) tool and create a deployable image to run in a Docker Swarm or Kubernetes runtime environment. Steps 1 - 5 cover this process. When it is complete, follow steps 6 and 7 to tag and push your image to an internal Docker registry. You can replace the last two steps with publishing your image to another public or private Docker registry such as Docker hub.

Procedure

1. Create a directory called “microservice”:

   ```
   $ mkdir microservice
   ```

2. Package your LoopBack app:

   ```
   $ npm pack microservice1
   ```

3. Copy the microservice-0.1.tgz to the microservice directory.
4. Create a Docker file by using the command `touch Dockerfile` then add the following content:

   ```
   FROM node:argon
   RUN mkdir -p /usr/src/app
   WORKDIR /usr/src/app/package
   ADD microservice1-1.0.0.tgz /usr/src/app/
   RUN npm install
   EXPOSE 3000
   ENV NODE_ENV production
   CMD [ "npm", "start" ]
   ```

5. Build your image:

   ```
   $ docker build -t microservice1
   ```

6. Tag your image:

   ```
   $ docker tag microservice1 <myregistryFQDN>:5000/microservice1
   ```

7. Push the image to the registry:

   ```
   $ docker push <myregistryFQDN>:5000/microservice1
   ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing API Connect collective

V5.0.6 and earlier Install the API Connect collective Member host and the API Connect collective controller. Add the API Connect collective to the Cloud Manager.

Before you begin

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

This task assumes that you installed IBM API Connect and that the host server can access the API Connect collective member package. This document provides the following instructions.

- Installing the collective controller
- Installing the collective member host
- Preparing to install multiple collectives
- Installing extra collectives to an existing installation

If you want to install more than one collective, you must provide a unique name. For instructions, see Preparing to install multiple collectives and Installing extra collectives to an existing installation.
Requirements:

- A host server with a supported operating system.
- Administrative privileges, such as root privileges, on the host machine.
- The Management server.
- A supported version of Node.js.
- C++ compiler, make, and Python.

For information about supported operating systems and Node.js versions, see the IBM API Connect Version 5.0 Detailed System Requirements report on the IBM Software Product Compatibility Reports site according to which API Connect offering you are using. Use the product component filter to specify the collective controller or collective member.

Installing the collective controller

Install the API Connect collective controller. Add the collective to the Cloud Manager.

Before you begin

As a prerequisite to installing the collective controller, you must install a supported version of Node.js.

For details of supported Node.js versions, follow the link on the IBM API Connect Version 5.0 requirements page to the IBM API Connect Software Product Compatibility Report for the API Connect offering you are using, then click the Prerequisites tab.

Node.js is available as part of the IBM SDK for Node.js.

About this task

Install and configure the API Connect collective controller host. The controller package comes with Liberty Profile and a JRE to run the API Connect collective and controller. The bundled packages come with the appropriate JRE for Liberty.
Note: Global installs with the `npm install -g` command might require root or administrator privileges or use of the `--unsafe-perm` parameter. In general, use the following guidelines based on how Node.js was installed:

- If Node.js is installed under a user account, then the controller and member installation commands do not require **sudo** or the `--unsafe-perm` argument.
- If Node.js and the controller and member installation commands are run under the root account, then the controller and member installation commands do not require **sudo** but do need to use the `--unsafe-perm` argument.
- If Node.js is installed under the `root` account but the controller or member installation commands are run from a non-root account, then the controller and member installation commands do require **sudo** and the `--unsafe-perm` argument.

Procedure

1. Download the IBM API Connect controller package from Passport Advantage®.
2. Install the collective controller:

```
[sudo] npm install -g [--unsafe-perm] /<path>/apiconnect-collective-controller-<platform>-
<version>.tgz
```

   where:
   - --unsafe-perm specifies that npm runs as the root account.
   - *<path>* is the path to the package.
   - *<platform>* is the operating system and processor architecture. The following values are valid:

    - darwin - Mac OS X
    - linux-x86_64 - Linux® 64-bit
    - linux-s390x - Linux z/OS®
    - linux-ppc64le - Linux Power® 64-bit Little-Endian
  - *<version>* is the current package version.

3. Setup the password:

   ```
   wlpn-controller setup --password=password [--keystorePassword=PASS] [...]
   ```

   where:
   - --password=*<PASS>*: Sets the administrative password of the controller. This password is used to login to the Admin Center, and is required to join members and host machines to the collective.
   - --user=*<USER>*: The administrative account name. Required to login to the Admin Center, and is required to join members and host machines to the collective.
   - --hostName=*<HOSTNAME>*: The hostname of the machine. The installation makes use of the available hostname for setting up the controller.
     Note: If the hostname is not a routable address, you must set this flag to a routable server name or IP address, for example, **`server.address.local`** or **`12.34.56.78`**.
   - --httpPort=*<PORT>*: The HTTP port that the controller listens on. The default value is 8080.
   - --httpsPort=*<PORT>*: The HTTPS port that the controller listens on. The default value is 9443.
   - --keystorePassword=*<PASS>*: The password for the PFX key that is used for HTTPS communication with other collective members. This certificate is generated by the Liberty Profile as a part of the controller installation. If the password is not set, it defaults to the password that is used for the administrative account.
     Note: Set a unique value for this option on setup.

4. Start the controller.

   ```
   wlpn-controller start
   ```

   When the controller starts, you can log into the controller from a browser. Log in to see the status of the controller and ensure that the configured ports are open for use.

   ```
   https://<controllerHostname>):<controllerPort>/adminCenter/
   ```

5. Add the API Connect collective to the Cloud Manager.
   a. Login to the Cloud Manager:

      ```
      https://<API_manager_hostname>/cmc
      ```

      where **`<API_manager_hostname>`** is the address of the API Manager.
   b. Click Services.
   c. Click Add > Add Collective, and enter a name for the collective.
   d. Click Create.
   e. Under the name of the API Connect collective that you added, click Add Controller and enter the controller name, IP address, port, user name, and password.
   f. Click Create.

Installing the collective member host

Install the collective member host.

Before you begin

The host on which you install the collective member must have an SSH daemon installed. You must be able to SSH into that host with the user name and password that are supplied to the **`wlpn-collective registerHost`** command by using the **`--rpcUser`** and **`--rpcPassword`** options; these options are separate from **`--user`** and **`--password`**, which are the credentials required to log in to the Admin Center.

You can install the collective member in any of the following ways:

- From the public npm registry.
- From a locally hosted IBM API Connect server.
- From a locally downloaded .tgz file.

If you are installing the collective member locally from an IBM API Connect server or a downloaded .tgz file, you must first install a supported version of Node.js

For details of supported Node.js versions, follow the link on the IBM API Connect Version 5.0 requirements page to the IBM API Connect Software Product Compatibility Report for the API Connect offering you are using, then click the Prerequisites tab.

Node.js is available as part of the IBM SDK for Node.js.

About this task

Install a member host to host your applications and the Micro Gateway.
Note: Global installs with the `npm install -g` command might require root or administrator privileges and use of the `--unsafe-perm` parameter; the `--unsafe-perm` parameter applies only when you are using the `sudo` command.

Procedure

1. Install the member host:
   - Install from the public npm registry:

     ```
     [sudo] npm install -g [--unsafe-perm] apiconnect-collective-member
     ```

   - Install from a locally hosted IBM API Connect server that was deployed from an OVA file with the OVF Tool:

     ```
     [sudo] npm install -g [--unsafe-perm] https://<host>/packages/apiconnect-collective-member
     ```

     where *<host>* is the host name or IP address of the IBM API Connect OVA.
   - Install from a locally downloaded .tgz file:

     ```
     [sudo] npm install -g [--unsafe-perm] /<path>/apiconnect-collective-member-<platform>-
     <version>.tgz
     ```

2. Register or update the host with the member:
   - If the member is hosted on a different host from the controller:

     ```
     wlpn-collective registerHost <HOSTNAME> --host=<name> --port=<num> --user=<name> --password=
     <pwd> [options]
     ```

   - If the member is hosted on the same host as the controller:

     ```
     wlpn-collective updateHost <HOSTNAME> --host=<name> --port=<num> --user=<name> --password=
     <pwd> [options]
     ```

   where:

   <HOSTNAME>
   : Required. The hostname corresponds to the member host machine. The value for this field is either the host name or the IP address of the member machine.

   --host=*<name>*
   : Required. The host name of the target collective controller.

   --port=*<num>*
   : Required. The HTTPS port number of the target collective controller.

   --user=*<name>*
   : Required. An Administrator user for the target collective controller. Required to log in to the Admin Center, and required to join members and host machines to the collective.

   --password=*<pwd>*
   : Required. The password for the Administrator user for the target collective controller.

   --rpcHost=*<name>*
   : Optional. The host on which the RPC or SSH mechanism is listening. Defaults to the *hostName*.

   --rpcPort=*<num>*
   : Optional. The port on which the RPC or SSH mechanism is listening. Defaults to SSH port 22.

   --rpcUser=*<name>*
   : Optional. The user with which to authenticate to the RPC or SSH mechanism. Defaults to the current operating system user.

   --rpcUserPassword=*<pwd>*
   : Optional. The password for the rpcUser. The default is to use SSH key authentication. Set this value if SSH is not supported for the host. Use only one authentication option, rpcUserPassword or sshPrivateKey, not both.

   --sshPrivateKey=*<path>*
   : Optional. The path to the SSH key to use to authenticate to the host. Defaults to a newly generated SSH key. Use only one authentication option, rpcUserPassword or sshPrivateKey, not both.

   --autoAcceptCertificates
   : Optional. Automatically trust SSL certificates during this command.
   If you setup the controller with the **`wlpn-controller setup --password=`*`password`*** command without other parameters, the controller uses a self-signed certificate. Then, if you run the **`wlpn-collective updateHost`** without the --autoAcceptCertificates, you get an error.

What to do next

If you are installing the Professional offering on-premises with the Micro Gateway, next install the IBM HTTP Server server or servers for load balancing.

Related reference

- IBM API Connect Version 5.0 requirements

Related information

- Software Product Compatibility Reports

Preparing to install multiple collectives

For an APIC Liberty collective being configured for the first time, you can provide a unique name that distinguishes it from existing or future collectives. Otherwise, the identical default name is applied to each new collective that you create.

Before you begin

The following instructions are to be completed before you enter **wlpn-controller start** and prior to registering the members.

About this task

To configure a collective with a unique name, you must create a file containing the unique name.

Procedure

1. From the CLI, change directories to collective by entering `-- cd /root/.liberty/wlp/usr/servers/controller/resources/collective`.
2. Create a file named "collective.name" and add a string that contains your collective's unique name. You can enter this command to accomplish the step: **echo "collective-<some-uniqueName>" > output; tr -d '\n' < output > collective.name**
3. Start the controller and then register the members.

Installing extra collectives to an existing installation

You can install extra collectives by creating a unique name to override the default collective name.

Before you begin

Ensure that the controller is started and functioning normally before you complete the steps.

About this task

Note: This task is to adjust an API Connect Liberty collective that is already configured and running, so that extra collectives can be created. The `apic-liberty-collective` uses a default collective name called defaultCollective. To create more collectives, you must assign a unique name to each.

Procedure

1. Change to the collective directory by entering **cd /root/.liberty/wlp/usr/servers/controller/resources/collective**.
2. Create a file called "collective.name", add to a string (for example, `collective-<some-uniqueName>`) and then save the file. The string "`collective-<some-uniqueName>`" replaces the default collective name, which is "`defaultCollective`". Be sure to replace "`<some-uniqueName>`" with an appropriate unique name for the given collective. Create the file by entering **echo "collective-<some-uniqueName>" > output; tr -d '\n' < output > collective.name**
3. Unregister the members.
4. Restart the controller by entering **wlpn-controller stop && wlpn-controller start**.
5. Register the members.
   Note: It takes several minutes before the members come into view.
6. Restart the DataPower® On Demand Router (ODR).
   a. Go to the DataPower WebGUI under On Demand Router.
   b. Disable, and then Enable the On Demand Router to **Apply** the object during the restart.

What to do next

Accomplish the steps on one controller to validate it comes back online. When the `odr-event` log target is enabled, you see:

- Before the change: `'/cell/defaultCollective/node/<hostname>.......`
- After the change: `'/cell/collective-<some_unique_name>/node/<hostname>......`

When the first collective is updated and you validate it is running as expected, proceed to do the other collectives.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect Controller commands

Use the following commands as needed to start, stop, or configure.

Start and stop the API Connect Controller

To start the controller:

```
wlpn-controller start
```

To stop the controller:

```
wlpn-controller stop
```

Return the version of the API Connect Controller

Returns the version of the controller in Semantic Versioning 2.0.0 format. For more information, see https://semver.org.

```
wlpn-controller version
```

Configure the API Connect Controller

The API Connect Controller provides `get` and `set` commands to retrieve and alter configuration values.

You can modify the following values:

- `user`: The administrative user for the API Connect Controller.
- `password`: The administrative user's password for the API Connect Controller.
- `httpsPort`: The HTTPS port the API Connect Controller listens on.
- `httpPort`: The HTTP port the API Connect Controller listens on.
- `rpcUser`: The username of a privileged account on the API Connect Controller's host machine.
- `rpcUserPassword`: The password for the privileged account on the API Connect Controller's host machine.
- `hostname`: The hostname of the API Connect Controller.
- `features`: The list of Liberty feature modules installed on the API Connect Controller. Read-only.

You can retrieve this list of API Connect Controller configuration values with the following CLI command:

```
wlpn-controller list
```

Some values cannot be retrieved with the `get` command, such as passwords. However, password fields can be changed to new values by using the `set` command.
Note: Users familiar with Liberty can find the relevant XML-based configurations in the *<serverDir>*/servers/controller directory. However, manual modification of these files is not supported by the API Connect Controller command line tool.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Finding the port number of an application running on a collective

You need the port number of an application's endpoint to be able to invoke it.

Before you begin

To complete this task, you must have installed and configured an API Connect collective and published an app. For more information, see Installing API Connect collective and Staging and publishing a project from the API Designer

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

Procedure

To find the port number of an application runnng on a collective, complete the following steps:

1. Log in to the machine on which your API Connect collective controller is hosted.
2. Run the wlpn server status command `wlpn server:status -p <server-name>`, where **`server-name`** is the name of the application instance under **`WLPN_USER_DIR`**
   If your application is running, the command will return a JSON object containing the **`pid`**, **`adminPort`** and **`appPort`** of your application. If the application is not running, you will receive an error.

Related tasks

- Configuring your DataPower Gateway and API Connect collective controller to communicate

Related information

- Staging and publishing a project from the API Designer

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing IBM HTTP Server

Install IBM® HTTP Server as a load balancer in front of the Micro Gateway instances.

Before you begin

Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

About this task

Install and configure the following components:

- Install IBM HTTP Server
- Configure IBM API Connect controller
- Register the IBM API Connect controller with the collective
- Configure IBM HTTP Server

Installing IBM HTTP Server

Install IBM HTTP Server

About this task

Download and use IBM Installation Manager to install IBM HTTP Server. The Installation Manager default installation directory depends on the operating system:

- On Linux®, the default directory is /opt/IBM/InstallationManager.
- On Windows, the default directory is C:\Program Files\IBM\Installation Manager.

Procedure

1. Create an IBM ID at the IBM ID registration website.
2. At the My IBM profile website, click IBM Sign in, and login with your IBM ID.
3. Download IBM Installation Manager.
    a. Go to the Download package section of the Installation Manager website.
    b. Click the IBM Fix Central link for the appropriate operating system, and follow the prompts to complete the download.

   The compressed installation file is downloaded with the following file name format: agent.installer.*platform_version*.zip
4. Extract the compressed file, and run the Installation Manager installation.
5. Start Installation Manager
    a. On Linux, run the **/opt/IBM/InstallationManager/eclipse/IBMIM** command.
    b. On Windows, click the IBM Installation Manager icon.
6. In Installation Manager, click File > Preferences > Repositories.
7. Click Add Repository, and enter the following URL: https://www.ibm.com/software/repositorymanager/V85WASIHSILAN
8. Enter your IBM ID credentials to connect to the repository.
9. When you are connected to the repository, close the preferences dialog.
10. Click the Install wizard.
11. Select the following packages to install: IBM HTTP Server for IBM WebSphere® Application Server and Web Server Plug-ins for IBM WebSphere Application Server
12. Follow the prompts to continue and to accept the license agreement.
13. Required: On the Package Group Name and Installation Directory dialog, modify the installation directory for IBM HTTP Server and for the Web Server Plug-ins to remove any spaces in the directory path.
    For example, use your home folder such as C:\Users\IBM_ADMIN\IBM.
14. Follow the prompts through the remaining dialog boxes to complete the installation.

Related information

- IBM HTTP Server
- IBM Installation Manager

Configuring IBM API Connect Profile Controller

On the IBM API Connect controller, generate the artifacts that are required by Web Server Plug-ins for IBM WebSphere Application Server.

About this task

Use the **wlpn-controller ihsSetup** command to generate the plugin-cfg.xml and the plugin-key.jks files that are used by the Web Server Plug-in. The generated configuration file contains an `<IntelligentManagement>` section that specifies the required properties for dynamic routing. The generated keystore file contains the required certificates for secure communications. The command copies the generated files to the user home directory.
The **wlpn-controller ihsSetup** command takes the following options:

- `--host=HOST` Required. The host name of the target collective controller.
- `--port=PORT` Required. The HTTPS port number of the target collective controller.
- `--user=USER` Required. An Administrator user for the target collective controller.
- `--password[=PASSWORD]` Required. The password for the Administrator user for the target collective controller. If no value is defined, you are prompted.
- `--keystorePassword=KEYSTORE_PASSWORD` Required. The password that is used to access the Java Keystore (JKS) file generated by this command. This keystore is required for secure communication between IHS and the collective controller.
- `--pluginInstallRoot=PATH` Required. Fully qualified path of the root directory of the Web Server Plug-ins on the web server host.
- `--webServerNames=NAME[,NAME]` Required. Comma-separated names of the web servers for which Web Server Plug-ins configuration files need to be generated. By default, the IBM HTTP Server server name is `webserver1`. The commands shown assume the default server name.
- `--certificateSubject=NAME` Optional. The DN for the generated SSL certificate. The default DN is `CN=<<value of --user argument>>,OU=client,O=ibm,C=us`

Procedure

1. Run the **wlpn-controller ihsSetup** command with the following options:
   ```
   wlpn-controller ihsSetup --host=HOST --port=PORT --user=USER
   --password=PASSWORD --keystorePassword=KEYSTORE_PASSWORD --pluginInstallRoot=PATH
   --webServerNames=NAME[,NAME,...] [--certificateSubject=NAME]
   ```
   The command generates the plugin-cfg.xml and the plugin-key.jks files in the current directory.

2. Create a directory called webserver1 in %PLUGIN_PATH%/WebSphere/Plugins/config on IBM HTTP Server. If you did not use the default web server name of `webserver1` in the `ihsSetup` command, create a directory with the name of your web server.
3. Copy the two generated files from the user home directory to the new directory, such as %PLUGIN_PATH%/WebSphere/Plugins/config/webserver1.

Example

The generated plugin-cfg.xml file resembles the following XML:

```
<?xml version="1.0"  encoding="UTF-8"?>
<Config  ASDisableNagle="false" AcceptAllContent="false"  AppServerPortPreference="HostHeader"
ChunkedResponse="false" FIPSEnable="false" IISDisableNagle="false"  IISPluginPriority="High"
IgnoreDNSFailures="false"  RefreshInterval="60" ResponseChunkSize="64"  SSLConsolidate="false"
TrustedProxyEnable="false"  VHostMatchingCompat="false">
<Log  LogLevel="DEBUG" Name="C:\Users\IBM_ADMIN\IBM\WebSphere\Plugins\logs\webserver1\http_plugin.log"/>
<Property  Name="ESIEnable" Value="true"/>
<Property  Name="ESIMaxCacheSize" Value="1024"/>
<Property  Name="ESIInvalidationMonitor" Value="false"/>
<Property  Name="ESIEnableToPassCookies" Value="false"/>
<Property  Name="PluginInstallRoot"  Value="C:/Users/IBM_ADMIN/IBM/WebSphere/Plugins/"/>
<!-- Configuration generated  using httpEndpointRef=defaultHttpEndpoint-->
<!-- The default_host  contained only aliases for endpoint defaultHttpEndpoint.
The generated  VirtualHostGroup will contain only configured web server ports:
webserverPort=80
webserverSecurePort=443  -->
<Property  Name="Keyfile"  Value="C:/Users/IBM_ADMIN/IBM/WebSphere/Plugins/config/webserver1/plugin-
key.kdb"/>
<Property Name="Stashfile"  Value="C:/Users/IBM_ADMIN/IBM/WebSphere/Plugins/config/webserver1/plugin-
key.sth"/>
<IntelligentManagement>
<TraceSpecification  name="default" specification=":DEBUG"/>
<Property  name="webserverName" value="webserver1"/>
<ConnectorCluster  enabled="true" maxRetries="-1" name="default"  retryInterval="60">
<Property  name="uri" value="/ibm/api/dynamicRouting"/>
<Connector  host="localhost" port="9443" protocol="https">
<Property  name="keyring" value="C:/Users/IBM_ADMIN/IBM/WebSphere/Plugins/config/webserver1/plugin-
key.kdb"/>
</Connector>
</ConnectorCluster>
</IntelligentManagement>
</Config>
```

Registering the IBM API Connect controller with the collective

Register the IBM API Connect controller with the collective.

About this task

Use the **wlpn-controller ihsRegister** command to register the configured IHS instance so that it is known to the collective.
The **wlpn-controller ihsRegister** command takes the following options:

- `--host=`*`HOST`* Required. The host name of the target collective controller.
- `--port=`*`PORT`* Required. The HTTPS port number of the target collective controller.
- `--user=`*`USER`* Required. An Administrator user for the target collective controller.
- `--password[=`*`PASSWORD`*`]` Required. The password for the Administrator user for the target collective controller. If no value is defined, you are prompted.
- `--ihsIp=`*`IP`* Required. The externally routable IP address of an IHS instance.
- `--ihsPort=`*`PORT`* Required. The externally routable port of an IHS instance.

Procedure

1. On the IBM API Connect controller, go to the %LibertyFolder%/wlp/bin directory.
2. Run the **wlpn-controller ihsRegister** command with the following options:
   ```
   wlpn-controller ihsRegister --host=HOST --port=PORT --user=USER
   --password=PASSWORD --ihsIp=IP --ihsPort=PORT
   ```

Configuring IBM HTTP Server

Configure the Web Server Plug-in on IBM HTTP Server to use the configuration and keystore files that were generated by the IBM API Connect server.

About this task

The Web Server Plug-in on IBM HTTP Server uses the configuration data in the plugin-cfg.xml file to route requests. Use the **LoadModule** command to enable the Web Server Plug-in, and use the **WebSpherePluginConfig** command to specify the location of the plugin-cfg.xml file.
Use the **gskcmd** command to convert the keystore to a format that the Web Server Plug-in can use.

Procedure

1. Edit the %IHS_Installed_Path%/HTTPServer/conf/httpd.conf configuration file, and append the following lines at the end of the file. Then, save the file.
    a. For Windows, append the following lines:
    ```
    LoadModule was_ap22_module %PLUGIN_PATH%/WebSphere/Plugins/bin/32bits/mod_was_ap22_http.dll
    WebSpherePluginConfig %PLUGIN_PATH%/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
    ```
    b. For Linux where IHS is installed at /opt/IBM/HTTPServer and the Web Server Plug-in is installed at /opt/IBM/WebSphere/Plugins, append the following lines:
    ```
    LoadModule was_ap22_module /opt/IBM/WebSphere/Plugins/bin/64bits/mod_was_ap22_http.so
    WebSpherePluginConfig /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
    ```
2. Create an empty file: %PLUGIN_PATH%/WebSphere/Plugins/logs/webserver1/http_plugin.log
3. From the %IHS_Installed_Path%/HTTPServer/bin directory, run the following commands to convert the keystore:
    ```
    gskcmd -keydb -convert -stash -pw <password> -db
    %PLUGIN_PATH%/WebSphere/Plugins/config/webserver1/plugin-key.jks -old_format jks -target
    %INSTALLED_FOLDER%/WebSphere/Plugins/config/webserver1/plugin-key.kdb -new_format
    cms
    gskcmd -cert -setdefault -pw <password> -db
    %PLUGIN_PATH%/WebSphere/Plugins/config/webserver1/plugin-key.kdb -label default
    ```

    Make sure that your present working directory is in the PATH or the command returns a `Command not found.` error. For this case, use `./` before the `gskcmd` command to include the present working directory in the search path.

    For the *<password>* value, specify the password for the keystore, or if the default password was used, specify `WebAS`. For more information, see the key management utility topic in IBM HTTP Server.
    These commands create the following files in %PLUGIN_PATH%/WebSphere/Plugins/config/webserver1/: plugin-key.rdb, plugin-key.sth, and plugin-key.kdb.

4. Start, or restart, the IBM HTTP Server.
    a. For Windows, use the **httpd.exe** command, which is in the bin directory of the IBM HTTP Server installation:
    ```
    httpd.exe -w -n "IBM HTTP Server V8.5" -k [start | restart]
    ```
    b. For Linux, use the Apachectl utility with one of the following commands:
    ```
    sudo ./apachectl -k [start | restart]
    sudo %IHS_DIR%/bin/apachectl -k [start | restart]
    ```

What to do next

If you are installing the Professional offering on-premises with the Micro Gateway, next install the Micro Gateway.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing the Micro Gateway for the Professional and Enterprise offering

Install the Micro Gateway and add it to the Catalog.

Before you begin

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
This instruction applies to IBM API Connect Professional and Enterprise offerings. For instructions to install the Micro Gateway for IBM API Connect Essentials, see Installing the Micro Gateway for the Essentials offering.

This task assumes that you installed API Connect collective and registered the Member host with the collective, and that you added the API Connect collective to the Cloud Manager; for details, see Installing API Connect collective.

As a prerequisite to installing the Micro Gateway, you must ensure that a supported version of Node.js is installed.

For details of supported Node.js versions, follow the link on the IBM API Connect Version 5.0 requirements page to the IBM API Connect Software Product Compatibility Report for the API Connect offering you are using, then click the Prerequisites tab.

Node.js is available as part of the IBM SDK for Node.js.

About this task

Create and publish the Micro Gateway.

Procedure

1. Add the Micro Gateway as an application:
   a. Start the IBM API Connect API Manager user interface.

      ```
      https://<API_manager_hostname>/apim
      ```

      Note: To be able to log in to the Management server to use the API Manager user interface, you must either be the owner of a provider organization or have been added to a provider organization; for more information, see Creating a provider organization account and Adding users and assigning roles.

   b. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰.

      The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
   c. On the Dashboard page, click Add > App.
   d. Enter a display name and name for the application (`microgateway` for example), select the collective that you want to publish the Micro Gateway to, then click Add .
   e. Click the Show app identifier icon 🔗 and copy the command string from the App Identifier window.

   For more information about Apps in the API Manager user interface, see Working with Apps.

2. Set the environment for the deployment by using the developer toolkit CLI.
   For details on installing and using the developer toolkit CLI, see Working with the toolkit.
   a. Open a command window.
   b. Paste and run the command that you copied in step 1.e; the command has the following form:

   ```
   apic config:set app=apic-app://API_manager_hostname/orgs/org_name/apps/app_name
   ```

   For example:

   ```
   apic config:set app=apic-app://myhost.com/orgs/myorg/apps/microgateway
   ```

3. Change to the directory that contains API Connect and create the Micro Gateway by entering the command.

   ```
   apic microgateway
   ```

4. When prompted, enter an application project name for the Micro Gateway; for example, **microgw-app**.
5. When prompted, enter a folder name to be created for the Micro Gateway application project; for example, **microgw-app**.
6. Create the public and private keys and copy them into the Micro Gateway application project folder. The following commands use OpenSSL, an open source implementation of the SSL and TLS protocols.
   a. Create the RSA private key.

      ```
      openssl genrsa -out id_rsa 4096
      ```

   b. Create the public key.

      ```
      openssl rsa -in id_rsa -outform PEM -pubout -out id_rsa.pub
      ```

   c. Copy the key files to the Micro Gateway application project folder.
7. Optional: Configure any user-defined policies; for more information, see Packaging and importing your policies into IBM API Connect.
8. Publish the Micro Gateway.
   a. Log in to your IBM API Connect Management server.

```
apic login
```

Specify the host name and login details for your IBM API Connect Management server. If you do not know the host name of your Management server, contact your IBM API Connect cloud administrator. To be able to log in to the Management server, you must either be the owner of a provider organization or have been added to a provider organization; for more information, see Creating a provider organization account and Adding users and assigning roles.

b. Ensure that you are in the Micro Gateway application project folder.
c. Publish the gateway.

```
apic apps:publish
```

What to do next

Add the Micro Gateway to a Catalog.
For further information about the Micro Gateway, see the following subtopics:

- **Adding a Micro Gateway to a Catalog**
  If you are using the Micro Gateway, then to be able to route calls to APIs that are published to a Catalog, you must add the Micro Gateway to the Catalog.
- **Micro Gateway environment variables**
  Use environment variables to customize the Micro Gateway configuration.
- **Logging**
  Use logging on the Micro Gateway to log information at various levels to files or to the console.
- **Troubleshooting the Micro Gateway**
  Troubleshooting topics for the Micro Gateway.
- **Installing the Micro Gateway for the Essentials offering**
  IBM API Connect Essentials uses the Micro Gateway to provide connectivity with a routing web server device like IBM HTTP Server (IHS) or Apache.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a Micro Gateway to a Catalog

If you are using the Micro Gateway, then to be able to route calls to APIs that are published to a Catalog, you must add the Micro Gateway to the Catalog.

Before you begin

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

About this task

You add the Micro Gateway to a Catalog by using the API Manager user interface.

For more information on Catalogs, see Working with Catalogs.

Procedure

To add a Micro Gateway to a Catalog, complete the following steps:

1. If not already logged in, log in to the API Manager user interface.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. On the Dashboard page, click the Catalog to display details.
4. Click Settings, go to the Gateway section, then click Configure Gateway.

5. If you are using the Micro Gateway with the IBM API Connect Professional or Enterprise offering, complete the following steps:
    a. Select the API Connect collective to which you published the Micro Gateway application in Installing the Micro Gateway for the Professional and Enterprise offering, and click Next.
    b. Select the Micro Gateway application; this is the Micro Gateway application name that you specified in Installing the Micro Gateway.
    c. Paste the public key of the Micro Gateway, and click Done.
    You can obtain the public key from the id_rsa.pub file that you created in Installing the Micro Gateway.
6. V5.0.2 + If you are using the stand-alone Micro Gateway with the IBM API Connect Essentials offering, complete the following steps:
    a. Select API Connect Standalone Micro Gateway, and click Next.
    b. Paste the public key of the Micro Gateway application, and click Done.
    You can obtain the public key from the id_rsa.pub file that you created in Installing the Micro Gateway for the Essentials offering.
    c. Save the Catalog by clicking the Save icon 💾.
    d. Click the Download Configuration File link to obtain the YAML formatted environment manifest file env.yaml; this file contains information that is required by API Connect to communicate with the Catalog.
    e. Place the file in your API Connect application project folder.
7. Click Endpoints, then, in the Custom Gateway URL field, enter the address of IBM HTTP Server or other load balancer for the API traffic.
8. Save the Catalog by clicking the Save icon 💾.

Related tasks

- Installing the Micro Gateway for the Professional and Enterprise offering
- Installing the Micro Gateway for the Essentials offering

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Micro Gateway environment variables

Use environment variables to customize the Micro Gateway configuration.

Micro Gateway environment variables

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
Environment variables can be used to customize the Micro Gateway configuration and behavior. Environment variables might apply to only the developer toolkit environment, to the on-premises environment, or to both environments.

Table 1. Micro Gateway environment variables

Name	Description	developer toolkit	On-premises
APIC_CONFIG_PATH	Specifies the directory path containing the system logging configuration file, logger-config.json.	Applicable	Not applicable
APIC_LOG_CONFIG_FILE	Specifies the path of the user-level logging configuration.	Applicable	Not applicable
APIC_LOG_CONSOLE_LEVEL	Specifies the lowest level of messages logged to the console stream. Use the following valid values: • **fatal** • **error** • **warn** • **info** • **debug** • **trace** The default value is **warn**.	Applicable	Not applicable

Name	Description	developer toolkit	On-premises
`APIC_LOG_FILE_LEVEL`	Specifies the lowest level of messages logged to the apic.log log file. Use the following valid values: • `fatal` • `error` • `warn` • `info` • `debug` • `trace` The default value is `info`.	Applicable	Not applicable
`APIMANAGER`	Specifies the host of the API Connect management server to connect to.	Not applicable	Applicable
`APIMANAGER_CATALOG`	Specifies the Catalog that the Micro Gateway is responsible for servicing.	Not applicable	Applicable
`APIMANAGER_PORT`	Specifies the port to connect to the API Connect management server on.	Not applicable	Applicable
`APIMANAGER_REFRESH_INTERVAL`	Specifies, in milliseconds, the interval to wait before fetching updated artifacts from the API Connect management server. The default value is 15 minutes.	Not applicable	Applicable
`CATALOG_HOST`	Specifies the IP address and port of the gateway in *<IPAddress>*:*port* format. The value is used in OpenAPI (Swagger 2.0) in the `$(catalog.host)` variable for use by the explorer to locate the gateway. The default value is `localhost:`***`gatewayport`***.	Applicable	Not applicable
`CONFIG_DIR`	Specifies the directory of OpenAPI (Swagger 2.0) to load.	Not applicable.	Applicable. This value is used for the initial load. For subsequent loads, on-premises loads into ./config.
`DATASTORE_PORT`	Specifies the port that the data store listens on for requests. By default, the data store binds to port 0, causing an ephemeral port listen. If set before startup, the data store listens on that port.	Applicable	Applicable
`LAPTOP_RATELIMIT`	Specifies the rate limit per hour to apply for all requests. This environment variable must be set before starting the Micro Gateway. The default value is 100. For example, the following command sets the rate limit to 1000 requests per hour: `export LAPTOP_RATELIMIT=1000/hour`	Applicable	Not applicable
`PORT`	Specifies the port that the Micro Gateway listens on for incoming traffic. Specify a value in the range of 1024 - 64000. The default values are 80 for HTTP and 443 for HTTPS stand alone when started with Node.js. Note: To listen on ports less than 1024 (including the default ports), the gateway needs to be started by the root user. There is potential risk to run an application as the root user, therefore, it is advised to specify a value greater than 1024.	Applicable	Applicable

Name	Description	developer toolkit	On-premises

`TLS_SERVER_CONFIG`	Contains the path name, possibly relative, to a JSON file that contains the TLS configuration for the Micro Gateway to expect HTTPS communication.
	The Micro Gateway includes a self-signed cert for development. To use the self-signed certificate, when prompted, update the browser's trusted keystore to include the Micro Gateway self-signed certificate.
	The Micro Gateway does not include a trusted CA-signed certificate for production. Instead, use the TLS_SERVER_CONFIG environment variable to specify your own trusted CA-signed certificate for the Micro Gateway.
	The JSON file contents should be equivalent to the properties for **`tls.createServer(options[,secureConnectionListener])`** as specified by the Node.js Transport Layer Security documentation.
	One exception is that certain properties can specify file names with locations that are relative to the location of the **`TLS_SERVER_CONFIG`** file. The following properties can specify relative file names:
	- `key`
	- `cert`
	- `ca`
	- `pfx`
	- `dhparam`
	- `ticketKeys`
	For example:
	`{`
	`  "cert" : "./server-cert.pem",`
	`  "key" : "-----BEGIN RSA PRIVATE KEY-----\nMIICXgIBAAKBgQDT9ONI21tV/pVnHX/...tLgZU4Tw==\n-----END RSA PRIVATE KEY-----"`
	`}`
		Applicable
		When any of the OpenAPI (Swagger 2.0) API definitions include HTTPS within the schemes, the gateway starts listening for HTTPS traffic. If TLS_SERVER_CONFIG is not specified, a default self-signed cert is used.
			Applicable

`WLPN_APP_ROUTE`	For API Connect, specifies the organization and environment URL for client requests.	Not applicable	Applicable

Related information

- Node.js Transport Layer Security documentation

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Logging

Use logging on the Micro Gateway to log information at various levels to files or to the console.

Overview

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
Logging in the Micro Gateway is implemented through the Bunyan Node.js package. Bunyan has much flexibility for what can be logged and output formats. The following features are supported:

- Output to a file or to the console. By default, output is logged to a file in JSON format, and formatting is available through Bunyan.
- Creation of child loggers. Different components can have log entries that are labeled and filtered based on the component.
- Customizable streams for output.

Configuring logging

Micro Gateway logging options, such as file location and logging levels, are controlled by environment variables and configuration files.

The following environment variables are supported:

APIC_CONFIG_PATH
    Specifies the directory path that contains the system logging configuration file, logger-config.json.
APIC_LOG_CONFIG_FILE
    Specifies the path to the user-level logging configuration file.
LOG_LEVEL (deprecated APIC_LOG_CONSOLE_LEVEL)
    Specifies the lowest level of messages logged to the console stream.
FILE_LOG_LEVEL (deprecated APIC_LOG_FILE_LEVEL)
    Specifies the lowest level of messages logged to the apic.log log file.

For example, on Linux®, you can set the lowest level of message to log with the following command:

```
export FILE_LOG_LEVEL=warn
```

For more details about these variables and valid values, see the topic about Micro Gateway environment variables.
The default logging configuration is extensible to define various logging options. You can define three levels of configuration that are used with the following order of precedence:

- A user-defined log configuration file. This file can be JSON or JavaScript. If both files are defined, the JSON configuration file takes precedence over the JavaScript configuration file. This file location is specified with the APIC_LOG_CONFIG_FILE environment variable.
- The system log file named ~/.apiconnect/logger-config.json or ~/.apiconnect/logger-config.js. The default configuration is merged with the user-defined configuration, but the user-defined configuration takes precedence. The system configuration file is used if no user-defined configuration file exists. The location of the system configuration file is specified with the APIC_CONFIG_PATH environment variable.
- The built-in Bunyan default configuration, which logs to the console and to the log file. The console log is formatted and defaults to a log level of `warn` or as defined with the LOG_LEVEL environment variable. The stream log is unformatted and defaults to a 10 file, 1-day rotation, with a log level of `info` or as defined with the FILE_LOG_LEVEL environment variable.

In the configuration file, you can configure the `bunyan` logger and the default Bunyan `filelogger` stream.

For information about the available Bunyan configuration options, see the Bunyan README.md file, https://github.com/trentm/node-bunyan. The options are specified in the `bunyan` element of the configuration file.

The `filelogger` stream supports the following keys:

Table 1. Supported Bunyan `filelogger` keys

Key	Default value	Description
compress	`false`	Optionally, compress the rotated log files with gzip compression.
file	~/.apiconnect/apic.log	The path and log file name
keep	`10`	The number of log files to keep. This number includes the primary log file.
size	`50M`	The maximum size if the log file. When the file reaches this size, the log rotates to the next log file. The size key supports `1024`, `1K`, `1M`, and `1G`.

JSON configuration files must represent a valid Bunyan configuration object. Field values must have fixed values.

Sample JSON configuration files:

```
{
  "bunyan": {
    "name": "custom logger",
    "src": "false",
    "streams": [{
      "level": "debug",
      "type": "file",
      "path": "custom_file.log"
    }]
  },
  "filelogger": {
    "size": "1K",
    "file": "custom.log",
    "keep": 5
  }
}
```

```
{
  "bunyan": {
    "name": "custom bunyan name",
    "streams": [{
      "level": "error",
      "type": "rotating-file",
      "period": "1d",
      "count": "10",
      "path": "/tmp/.apiconnect/apic.log"
    }]
  }
}
```

JavaScript configuration files create Bunyan-compatible JSON configuration output and make it easier to specify complex streams. The exported output of a JavaScript configuration file must be a JSON object with the **bunyan** keyword and a value that represents a valid Bunyan configuration object. Field values, such as **path**, can be determined programmatically.

Sample JavaScript configuration files:

```
module.exports =
{
  bunyan: {
    name: 'userConnect',
    streams: [{
      level: 'debug',
      type: 'file',
      path: 'custom_file.log'
    }]
  },
  filelogger: {
    file: 'custom.log',
    size: '50M',
    keep: 10
  }
};
```

```
var osenv = require('osenv');
module.exports =
{
  bunyan: {
    name: 'custom bunyan name',
    streams: [{
      level: 'warn',
      type: 'rotating-file',
      period: '1d',
      count: '10',
      path: path.resolve(osenv.home(),'custom_file.log')
    }]
  }
};
```

Log levels

Log levels are controlled by the LOG_LEVEL and FILE_LOG_LEVEL environment variables or by methods used for the Bunyan logger object. Logging supports the following levels.

- debug
- info
- warn
- error
- fatal

For example, to change the logging level of the console stream, set **LOG_LEVEL** environment variable.

Default log output

By default, logs are streamed to the console and to the $APIC_CONFIG_PATH/apic.log file. If the environment variable APIC_CONFIG_PATH is not set, this log file is in the ~/.apiconnect/ directory.

The file logging stream consists of Bunyan JSON format, which is a good format to process programmatically. To see it in a readable format, you can run the `cat` command:

```
cat ~/.apiconnect/apic.log
```

Bunyan can be used to format the apic.log file to a more human-readable format.

```
cat ~/.apiconnect/apic.log | ../node_modules/.bin/bunyan
```

The `apic logs` command can be used to output a more compact formatted log to the console. The log will display the end of the output by default.
The console log stream uses the formatter stream that is specified in the lib/formatter.js file.

Logging in user-defined and JavaScript policies

User-defined and JavaScript policies can log messages to the system logger. The Bunyan logger object is passed into the flow engine object as parameters to the function. You can use the standard Bunyan logger methods.
For example, the log levels are exposed as methods on the Bunyan logger object, and the methods can be used, such as `logger.info(...)`, `logger.warn(...)`.

The following JavaScript snippet uses the `logger` object to log a message at the `info` log level.

```
module.exports = function ( config ){

  return function (props, context, flow){
    var logger = flow.logger;
    logger.info('ENTER mypolicy, params:', JSON.stringify(props));

    //... remainder of policy logic
  };
};
```

Log output is sent to the apic.log file and to output specified with the **apic logs** CLI command or to any other stream that is configured through the configuration file.

Flushing the log files

The logger provides methods to flush the logger. Call these methods before the CLI exits to ensure that all logs are written to disk. Both methods return a Promise object.

- `logger.flush()`
- `logger.exit(exitCode)`

Related tasks

- Installing the Micro Gateway for the Professional and Enterprise offering

Related reference

- Micro Gateway environment variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Troubleshooting the Micro Gateway

Troubleshooting topics for the Micro Gateway.

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

Specifying HTTP as scheme in an API might result in the API running on HTTPS.

You create an API in API Manager, add a path, and set an invoke policy target URL. Specify HTTP in the Schemes section of the Design tab. The OpenAPI (Swagger 2.0) document shows:

```
schemes:
  - http
```

Click Run > Start and you get an error, and the Micro Gateway does not start.

Solution

The Micro Gateway does not support both HTTP and HTTPS simultaneously. If on start, the Micro Gateway detects any OpenAPI (Swagger 2.0) document with HTTPS enabled, HTTPS is used for all APIs.

For a single instance of the Micro Gateway, all APIs must use either HTTP or HTTPS. Do not mix APIs that use HTTP with APIs that use HTTPS on the same Micro Gateway instance.

Authentication request timed out.

Basic Authentication to an HTTP server might return an error before the configured timeout. The Micro Gateway response to a client whose request cannot be completed due to the server authentication request timing out does not indicate that the cause is a timeout. You might receive an error similar to the following message:

```
Status Code
401:Error: Unauthorized.
```

Solution

The default timeout values for LDAP and Basic Authentication are not the same. The LDAP timeout is 60 seconds, and the Basic Authentication timeout is 120 seconds.

Unsupported Node.js versions might cause segmentation faults.

If unsupported Node.js versions are used, the Micro Gateway might fail with segmentation faults.

Solution

Ensure that a supported version of Node.js is installed. In particular, do not run Node.js version 5.9.0, which causes a known segmentation fault in the contextify code.
Check for the most recent software requirements for IBM API Connect Version 5.0. For more information, see the Software requirements section of the IBM API Connect Version 5.0 requirements.

How to replace an existing Micro Gateway with an updated version.

When you attempt to republish a Micro Gateway, instead of replacing the existing one, a second Micro Gateway is created.

Solution

To update the component, you must remove the old Micro Gateway, and republish it. For instructions, see Removing an application from a collective.

Related tasks

- Installing the Micro Gateway for the Professional and Enterprise offering

Related reference

- IBM API Connect Version 5.0 requirements

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing the Micro Gateway for the Essentials offering

IBM® API Connect Essentials uses the Micro Gateway to provide connectivity with a routing web server device like IBM HTTP Server (IHS) or Apache.

Before you begin

Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
As a prerequisite to installing the Micro Gateway, you must have access to SSL, and install IBM API Connect Essentials. For installation on the IBM API Connect Professional or Enterprise offering, see Installing the Micro Gateway for the Professional and Enterprise offering

For details of supported Node.js versions, follow the link on the IBM API Connect Version 5.0 requirements page to the IBM API Connect Software Product Compatibility Report for the API Connect offering you are using, then click the Prerequisites tab.

Node.js is available as part of the IBM SDK for Node.js.

For troubleshooting tips, see: Troubleshooting the Micro Gateway.

About this task

Create and publish the Micro Gateway.

Procedure

1. Change to the directory that contains API Connect and create the Micro Gateway by entering the command.

   ```
   apic microgateway
   ```

2. When prompted, enter an application project name for the Micro Gateway; for example, `microgw-app`.
3. When prompted, enter a folder name to be created for the Micro Gateway application project; for example, `microgw-app`.
4. Create the public and private keys and copy them into the Micro Gateway application project folder. The following commands use OpenSSL, an open source implementation of the SSL and TLS protocols.
   a. Create the RSA private key.

      ```
      openssl genrsa -out id_rsa 4096
      ```

   b. Create the public key.

      ```
      openssl rsa -in id_rsa -outform PEM -pubout -out id_rsa.pub
      ```

   c. Copy the key files to the Micro Gateway application project folder.
5. Add the Micro Gateway to a catalog by following the instructions at Adding a Micro Gateway to a catalog.
6. To control configuration at run time, place a YAML formatted environment manifest configuration file env.yaml into your application project folder. The env.yaml must contain the API Manager port, URL, and catalog ID. You can download an env.yaml configuration file specific to your catalog from the API Manager Gateway Settings user interface as described in Adding a Micro Gateway to a catalog.
7. Start the Micro Gateway by entering the command **node .** ensuring that you include the trailing period as part of the command.
   You can modify the command by entering an environment variable before the **node .** command. Environment variables that are entered in the command line supersede configuration settings in the env.yaml file. You can find available environment variables with their descriptions at Micro Gateway environment variables.

Results

The Micro Gateway is installed in your application project folder and configured to communicate with the catalog in API Manager.

What to do next

Continue to develop APIs and applications by creating a project in API Designer. For an overview of developing your APIs and applications, see Developing your APIs and applications.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using physical DataPower appliances as Gateway Servers

A physical appliance can serve as a Gateway server.

About this task

For a Gateway server, you can use any of the following physical appliances:

- IBM® WebSphere® DataPower® Service Gateway XG45
- IBM WebSphere DataPower Integration Appliance XI52
- IBM WebSphere DataPower B2B Appliance XB62
- IBM DataPower Gateway

Before you add a Gateway server, your physical appliance must be running the minimum level of firmware that is required to support API Connect. For more information, see IBM API Connect Version 5.0 requirements.

Procedure

For each Gateway server, complete the following steps:

a. If you ordered a new IBM DataPower physical appliance, configure the appliance according to the installation instructions provided with the appliance. For more information, see IBM WebSphere DataPower documentation.
b. Identify the IP address of the appliance.
c. Ensure that the appliance is powered up and active.
d. Configure your Gateway server by using the API Connect cloud console.
For more information, see Adding Gateway servers.
When you add the Gateway server, the API Connect configuration for IBM DataPower is deployed to the appliance.

Results

API Connect is installed in an application domain within the IBM DataPower appliance.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using virtual DataPower appliances as Gateway Servers

A virtual DataPower® appliance can serve as a Gateway server. To use a virtual appliance as a Gateway server, you must complete a number of configuration tasks. You then add the appliance to your API Connect topology by using the cloud console.

Before you begin

You will need to size and allocate resources to your virtual DataPower Gateway based on expected API Connect workload. If you are uncertain of your expected workload, then to achieve an acceptable level of performance in test, staging, and production environments, use the Standard configuration of 8 virtual processors (vCPU) and 16 GB RAM as the starting point. The absolute minimal configuration is 4 vCPU and 8 GB RAM and is meant primarily for sandbox and development environments. You can increase resource settings after deployment to meet workload requirements.

Table 1. Resource allocation for named configuration

Configuration name	vCPU	RAM (GB)
Small	4	8
Standard (default)	8	16
Enterprise	16	96

Procedure

To prepare your virtual DataPower appliance for use as an API Connect Gateway server, complete the following configuration tasks:

1. Initialize the appliance; see Initializing DataPower Gateway Virtual Edition.
2. Enable the XML management interface; see Enabling interface services.
3. Configure Network Time Protocol (NTP); see Managing NTP servers.
4. Set the time zone to be consistent with the Management servers in your API Connect topology; see Setting the local timezone.
5. Configure your Gateway server by using the IBM® API Connect cloud console.
   For more information, see Adding Gateway servers.
   When you add the Gateway server, the API Connect configuration for IBM DataPower is deployed to the appliance.

Results

IBM API Connect is installed in an application domain within the IBM DataPower virtual appliance.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deploying the Developer Portal OVA file

You create a Developer Portal node by deploying the Developer Portal OVA template. After you have deployed the Developer Portal OVA template, you can install the Developer Portal.

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 carry out specific administrative actions for only that node.

Important:

- **(IBM® API Connect V5.0.8.1 and later releases)** You must deploy the Developer Portal OVA template by using a version of the VMware vSphere Client that supports the SHA-256 Cryptographic Hash Algorithm.
- The Developer Portal node is initially configured with a default password of `!n0r1t5@C`, with the user name of `admin`. For security reasons, change the default password by completing one 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 logging 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, please ensure that you have typed the password correctly.

The Developer Portal OVA file is configured to use DHCP to obtain an IP address. If you choose to use DHCP, you might want to specify a MAC address, if your IP addresses are given out based on MAC addresses. This can be done in the OVA settings, and should be done before the machine is powered on for the first time.

After you have deployed your OVA file, you can install the Developer Portal; see Installing the Developer Portal.

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.

Procedure

1. Download a local copy of the Developer Portal OVA file.
   Important: Download the OVA file to a storage device that is accessible by the virtual infrastructure and where the software is to be installed.
2. In the VMware Infrastructure Client navigation pane, select the virtual machine on which to install the IBM API Connect software.
3. Select File > Deploy OVF Template.
   The Deploy OVF Template wizard is displayed.
4. Select the relevant location to Deploy from a file or URL.

Choose the file location where you downloaded the OVA file. You can enter a URL if you want to download the OVA file from the Internet.

5. Click Next, review the OVF template details, and click Next again.
6. Specify the name and location for the deployed template, and click Next.
7. Select the host or cluster on which to run the deployed template, and click Next.
8. Select a resource pool in which you want to deploy the template, and click Next.
9. Select the format in which you want to store the virtual disks, and click Next.
10. For each row in the table, click on the value under DestinationNetworks, and select the correct network. Map the networks that are used in the OVF template to networks in your inventory, and click Next.
11. Review your deployment settings, and click Finish. Unless you know you have completed all the required configuration, leave the "Power on after deployment" option deselected.
    The OVF template is deployed to your virtual machine.
12. Optional: Increase your virtual disk size. The Developer Portal is pre-configured with a 25 GB disk, containing a single partition managed by Logical Volume Manager.
    The minimum hardware specification for the Developer Portal is 4 GB RAM, 2 CPUs, and 25 GB disk space. However, this specification is recommended only for basic testing and development work. For production deployments the recommended specification is shown in the following table:

    Table 1. Developer Portal hardware requirements

Number of sites	Number of concurrent users	Number of CPUs	Amount of RAM (GB)	Disk Size (GB)
20	5	4	8	60
100	20	4	16	100
500	100	8	48	300

    If your sites contain a lot of data then you will need to increase your disk size further. If your concurrent users are highly active then you will need to configure more CPUs or RAM. To learn how to increase your virtual disk size, see Increasing your virtual disk size.
    Important: You must not modify your primary default disk to increase virtual disk size. You increase disk size by adding a new virtual disk.
13. If your Developer Portal is behind a firewall, the following ports must be open:
    - SSH: 22 (This is the port that the API Manager communicates with the Developer Portal via SSH on, and is the default value configured in the Cloud Manager user interface.)
    - HTTPS: 443 (This port must be open to the internet.)
    - To redirect from HTTP to HTTPS: 80

What to do next

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.

You can now install the Developer Portal. For more information, see Installing the Developer Portal.

- **Increasing your virtual disk size**
  The Developer Portal is pre-configured with a 10 GB disk , or 20 GB for API Connect version 5.0.2.0 and later, containing a single partition managed by Logical Volume Manager. You can increase the disk size by adding a new virtual disk.
- **Editing the config.ini file**
  After the Developer Portal is installed, you can configure settings in accordance with your company policy or available disk space by editing the config.ini file.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Increasing your virtual disk size

The Developer Portal is pre-configured with a 10 GB disk , or 20 GB for API Connect version 5.0.2.0 and later, containing a single partition managed by Logical Volume Manager. You can increase the disk size by adding a new virtual disk.

Before you begin

The machine that you are configuring must be powered off.

About this task

The size of disk that you need for the Developer Portal varies depending on the number of sites hosted and the number of concurrent users you expect your site to have. For more information, see Deploying the Developer Portal OVA file.
For this task, you are increasing your disk size to 300 GB.

Important: You must not modify your primary default disk to increase virtual disk size. You can increase disk size by adding a new virtual disk.

Procedure

Add a new 300 GB virtual disk by using your virtualization software.

a. Right-click your virtual appliance name and select Edit settings.
b. Select the Hardware tab and click Add.
c. Choose Hard disk from the list of options, then click Next.
d. Select the Create a new virtual disk radio button, then click Next.
e. Under the Capacity heading, enter `300`and `GB` into the available fields.
f. Under the Disk provisioning heading, select the appropriate option for your organization.
g. Under the Location heading, select Store with the virtual machine, then click Next.
h. Leave all content under the Virtual device node heading as default, then click Next.
i. Review your settings, then click Finish.
j. V5.0.7 + Reboot the Developer Portal server.
The new 300 GB virtual disk has been added to the root partition. You can verify this increase by logging in to your Developer Portal virtual appliance with a Secure Shell connection, and running the following command:

```
df -h /
```

k. V5.0.6 and earlier Log in to your Developer Portal virtual appliance with a Secure Shell connection, by using the following credentials:

- User ID: `admin`
- Password: `!n0r1t5@C`
  Note: 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, please ensure that you have typed the password correctly.

See Logging in to the CLI with a Secure Shell session connection for information on logging in to the virtual appliance with a secure shell connection.

l. V5.0.6 and earlier To find the name of the new virtual disk you created in the preceding steps, enter the following commands:

```
sudo -i
lsblk
```

You will find a partition of size 300 GB which will be named, for example, `sdb`.
m. V5.0.6 and earlier Enter the following commands to extend the root file system onto the new disk.
For a Debian OVA:

```
sudo -i
pvcreate /dev/sdb
vgextend devportal /dev/sdb
lvextend /dev/devportal/root -l +100%FREE
resize2fs /dev/devportal/root
logout
```

Note: If you repeat this procedure to add another virtual disk, replace **/dev/sdb** with **/dev/sdc** in the previous commands.
n. V5.0.6 and earlier Reboot the Developer Portal server.
The new 300 GB virtual disk has been added to the root partition. You can verify this increase by running the following command:

```
df -h /
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Editing the config.ini file

After the Developer Portal is installed, you can configure settings in accordance with your company policy or available disk space by editing the config.ini file.

About this task

You must be an administrator and have access to the CLI for your Developer Portal virtual machine.

Procedure

If you change a setting, uncomment the line by removing the # symbol, for the change to become effective.

Edit the following file:

```
/home/admin/config/config.ini
```

Within this file, you can configure the following options:

Networking configuration

This setting dictates which network card is used to detect the IP addresses, which is eth0 by default. Remove the # from the following line:

```
#NETWORKCARD_FOR_IP=eth0
```

This uncomments the line and allows you to edit which network card is configured, the following line is an example of the line configured for an `eth1` network card:

```
NETWORKCARD_FOR_IP=eth1
```

You can find out what your network cards are by entering the following command:

```
ifconfig
```

When you have edited the **NETWORKCARD_FOR_IP** setting, enter the following command:

```
set_hostname
```

Developer Portal queuing mechanism

You can edit variables that control the queuing mechanism for the Developer Portal. Any action that changes a site (add, update, upgrade, add_language, delete) is executed via a queue and the throughput will be affected by these variables. The following lines are examples of configurable queuing mechanism options
Note: Only one action can be executed per site, per cluster

```
#MAXEXECUTORS=4
```

This value corresponds to the maximum number of tasks that will execute from the queue on each machine. By default, this value is equal to the number of processors in your machine.

```
#MAX_LANG_EXECUTORS=3
```

This value corresponds to the maximum number of `add_language` tasks that will run from the queue on each machine.

```
#QUEUE_TASK_MAX_RETRIES=3
```

This value corresponds to the number of times a failed task will be retried. By default, this value is 3.

```
#TASK_CHECK_EXECUTE_SECS=90
```

This value corresponds to the number of seconds before the systems checks that a task is still running. By default, this value is set to 90 seconds.

```
#QUEUE_LOCK_CHECK_FAIL_MAX=3
```

This value corresponds to number of failed attempts to test the process at the `TASK_CHECK_EXECUTE_SECS` interval before the queue lock is forced to be unlocked. By default, this value is 3.

```
#TASK_CHECK_SSH_TIMEOUT_SECS=5
```

This value corresponds to the number of seconds to wait for an SSH response when checking that a task is still running on a remote node. By default, this value is set to 5.

Timestamp check of cluster file system sync

You can configure time difference, between the current system time and the entry in a node's timestamp file, for which the **timestamps** command returns a non-zero return code, indicating a bad time stamp. To configure this value, edit the following line:

```
#FILESYSTEM_SYNC_MAX_TIME_DIFF=90
```

By default, this value is set to 90 seconds.

Historical log and backup files

You can control how many historical files for logs and backups are kept. The following lines are examples of configurable historical file settings:

```
#GENERATE_LOG_FILES_TO_KEEP=3
```

This variable corresponds to the number of stored output files in /home/admin/logs that are generated by `generate_logs` before the oldest files are removed. By default, this value is set to 3.

V5.0.0 ONLY `#SITE_BACKUPS_DAYS_WORTH_TO_KEEP=7`
V5.0.1+ `#SITE_BACKUPS_DAYS_WORTH_TO_KEEP=14`

This value corresponds to the number of days of site backup files in /var/aegir/backups that are kept before the oldest files are deleted.
V5.0.0 ONLY By default this value is set to 7 days worth of site backup files, and these days do not have to be 7 consecutive days of backing up.

V5.0.1+ By default this value is set to 14 days worth of site backup files, and these days do not have to be 14 consecutive days of backing up.

Multiple backup files generated on the same day will all be counted as one day when you set this value.

```
#NODE_BACKUPS_DAYS_WORTH_TO_KEEP=7
```

This value corresponds to the number of days of node backup files in /home/admin/backups are kept before the oldest files are deleted. The days do not need to be sequential, and multiple backup files generated on one day will all be counted as one day when you set this value. By default this value is set to 7.

```
#UNUSED_PLATFORMS_TO_KEEP=1
```

This value corresponds to the number of unused platforms to be kept when all sites have been migrated to the latest platform. By default this value is set to 1.

```
#DB_BINARY_LOGS_MAX_PERCENTAGE_DISK_SPACE=10
```

This value corresponds to the percentage of the total disk space that the binary logs can consume. When this limit is reached, older logs are removed. By default, this value is set to 10%.
Note: You can disable this feature by uncommenting the line and setting the value to `0`.

SSL Protocols and Ciphers for the web server

You can edit SSL Protocol and Cipher enablement for the web server by following the guiding notes in the config.ini file.
Note: If you edit either the SSL_PROTOCOLS or SSL_CIPHERS settings, you must run the following command on all cluster members:

```
update_nginx_ssl
```

To change the SSL_PROTOCOLS and SSL_CIPHERS settings, you must configure the following settings:

```
SSL_PROTOCOLS='TLSv1.1 TLSv1.2'
```

```
SSL_CIPHERS='EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384  EECDH+ECDSA+SHA256
    EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH EDH+aRSA  !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK
!SRP
    !DSS !RC4'
```

Note: The contents for **`SSL_PROTOCOLS`** and **`SSL_CIPHERS`** settings in the previous examples are default values that can be configured.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing the Developer Portal

In order to use the Developer Portal, you need to complete a one-time installation and setup process.

Before you begin

You must complete the following tasks:

- Accessing the cloud console user interface
- Deploying the Developer Portal OVA file

Warning: Do not change the time zone of the Developer Portal appliance. The Developer Portal appliance must be set to the default time zone of `Etc/UTC`. Do not set the `TZ` environment variable for any users, or run the **`dpkg-reconfigure tzdata`** command to set the global time zone. Setting the time zone to anything other than `Etc/UTC` can lead to unpredictable behavior in the Developer Portal.

Procedure

In order to install and set up the Developer Portal, complete the following steps.

Note: These steps should be completed by a System Operator.

1. Install the Developer Portal virtual appliance:
   a. The Developer Portal OVA is configured to use DHCP to obtain an IP address. If you choose to use DHCP, you might want to specify a MAC address, if your IP addresses are given out based on MAC addresses. This can be done in the OVA settings, and should be done before the machine is powered on for the first time. Alternatively, to use a static IP address, complete the following steps.
      i. Enter the following command to become the root user:

      ```
      sudo -i
      ```

      ii. For Developer Portal versions 5.0.8.10-GA or earlier, on a Ubuntu16.04 base, edit the file /etc/network/interfaces, and configure the following entries:

      ```
      allow-hotplug eth0
      iface eth0 inet static
      address ip_address_of_virtual_appliance
      netmask subnet_mask_in_dotted_decimal_format
      gateway gateway_ip_address
      ```

      Where:
      ip_address_of_virtual_appliance is the IP address that you want to set for the virtual appliance, and *gateway_ip_address* is the IP address of your gateway.
      Note: You can use nano to edit the file.
      For Developer Portal version 5.0.8.10-iFix or later, on a Ubuntu18.04 base, edit the file /etc/netplan/00-installer-config.yaml, and configure the following entries:

      ```
      network:
        ethernets:
          eth0:
            dhcp4: no
            addresses: [ip_address_of_virtual_appliance/subnet_mask_in_cidr_notation]
            gateway4: gateway_ip_address
        version: 2
      ```

      Where:
      ip_address_of_virtual_appliance is the IP address that you want to set for the virtual appliance, and *gateway_ip_address* is the IP address of your gateway. *subnet_mask_in_cidr_notation* is the shorthand number for the subnet mask, for example **`192.168.10.0/24`** is equal to the network **`192.168.10.0`** with a **`255.255.255.0`** subnet mask.
      iii. Configure the following entries depending on whether you are installing your Developer Portal OVA on a Debian or Ubuntu base.

      V5.0.7 and earlier Installing on a Debian base
         Edit the file /etc/resolv.conf and configure the following entries:

         ```
         nameserver nameserver_ip_address_1
         nameserver nameserver_ip_address_2
         domain domain_name
         search domain_name
         ```

         Where:
         nameserver_ip_address_1 is the host name or IP address of the domain name server. Additional domain name servers are added on separate lines in the file.
         domain_name is the domain name that you want the machine to have.
      V5.0.8 + Installing on an Ubuntu 16.04 base
         Edit the file /etc/network/interfaces and configure the following entries:

```
dns-search domain_name
dns-nameservers nameserver_ip_address_1 nameserver_ip_address_2
```

Where:
domain_name is the domain name that you want the machine to have.
nameserver_ip_address_1 is the host name or IP address of the domain name server. Additional domain name servers are added on the same line but with one space in between.

Installing on an Ubuntu 18.04 base for Developer Portal version 5.0.8.10-iFix or later

Edit the file /etc/netplan/00-installer-config.yaml and configure the following entries:

```
network:
  ethernets:
    eth0:
      dhcp4: no
      addresses: [ip_address_of_virtual_appliance/subnet_mask_in_cidr_notation]
      gateway4: gateway_ip_address
      nameservers:
        addresses: [nameserver_ip_address_1, nameserver_ip_address_2]
        search: [domain_name]
  version: 2
```

Where:
domain_name is the domain name that you want the machine to have.
nameserver_ip_address_1 is the host name or IP address of the domain name server. Additional domain name servers are added on the same line but with a comma and a space in between.

iv. Configure the machine to detect the changes you have made.

V5.0.7 and earlier Installing on a Debian base:

```
service networking restart
ifup eth0
```

V5.0.8 + Installing on an Ubuntu base:

```
ifdown eth0 && ifup eth0
```

Installing on an Ubuntu 18.04 base:

```
netplan apply
```

Note: you will still be logging into the CLI, but via the virtual appliance console, which can be accessed without the network card having an address. If your machine has picked up an automatic address from DHCP, SSH should be used to change it to static. If your changes are not being reflected, please try restarting the Developer Portal Machine.
Note: For more information and configuration examples for **netplan**, see https://netplan.io/examples/.

v. Exit the root shell:

```
logout
```

See Logging in to the CLI with a Secure Shell session connection for information on logging in to the virtual appliance using a secure shell connection.

2. Configure the Developer Portal in the Cloud Manager:
   a. Log in to the Cloud Manager as a cloud administrator.
   b. In the navigation section of the Cloud Manager, click the Settings icon ⚙. The Settings pane displays.
   c. In the Developer Portals section of the Settings pane, enable the Developer Portal by selecting the Enable the developer portal toggle. This setting allows the Developer Portal to be selected as the portal for any Catalogs that are created.
   d. Enter the virtual appliance host name or IP address in the Portal Management Address field. For more information, see Load balancing the requests the Cloud Manager sends to the Developer Portal.
   e. Ensure the port number, configured for the virtual appliance, in the Port field is set to `22`.
      The API Manager communicates with the Developer Portal via SSH, which is on port 22 by default.
      Important: There is no support for presenting a client certificate when a call to /v1/portal/ is made.
   f. Copy the contents of the Public Key field and store the content so it is available for use in the next step.
      In the next step you will place this key onto the Developer Portal server to allow the API Manager to create an Developer Portal site for each Catalog that is created.
   g. Click Save in the Developer Portals section.
3. Configure the Developer Portal virtual appliance:
   a. Log in to the Developer Portal virtual appliance CLI with the following credentials:
      - User ID: `admin`
      - Password: `!n0r1t5@C`
        Note: 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, please ensure that you have typed the password correctly.

   See Logging in to the CLI with a Secure Shell session connection for information on logging in to the virtual appliance using a secure shell connection.

   b. Enter the following commands:
      i. Create a file on the file system of the Developer Portal virtual appliance by copying and pasting the public key contents from the location you stored it in during the previous section:

      ```
      echo "contents_pasted_from_the_public_key_field" > key_file_name
      ```

      Where: *key_file_name* is the name of the file that will be created on the Developer Portal virtual appliance.
      Note: If you cannot paste your command into the console, you can **ssh** in with a client such as PuTTY, or use **winscp** to copy a file across with the relevant contents.

      ii. Run the `set_apim_host` command to connect the Developer Portal to the Management cluster:

      ```
      cat key_file_name | set_apim_host devportal_fqdn apic_fqdn
      ```

      Where:
      - *devportal_fqdn* is the fully qualified domain name of the Developer Portal virtual appliance. If you are setting up a cluster, then each Developer Portal in the cluster must have a unique name.
      - *apic_fqdn* must be the same value as the address in the Management service on the APIC Cloud Manager.

      Important: You cannot use the IP address of your virtual machine as the *devportal_fqdn* in this command. For more information, see set_apim_host.

      iii. Configure the SMTP server that will be used to send emails from the Developer Portal.

      ```
      set_smtp mail_server_host_name port user password
      ```

      Where:
      mail_server_host_name must resolve to the IP address of the mail server.
      port is the port to use when connecting to the SMTP server.
      If the SMTP server requires account information then:
      user is the user name to authenticate to the SMTP server.
      password is the password to authenticate the user with.

   If your SMTP server setup does not work after configuration with the **set_smtp** command, run the following command:

   ```
   sudo nano /etc/postfix/main.cf
   ```

   and change any values that your SMTP server requires. Then, save the file and run the follow commands to ensure that postfix is using the new values:

   ```
   sudo service postfix restart
   sudo postfix reload
   ```

4. V5.0.0 ONLY Configure which languages are available in the Developer Portal.
   Edit the file ~/config/languages.txt.
   Each line of the file is a language code that is supported by the Developer Portal, remove any language codes that you do not require. Removing language codes will decrease server load, disk space used and the time taken to create a new Developer Portal for each Catalog.
   Note: English US is always available, all lines in the file are additional languages.
5. Note: You must complete only one of the following three options in this step for your installation to be successful.
   If you have installed a TLS certificate that is signed by a known Certificate Authority on the API Manager then the Developer Portal should already trust the API Manager. If not then you need to configure the Developer Portal to trust the API Manager node. This can be done automatically, manually or you can configure the Developer Portal to trust any TLS certificate; you can use one of the following commands for this depending on which of these ways you wish to configure your Developer Portal:
   - To automatically obtain the TLS certificate, enter the following command:

     ```
     download_apim_cert
     ```

     This causes the Developer Portal to contact the configured Management cluster, download the TLS certificate and add it to the Developer Portal trust store.
   - If you have a file that contains the TLS certificate and want to use this to configure trust, enter the following command:

     ```
     cat filename.crt | set_apim_cert -s
     ```

     Where:
     filename.crt is the name of the file that contains your TLS certificate (in PEM format) .
   - For proof of concept and testing, you can enter the following command:

     ```
     set_apim_cert -i
     ```

     This will cause the Developer Portal to trust any TLS certificate served by the API Connect node.
     Note: This should only be used for development and testing purposes as it is not secure and leaves the Developer Portal exposed to a man-in-the-middle attack.
6. Optional: Configure the Developer Portal to serve your own TLS certificate.

Note: The Developer Portal can support only one TLS certificate, which can be a wildcard or multi-domain certificate. If you want to serve different TLS certificates for each site that is hosted on the Developer Portal, you must front the Developer Portal with a reverse proxy server.
To serve different certificates for each site, the reverse proxy server must be configured to terminate the TLS for the connection to the Developer Portal.

a. If you want the Developer Portal to serve your own TLS certificate, rather than the default, use the following steps. You will need two files with the following content:
   - `myorg.key`, which must contain the TLS private key in PEM format.
   - `myorg.crt`, which must contain the TLS certificate in PEM format.

b. Enter the following commands:

```
cat myorg.key | set_devportal_key
cat myorg.crt | set_devportal_cert
```

Note: If you are using a certificate chain, then the Developer Portal certificate file must be concatenated in the correct order:

```
-----BEGIN CERTIFICATE-----
Insert SSL Certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Insert Intermediate Certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Insert Root Certificate
-----END CERTIFICATE-----
```

Note:
   - V5.0.4 and earlier You will see a validation error after entering the first command, this is because the new key does not match the old certificate. This error should resolve after you have entered the second command.
   - The process to renew SSL certificates is the same as the process to set up new ones.

What to do next

You can configure Catalogs and test the Developer Portal in the Configuring Catalogs and testing the Developer Portal topic.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Backing up and restoring the Developer Portal

You can back up individual Developer Portal sites, or all of the sites and Developer Portal configuration settings. You can then restore backed up sites and portal configurations. You can also take snapshots of your Developer Portal servers.

About this task

You can use the **`backup_site`** command to back up all of the information that is needed to restore a single Developer Portal site. Or use the **`backup_devportal`** command to back up all of your Developer Portal sites and all of the cluster configuration that was set with other commands, such as **`set_apim_host`**. For more information, see backup_site and backup_devportal.

You can also take a snapshot of your Developer Portal server by using the tools that are provided by your virtualization environment. Snapshots are useful when you want to preserve the state of a server, so you can return to the same state if needed. For example, for short-term scenarios such as taking a checkpoint immediately before you apply a fix pack upgrade. You can then delete the snapshot after a few days when the changes are confirmed as successful.
Note:

- Snapshots are not the same as backups. A snapshot is a change log of the original virtual disk, and is not portable. Therefore, do not rely on it as your only backup process. The virtual machine runs on the most current snapshot, not the original vmdk disk files. For optimum recovery readiness, you should take both snapshots and backups of your Developer Portal configuration.
- Site and Developer Portal backup files don't store details of the users, applications, and subscriptions. This information is stored by the Management server. Backups contain Portal configuration and customization data. If a non-PDUR site hasn't been customized in any way, then it can be restored by re-creating the site in the API Manager UI; for more information see Restoring a Portal site by re-creating in the Troubleshooting the Developer Portal guide.

- Backup files are intended for recovery on the same API Connect deployment on which they were taken. Extra steps might be needed if you're attempting to restore a backup file on a different API Connect deployment. For example, updating IP addresses and hostnames, restoring Portal sites to different UUIDs (`restore_site -i`), and so on.

- Backing up individual Developer Portal sites
- Restoring individual Developer Portal sites
- Backing up your whole Developer Portal
- Restoring your whole Developer Portal
- Taking a snapshot of a single Developer Portal server
- Taking a rolling snapshot of a Developer Portal cluster with zero downtime
- Taking a snapshot of a Developer Portal cluster with downtime

Procedure

Backing up individual Developer Portal sites

1. Log in to the Developer Portal CLI.
2. To list all of the sites in your Developer Portal, enter the following command:

   ```
   list_sites
   ```

   Your sites are listed in the console, each ending in the site URL.
3. To back up one of the listed sites, enter the following command:

   ```
   backup_site -u my.url.com
   ```

   where *my.url.com* is the URL of the site you want to back up.
   The console lists the filepath where the backup has been created, as follows:
   `Backup created: backup_file_absolute_path`
   where *backup_file_absolute_path* is the location of the backup file that has been created.
   Important: Copy the backup file to a remote server so that you can restore the Developer Portal site if there is a catastrophic failure.

Restoring individual Developer Portal sites

1. To list all of the backup files for an individual Developer Portal site, enter the following command:

   ```
   ls -ltr /var/aegir/backups/your_site*
   ```

   Note: These backup files are listed with the **newest at the end of the list**.
2. To restore a target site, select an appropriate backup file, then enter the following command:

   ```
   restore_site backup_file_absolute_path
   ```

   where *backup_file_absolute_path* is the location of the backup file you created for the site you want to restore.

To overwrite any existing sites, run the following command:

```
-f
```

To specify a new URL to restore the site to, run the following command:

```
-u
```

To specify a new UUID to restore the site to, run the following command:

```
-i
```

By using `-u` and `-i`, you can create a copy of the site at a new URL and UUID, but the UUID must match the UUID that is used by the API Manager. For more information, see Cloning a site into the same cluster.

Backing up your whole Developer Portal

1. To back up your whole Developer Portal, enter the following command:

   ```
   backup_devportal
   ```

   The console lists the filepath where the backup has been created, as follows:
   `Backup created: backup_file_absolute_path`
   where *backup_file_absolute_path* is the location of the backup file that has been created.
   Important: Copy the backup file to a remote server so that you can restore the whole Developer Portal if there is a catastrophic failure.

Restoring your whole Developer Portal

1. To list all of the backup files for the whole Developer Portal, enter the following command:

   ```
   ls -ltr /home/admin/backups
   ```

   Note: These backup files are listed with the most recent files listed at the end:
2. To restore your whole Developer Portal, enter the following command:

   ```
   restore_devportal backup_file_absolute_path
   ```

   where *backup_file_absolute_path* is the location of the backup file that you created for your portal. If you want to restore the node specific configuration, you must include the option `-n` in the command. See restore_devportal for more options.

Taking a snapshot of a single Developer Portal server

While it is possible to take snapshots with the database running, to ensure data integrity it is recommended that you quiesce the database before you take the snapshot.

1. Stop the database for your Developer Portal server by using the following command:

   ```
   stop_db
   ```

2. Take a snapshot of your server by using the tools that are provided by your virtualization environment.
3. Start the database for your Developer Portal server by using the following command:

   ```
   start_db
   ```

Taking a rolling snapshot of a Developer Portal cluster with zero downtime

To avoid the risk of database reclustering issues on restoration, when you take a snapshot of a Developer Portal cluster, the database of each server should be stopped before the snapshot is taken, and then restarted again after. In this way, you can achieve a rolling snapshot of your cluster with zero downtime. If you must restore all the snapshots at the same time, then the cluster will automatically bootstrap itself with the data from the last server that was snapshot.

1. Stop the database for one of your Developer Portal servers by using the following command:

   ```
   stop_db
   ```

   Note: If your cluster uses a load balancer, you must stop the load balancer from directing any requests to the server before stopping the database on that server.
2. Take a snapshot of your server by using the tools that are provided by your virtualization environment.
3. Start the database for your Developer Portal server by using the following command:

   ```
   start_db
   ```

   If your cluster uses a load balancer, you can resume directing requests to the restarted database server.
4. Repeat Steps 1 to 3 for each server in your cluster.

Taking a snapshot of a Developer Portal cluster with downtime

If the databases on all the cluster members are stopped at the same time, the databases will attempt to auto-recover, unless you stop the cron processes first. In this instance, use the following instructions to take snapshots of your Developer Portal cluster:

1. Stop the cron processes by using the following command on each cluster member server:

   ```
   sudo service cron stop
   ```

2. Stop the databases on all cluster member servers by using the following command on any single server:

   ```
   bootstrap_cluster -t
   ```

3. Take a snapshot of each server by using the tools that are provided by your virtualization environment.
4. Bootstrap the database cluster by using the following command on any single cluster member:

   ```
   bootstrap_cluster -b
   ```

   This command bootstraps the databases by using the most up-to-date stopped cluster member as the master replica.
5. Start the cron processes again by using the following command on each cluster member:

   ```
   sudo service cron start
   ```

6. Wait for all the cluster members to restart. You can check on the current state of the databases on all cluster members that are reachable by running `bootstrap_cluster -s`.

Note: If you need to restore a snapshot of a single Developer Portal server in a cluster, you must run `start_db` on its database so the server rejoins the cluster.

You can manage the number of Developer Portal backups that are kept on the file system by setting the variable `SITE_BACKUPS_DAYS_WORTH_TO_KEEP` in the config.ini file.

Related reference

- bootstrap_cluster
- start_db
- stop_db

Related information

- Restoring a whole cluster

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Configuring your DataPower Gateway and API Connect collective controller to communicate

To be able to use the DataPower® Gateway to invoke applications, you must configure TLS certificates on the collective controller and on the DataPower appliance so that they can communicate.

Before you begin

- You must have set up your DataPower Gateway. For more information, see Using physical DataPower appliances as Gateway Servers or Using virtual DataPower appliances as Gateway Servers.
- You must have administrator privileges for the configuration of your DataPower appliance.
- You must have set up your API Connect collective. For more information, see Installing API Connect collective.
- You must have privileges to edit files in the directory in which you installed the collective controller.

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

About this task

To use the DataPower Gateway to invoke applications, you configure an On-Demand Router in DataPower. An On-Demand Router is similar to a load balancer and redirects calls from your API assembly to the endpoints of your application, which are hosted on a collective and available on several ports. For more information about on demand routing, see On Demand Router in the DataPower documentation.

Important: The tasks in this topic describe the essential steps required to configure DataPower and your API Connect collective controller to communicate. Many of the following tasks can be adapted with additional settings in DataPower, according to your specific needs.

Procedure

To configure your DataPower Gateway and API Connect collective controller to communicate, complete the following tasks:

1. Optional: Creating a TLS key and certificate pair through DataPower
2. Import the generated DataPower certificate into the controller's trust store
3. Download the API Connect collective controller's default certificate
4. Create a new SSL client profile in DataPower
5. Configure an "on demand router" in DataPower

Creating a TLS key and certificate pair through DataPower

About this task

You can create a TLS key and certificate pair by using the key generation tool in DataPower. Alternatively, you can use your own key and certificate pair and upload these later in this topic.

Procedure

To create a TLS key and certificate pair through DataPower, complete the following steps:

1. Log in to your DataPower appliance. By default, port 9090 is used.
2. In the navigation bar on the left, in the Search field, enter `Crypto Tools` and then click Crypto Tools in the search results.
3. In the Common Name (CN) field, provide a common name for the certificate and key pair.
4. In the File Name field, provide a file name for the certificate and key pair.
5. Click on for Export Private Key.
6. Click Generate Key to generate the certificate and key pair.
7. Click Confirm and then Close.
8. In the navigation bar on the left of the DataPower console, in the Search field, enter `File Management` and then click File Management in the search results.
9. Expand temporary and then, by right-clicking it and then selecting Save Link As, download the certificate *name*-sscert.pem, where *name* refers to the name that you specified in step 4.

Import the generated DataPower certificate into the controller's trust store

About this task

The certificate that is used by the connector group in DataPower must be added to the trust store of the collective controller. This certificate can be the one you produced through DataPower or it can be your own certificate that you later upload to DataPower.
Note: This task includes use of the **`keytool`** command. A **`keytool`** utility is provided with the API Connect controller, and is located in the *controller_installation_path*/lib/jre/bin/keytool folder; for example, /usr/lib/node_modules/apiconnect-collective-controller/lib/jre/bin/keytool.

Procedure

To add a certificate to your controller's trust store, complete the following steps:

1. Transfer the certificate that you want to use to the ~/.liberty/wlp/usr/servers/controller/resources/security directory of the appliance on which your collective controller is situated.
2. From the /.liberty/wlp/usr/servers/controller/resources/security hidden directory, import the certificate into the controller's trust.jks keystore by running the command:

   ```
   keytool -import -trustcacerts -file name-sscert.pem -alias datapower -keystore trust.jks
   ```

   where *name-sscert.pem* is the file name of the certificate that you copied into the folder.
3. When prompted to specify whether the certificate is trusted, return `yes`.
4. When prompted for the keystore password, provide the password used for the controller when it was installed.
5. Stop the controller by running the command `wlpn-controller stop` and then restart the controller by running the command `wlpn-controller start`.

Download the API Connect collective controller's default certificate

About this task

If you want DataPower to validate the credentials of your collective, you need to download the controller's default certificate in order to add it to the list of trusted certificates in DataPower.
Note: This task includes use of the **`keytool`** command. A **`keytool`** utility is provided with the API Connect controller, and is located in the *controller_installation_path*/lib/jre/bin/keytool folder; for example, /usr/lib/node_modules/apiconnect-collective-controller/lib/jre/bin/keytool.

Procedure

To download the collective controller's default certificate, complete the following steps:

1. Access and log in to the appliance on which your API Connect collective controller is situated.
2. From the hidden directory
   liberty/wlp/usr/servers/controller/resources/security, run the following command:

```
keytool -export -alias default -keystore key.jks -rfc -file controller.crt
```

The command will generate a file containing the default certificate of the collective controller.

3. Copy the controller.crt file to your local directory for use later.

Create a new SSL client profile in DataPower

About this task

A new SSL client profile is needed for the connector group in DataPower to use.

Procedure

1. If you have logged out or been logged out of your DataPower appliance, log in again.
2. In the navigation bar on the left, in the Search field, enter `SSL Client Profile` and then click SSL Client Profile in the search results.
3. Click Add.
4. In the Name field, provide a name for your client profile.
5. In the Credential section, for Identification Credentials, click +.
   The Configure Crypto Identification Credentials window opens.
6. In the Name field, provide a name for your identification credentials.
7. For Crypto Key, click +.
8. In the Name field, provide a name for your crypto key.
9. In the second, URL, drop-down list for File Name, select your private key from the first task if you created it in DataPower, or click upload and select your private key if you are using another key.
10. Click Apply.
11. For Certificate, click +.
12. In the Name field, provide a name for your certificate.
13. In the second, URL, drop-down list for File Name, select your certificate from the first task if you created it in DataPower, or click upload and select your certificate if you are using another key.
14. Click Apply to save your certificate settings.
15. Click Apply in the Configure Crypto Identification Credentials window to save your identification credentials settings.
16. Optional: If you do not want to validate the server certificate, click off for Validate server certificate.
17. Optional: If you want to validate the server certificate, complete the following steps:
    a. For Validation credentials, click +.
       The Configure Crypto Validation Credentials window opens.
    b. In the Name field, provide a name for credentials.
    c. In the Certificates section, click +.
       The Configure Crypto Certificate window opens.
    d. Click Upload and, in your file system, select the controller certificate that you downloaded from your collective controller the previous task.
    e. In the Configure Crypto Certificate window, click Apply.
    f. In the Configure Crypto Validation Credentials window, click Apply.
18. On the Configure SSL Client Profile page, click Apply.

Configure an "on demand router" in DataPower

About this task

The "on demand router" and connector group in DataPower communicate with the collective controller and route requests to the appropriate applications.

Procedure

To configure your on demand router, complete the following steps:

1. In the navigation bar on the left of the DataPower console, in the Search field, enter `On Demand Router` and then click On Demand Router in the search results.
2. For Administrative State, click enabled.
3. For Connector Groups, click +.
   The Configure ODR Connector Group window opens.
4. In the Name field, provide a name for your connector group.
5. Set SSL client type to Client Profile and then, in the SSL Client Profile field, select the client profile that you created in the previous task.
6. Configure an XML Manager by completing the following steps. You can create a new XML manager or you can use the default XML manager; to use the default XML manager, proceed directly to step 6.c.
    a. For XML Manager, click +. The Configure XML Manager window opens.

   b. In the Name field, provide a name for your XML manager.
   c. Define a User Agent Configuration by completing the following steps. You can define a new User Agent Configuration or you can use the default User Agent Configuration; to use the default User Agent Configuration, proceed directly to step 7.c.iii.
      i. For User Agent Configuration, click +. The Configure User Agent window opens.
      ii. In the Name field, provide a name for your configuration.
      iii. Click the Basic Auth Policy tab.
      iv. Click Add. The Edit Basic-Auth Policy window opens.
      v. In the URL Matching Expression field, enter `*` to act as a wildcard.
      vi. In the Username field, enter the user name that is used to log in to the collective controller.
      vii. For Password alias, click +. The Configure Password Map Alias window opens.
      viii. In the Name field, provide a name for your password alias.
      ix. In the two Password fields, provide the password that is used with the user name provided in step 7.c.vi.
      x. Click Apply in the Configure Password Map Alias window.
      xi. Click Apply in the Edit Basic-Auth Policy window.
      xii. Click Apply in the Configure User Agent window.
   d. Click Apply in the Configure XML Manager window.
7. Add the host and port of your collective controller by completing the following steps:
   a. For ODR Connectors, click Add. The Edit ODR Connectors window opens.
   b. In the Connector Host field, enter the IP address of your collective controller and, in the Connector Port field, enter the HTTPS port that it is using. Unless you have specified otherwise, for example during the set up of your controller, the HTTPS port is **`9443`**.
   c. Click Apply in the Edit ODR Connectors window.
8. From the Configure ODR Connector Group page, click the Custom Properties tab.
9. For Custom Properties, click Add.
   The Edit Custom Properties window opens.
10. In the Name field, enter `profileType` and, in the Value field, enter `Liberty`.
11. In the Edit Custom Properties window, click Apply.
12. Click Apply in the Configure ODR Connector Group window.
13. In the Configure on Demand Router page, click Custom Properties, then click Add.
    The Edit Custom Properties window opens.
14. In the Name field, enter `enableEndpointSelectionBasedOnMatchedVHost` and, in the Value field, enter `true`.
15. In the Edit Custom Properties window, click Apply.
16. From the Configure On Demand Router page, for Connector Groups, ensure that your newly created connector group has been added.
17. Click Apply.
18. In the alert bar at the top of the page, click Review changes.
19. Review the changes that you have made and, when you are satisfied, click Save config.

Results

You have configured your DataPower Gateway and API Connect collective controller to communicate with one another.

What to do next

Modify your assembly so that you can invoke the endpoint of your application as part of an API assembly. For more information, see Modifying the assembly to call an application endpoint hosted on a collective
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing API Connect Professional on-premises with the Micro Gateway

Install and configure the components for the API Connect Professional offering with the Micro Gateway in an on-premises environment.

Before you begin

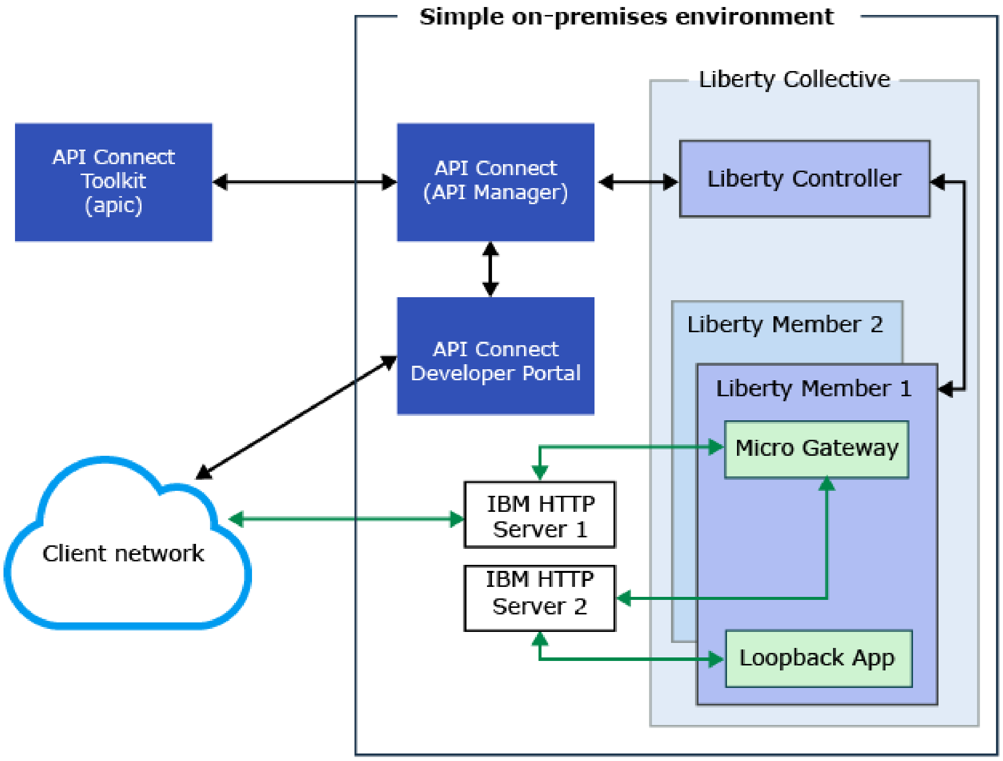
Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
This task assumes that you have completed the following prerequisite tasks:

- Deploy the IBM API Connect Developer Portal OVA file. See Deploying the Developer Portal OVA file
- Install the IBM API Connect Developer Portal. See Installing the Developer Portal.
- Install the Management virtual server. See Installing the Management virtual server.
- Install the API Connect developer toolkit. See Installing the toolkit.

About this task

At a high level, this task describes which components to install for the Professional offering on-premises environment. This configuration uses the following components:

- API Manager to work with your APIs and Products.
- API Connect Toolkit to publish the Micro Gateway, Apps, and products.
- API Connect collective Controller to stop and start, cluster, deploy, and health check Apps.
- API Connect collective Member to host the Micro Gateway and Apps on a member of a cluster.
- API Connect Developer Portal to enable application developers to discover and use APIs.
- IBM HTTP Server as a load balancer.
- Micro Gateway to provide the enforcement component of an API flow.
- LoopBack® application to create APIs.



Procedure

1. Install the API Connect collective Controller and the Collective Member. See Installing API Connect collective.
2. Install IBM HTTP Server by completing the steps in Installing IBM HTTP Server.
3. Install the Micro Gateway by completing the steps in Installing the Micro Gateway.

What to do next

- For details of how you can publish an application to a collective so that it can be called through the Micro Gateway, see Staging and publishing a project from the API Manager.
- For details of how to publish an API to the Micro Gateway, see Creating and publishing an API definition from the command line.

Configuring IBM API Connect® to use a DataPower Tenant

You can configure IBM® API Connect to work with a tenant on a physical Type 8436 DataPower Gateway.

Before you begin

Prepare the Type 8436 DataPower Gateway to support tenants.

- Ensure that the DataPower Gateway is running firmware version 7.6.0.0 or later.
- Ensure that the Application Optimization and Tenant modules are activated on the DataPower Gateway.
- Create a tenant and install its initial firmware. For instructions, see Creating a tenant and installing its firmware.

For additional information about tenants on a DataPower Gateway, see Tenants.

About this task

A Type 8436 DataPower Gateway with the Tenant module can host an independent gateway instance known as a tenant. The following procedure configures IBM API Connect to use a DataPower tenant.

Procedure

1. Define the memory, CPU, and management connections to the DataPower tenant.
   From the **`default`** domain of the landlord, you can modify the connection details for the tenant. Each tenant and the landlord have different management interfaces and a different quota enforcement server.
   - When you define the addresses for tenant management interfaces, you must use an explicit IP address, not a host alias, to isolate tenant management traffic from the landlord management traffic. Tenant services that use the IP address values of **`0.0.0.0`** or **`::`** can cause port contention on the landlord.
   - When you define listening ports for tenant management interfaces, comply to the following guidance.
     - Use a value that is greater than 1024.
     - Ensure that they are unique across the tenants and the landlord.

   The following configuration example is for only illustrative purposes. Define the setting based on your environment and processing needs.
   a. In the CPUs field, enter `16` as the number of CPU threads to allocate to the tenant.
   b. In the Memory field, enter `32` as the memory in GiB to allocate to the tenant.
   c. Set the name, address, and listening port Telnet service.
      - Name: `telnet1`
      - Address: `9.1.2.3`
      - Port: `5002`

   d. Set the address and port for the SSH service.
      - Address: `9.1.2.3`
      - Port: `5003`

   e. Set the address and port for the web management service.
      - Address: `9.1.2.3`
      - Port: `5004`

   f. Set the address and port for the XML management service.
      - Address: `9.1.2.3`
      - Port: `5005`

   g. Set the address and port for the REST management service.
      - Address: `9.1.2.3`
      - Port: `5005`

   h. Set the server port for the quota enforcement service to `16599`.
   i. Set the monitor port for the quota enforcement service to `26599`.
2. Apply and persist the connection details for the tenant.
3. Access the tenant and persist its configuration.
   a. Open a session to the tenant on the configured web management interface.
   b. Log in to the web with default user name of `admin` and password of `admin` and accept the terms of the license agreements.

    The web manage interface restarts.
   c. Log in again to change password for the **admin** account.
      The interface restarts.
   d. Log in again with the new password, and persist the configuration changes.
4. Configure API Connect for the tenant from the Cloud Manager Console.
   a. Select the services tab ☰, then click ⊕ Add and select Add DataPower Service.
      The New DataPower Service window appears.
   b. Configure the new DataPower Service with a display name, either the host name or address of the DataPower device, the Management Server port you wish to use for inbound API calls, and the starting port number (Port Base) of a group of ports on the Management Server that are automatically assigned as needed.
      A new service appears under the DataPower Services section.
   c. Click Add Server under the service that you created.
      The New DataPower Server window appears.
   d. Configure the new gateway server with the address of the device that hosts the tenant, the tenant XML management port, the log in credentials you configured, and the network interface configuration of the gateway device. Ensure that the entry matches your device.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Upgrading your API Connect cloud

You can upgrade your API Connect on-premises cloud to take advantage of the most recent fixes and minor enhancements.

Check IBM® Fix Central to see whether upgrades are available. You should also check the upgrade paths between the supported versions of API Connect; see Supported upgrade paths between product levels of IBM API Connect on the IBM Support Portal.

When you upgrade the API Connect cloud, the new level of the product overwrites the existing level. However, existing user configuration is not overwritten during the upgrade and all organizations, APIs, and resources are preserved.

- **Changes to API Connect after upgrading**
  By upgrading an IBM API Management Version 4.0 or later appliance to IBM API Connect Version 5.0 or later, existing configurations and concepts are changed to align with the new format.
- **Validated upgrade path for API Connect**
  If you are upgrading from IBM API Management Version 4.0, you must upgrade from IBM API Management Version 4.0.4.6 to the latest version of IBM API Connect Version 5.0.
- **Downloading the images to upgrade IBM API Connect servers**
  You can download the firmware images from IBM Fix Central to upgrade your API Connect servers.
- **Testing the upgrade process and path**
  It is recommended that you test the upgrade process on an installation with a configuration backup of the API Management installation that you want to upgrade, because upgrading your installation is a non-reversible process. By testing the upgrade process, you can validate the upgrade path that you are taking without risking your active API Management installation.
- **Upgrading your IBM DataPower Gateways in API Connect**
  When you upgrade IBM API Management Version 4.0 or later to IBM API Connect Version 5.0 or later, you must upgrade your DataPower® Gateway to version 7.5 or later.
- **Upgrading your API Connect solution**
  Apply a product image upgrade for each server in the Management service of your IBM API Connect cloud environment. Then refresh servers in the Gateway services to apply the corresponding maintenance level.
- **Verifying maintenance level**
  After applying an IBM API Connect upgrade, verify that the installation is complete and the correct version is applied to all servers.
- **Upgrade the Developer Portal**
  You maintain the Developer Portal by applying an upgrade to take advantage of the most recent fixes and minor enhancements. If you have already installed the latest fix pack, you can also upgrade the Drupal code and contributed modules as well as the Linux system and associated packages.
- **Troubleshooting the upgrade process to IBM API Connect Version 5.0 or later**
  You can use commands and look for errors to help solve any issues that you face during the upgrade process to IBM API Connect Version 5.0 or later. In addition to solving issues in an upgrade to an existing system, you can apply the same answers to a test upgrade.

Related information

- Cloud Console overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changes to API Connect after upgrading

By upgrading an IBM® API Management Version 4.0 or later appliance to IBM API Connect Version 5.0 or later, existing configurations and concepts are changed to align with the new format.

During the upgrade process, any configurations that you have created in IBM API Management Version 4.0 or later are automatically converted to the new format. The following table shows the specific configuration and conceptual changes that occur during the upgrade process:

Table 1. The configuration and conceptual changes between IBM API Management Version 4.0 and IBM API Connect Version 5.0

IBM API Management Version 4.0	IBM API Connect Version 5.0
Plans are used to collect APIs into a publishable unit. Developer applications can subscribe to Plans.	Products are used to collect APIs into a publishable unit. Products can contain multiple Plans, and Plans can contain multiple APIs. Developer applications subscribe to Plans within Products.
Environments are a staging target for Plans. Each environment has an associated Developer Portal.	Catalogs are the equivalent of Environments. Products are staged to a Catalog, and each Catalog has an associated Developer Portal.
The Basic Developer Portal and the Advanced Developer Portal are the two types of Developer Portal that are available to an API Management appliance.	The Developer Portal is the only type of Developer Portal that is available. The Basic Developer Portal no longer exists.
SSL profiles are used to secure transmission of data between the API Management appliance and other systems.	TLS Profiles are the equivalent of SSL Profiles.
APIs are stored in an IBM proprietary format. An API definition can be extracted as an OpenAPI (Swagger 2.0) file. The generated OpenAPI (Swagger 2.0) cannot represent the operation definitions because the assembly flow cannot be represented in OpenAPI (Swagger 2.0) format.	APIs are stored in OpenAPI (Swagger 2.0) format. API definitions can be extracted and stored offline in your change control system. The OpenAPI (Swagger 2.0) format has been augmented to fully represent the API operations, including the assembly flow.
V5.0.7+ Analytics are used to view logs and information about your APIs and your system. Attention: The structure and settings for your analytics are migrated from Version 4.x to Version 5.0.7.0, or later. The existing analytics data that was collected on Version 4.x is not migrated to Version 5.0.7.0, or later.	V5.0.7+ Attention: The structure and settings for your analytics are migrated from Version 4.x to Version 5.0.7.0, or later. The existing analytics data that was collected on Version 4.x is not migrated to Version 5.0.7.0, or later.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Validated upgrade path for API Connect

If you are upgrading from IBM® API Management Version 4.0, you must upgrade from IBM API Management Version 4.0.4.6 to the latest version of IBM API Connect Version 5.0.

Although unsupported, it is possible to follow an upgrade path that has not been validated, but you are recommended to test any upgrade path, validated or non-validated, on a non-critical system before applying it to your active system to ensure that specific configurations are upgraded correctly. For more information, see Testing the upgrade process and path.

In addition to the specific validated upgrade paths for the API Management appliance, you must upgrade your IBM DataPower® Gateway appliance. For more information, see Upgrading DataPower for Gateway servers.

Downloading the images to upgrade IBM API Connect servers

You can download the firmware images from IBM® Fix Central to upgrade your API Connect servers.

Before you begin

- Check IBM Fix Central to see whether fix packs are available.

About this task

For the Management server, the upgrades are in the WebSphere and IBM API Connect product groups.

For the DataPower® appliance, the upgrades are in the WebSphere and IBM DataPower Gateways product groups.

Downloading the IBM API Connect image

Procedure

To download the API Connect image, complete the following steps:

1. Go to IBM Fix Central.
2. In the Product Group field, select WebSphere.
3. In the Select from WebSphere field, select IBM API Connect.
4. In the Installed Version field, select the version of the product that you want to install, click Continue.
5. Select Browse for fixes and click Continue.
6. Locate the file that matches the version that you want to upgrade.
   For example, APIConnect_Management_*version*_*timestamp*_*unique_id*.vcrypt2.
   Note: In addition to the vcrypt2 file, there might also be an equivalent ova file. Ensure that you select the vcrypt2 file.
7. Select the file that you want to download, click Continue.
8. Log in and agree to the Terms and Conditions. A download link is displayed for each file.
9. Save the file locally.

Results

The upgrade image is stored locally.

Downloading the DataPower firmware image for Gateway servers

Procedure

To download the DataPower firmware image for Gateway servers, complete the following steps:

1. Go to IBM Fix Central.
2. In the Product Group field, select WebSphere.
3. In the Select from WebSphere field, select IBM DataPower Gateways.
4. In the Installed Version field, select the version of the product that is installed.
5. Click Continue.
6. Select Browse for fixes and click Continue.
7. Locate the fix pack that matches the version to which you want to upgrade.
   For example, XG45-7198-6.0.1.1-Firmware.
   For more information about how to determine the firmware image for DataPower, see Upgrading your IBM DataPower Gateways in API Connect.
8. Click Continue.
9. Log in and agree to the Terms and Conditions. A download link is displayed for the file.
10. Save the file locally.

Results

The upgrade image is stored locally.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Testing the upgrade process and path

It is recommended that you test the upgrade process on an installation with a configuration backup of the API Management installation that you want to upgrade, because upgrading your installation is a non-reversible process. By testing the upgrade process, you can validate the upgrade path that you are taking without risking your active API Management installation.

Before you begin

You must take a configuration backup from your existing API Management system to test the upgrade on. For more information, see Creating a backup of an API Connect configuration.
Ensure that any DataPower® appliances that you are using during the testing process are running with firmware at the minimum required level. For more information, see Upgrading your DataPower appliances.

About this task

You can test an upgrade process and path for an IBM® API Connect Version 5.0 installation to later versions.

Procedure

1. V5.0.6 + Verify that your Informix database for your existing system is in a healthy state by entering the following command using the Command Line Interface: `stat show apiconfig`.
   The information that is returned provides statistics about the health of the database. This includes the following information:

   Internal check summary
   : Lists PASS/FAIL information from running Informix verification utilities. All checks must report a status of `Pass`.

   Memory segments
   : Shows the amount of RAM, both as a percentage and as an aggregate amount, that the Informix engine is using on the current server. The percentage is compared to its maximum allowed memory that is provided to the virtual machine. Review this to make sure that the Informix engine has enough available RAM.

   Database spaces
   : Shows how much disk space the Informix engine is using on the current server. Review this to ensure the database engine has enough available disk space. Note that only Primary servers use `tempsblobsp`, so its output is suppressed on other servers.

   Online server roles
   : Lists online servers, roles, and database transaction log replication state. Database transaction log replication state is represented by a combination of the LogID and LogPos values. LogID is an indication of which transaction log is currently being used; LogPos is the most recently used position within that log. If you watch these values over time LogPos grows to a preconfigured limit, then LogID is incremented by one and LogPos is reset to zero and begins growing again. Database content changes appear first on the primary servers, and then are copied to HDR and RSS servers. In a healthy configuration, the values for LogID/LogPos on the primary server are quickly reflected on the other servers.

   If the checks fail, a line is added to the end of the output that warns you about the situation. The text is similar to the following message:

   ```
   WARNING: Errors detected, see 'stat show apiconfig verbose'.
   ```

   See Testing Management servers for more information about the results of this step.

2. Install a new API Management appliance and ensure it is at the same version as the appliance that you want to upgrade.
   Note: The new management server must be configured so that it cannot access the original management servers, DataPower Gateway servers, or Developer Portal servers.
3. Restore your configuration backup to your new appliance.
   After the back up is restored, any DataPower appliances that are in your main system appear in the restored system.
   V5.0.8 + From API Connect Version 5.0.8.3 onwards, you can load a previous version of your API Connect configuration in isolation mode. By using the `isolate` option on the **config load apiconfig** command, the management configuration file is loaded in isolation, in other words without any references to DataPower Gateway servers, Developer Portal servers, or any third-party systems for

analytics offload. This mode is the preferred way of testing the upgrade process. However, there is also a `restore` option on the command if you do not want to load your configuration in isolation mode.

V5.0.8 + Note: When you restore your configuration in isolation mode, or for testing an upgrade if the new management server cannot reach the original portal server(s), and you want the Developer Portal in the restored configuration to be able to work with the existing catalogs, you must perform the following steps in the newly restored API Manager UI:

a. For each Catalog, select Settings > Portal, and change the Developer Portal setting to None. Save your changes.
b. When you want to enable a Developer Portal, for each Catalog you must complete the Settings > Portal section with the new host name for the Developer Portal URL. Save your changes.

V5.0.7 and earlier Load a previous version of your API Connect configuration in restore mode by using the **config load apiconfig** command.

For more information about both the isolate and restore modes, see Restoring an API Connect configuration.

4. Optional: Remove any of your main system's DataPower appliances from the restored system only if your new API Management appliance cannot reach them.
Important: If your restored system has network connectivity to a DataPower appliance and you removed it, any API configurations on the DataPower appliance are also removed. The DataPower appliance is also removed from your main API Management or API Connect installation, and might cause an outage to production API traffic.
It is recommended that you only remove the DataPower appliances from the restored system if the restored system has no network connectivity to the DataPower appliances.

   If you do not remove the DataPower appliances from the restored system, configuration changes are not made to the appliances. After you upgrade the API Management appliance, new configurations are uploaded to DataPower appliances only when they are removed and added in the Cloud Manager UI, when changes are made to the Gateway cluster in the Cloud Manager UI, or when new Products are published to Catalogs.

5. Configure a DataPower appliance to function with the restored system.
6. Run the upgrade command on the restored system, through the CLI. For more information, see Upgrading your IBM API Connect.
7. After the system has upgraded, log in to the Cloud Manager to verify that the configuration for it is correct.
8. Log in to the API Manager to verify that your APIs, Products, and Catalogs are successfully converted to the upgraded format.
   - For APIs, check for any OpenAPI (Swagger 2.0) validation and assembly definition errors. APIs can be invoked for validation. For more information on invoking APIs for validation, see **Testing an API with the API Manager test tool**
   - For Products, ensure that your Products are published to the correct Catalogs, and that any application developer subscriptions to Products are also correct.
   - For Catalogs, ensure that any users, roles, and Developer Portal settings are also correct.

   Note: The URL path for API Manager has changed to /apim.
9. Install and configure a Developer Portal site to ensure that your APIs and Products are displayed and function correctly in the Developer Portal. For more information on the Developer Portal, see Discovering and using APIs through the Developer Portal.

Results

You have completed a test of the upgrade process, and determined the validity of your upgrade path.

What to do next

- If your test was successful and your upgrade path is valid, you can progress to upgrading your existing API Management appliance. For more information, see Upgrading your API Connect solution
- If your test was unsuccessful and your upgrade path is invalid, you must resolve the any issues that caused the failure. For more information, see Troubleshooting the upgrade process to IBM API Connect Version 5.0 or later.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Upgrading your IBM DataPower Gateways in API Connect

When you upgrade IBM® API Management Version 4.0 or later to IBM API Connect Version 5.0 or later, you must upgrade your DataPower® Gateway to version 7.5 or later.

Before you begin

- Verify that the maintenance level you apply to the DataPower Gateway is compatible with all the IBM API Connect environments and other applications running on your gateway.
- Verify that all Management server upgrades are complete, see Verifying maintenance level.
- Schedule the upgrade during periods of low activity because overall capacity is reduced during maintenance.
- If the Cloud Manager contains only one DataPower Gateway server, ensure that the API Connect environment does not require access while you upgrade.

About this task

- The DataPower Gateway is maintained separately from other API Connect components.
- Upgrading API Connect does not always require you to upgrade the DataPower Gateway.
- When you remove a DataPower Gateway server, historical monitoring information relating to CPU, memory and disk usage, and server load is lost. However, analytics data relating to API usage is retained.
- Removing the DataPower Gateway from the Cloud Manager removes it from the load balancing group. When you re-add the Gateway, it obtains the latest DataPower configuration.
- You can perform a rolling upgrade for multiple DataPower Gateway servers in a load balancing group by removing one, then updating and re-adding it before continuing to the next.

Procedure

Following are general steps to upgrade a DataPower Gateway to meet the requirements of API Connect:

1. Turn off upstream traffic.
2. Wait for the upstream traffic to stop.
3. In the Cloud Manager, remove the DataPower Gateway. For more information, see ../com.ibm.apic.cmc.doc/node_remove.html.
4. Depending upon your platform, perform the product upgrade, or decommission your virtual machine image, or remove your container from service.
   The following examples link to IBM documentation for DataPower Version 7.5.0, but you can choose the target version of your desired upgrade with the selector found on the page.
   a. To upgrade a DataPower hardware appliance, see IBM Data Power Appliance Firmware Installation Version 7.5.0
   b. To upgrade all other DataPower products, select your platform in the Virtual section of Data Power Offerings for Version 7.5.0
5. Deploy the new DataPower Gateway.
6. Test your new deployment.
7. Turn on upstream traffic.
8. Add the Gateway server to a service.
   a. Add the Gateway server by using the Cloud Manager. For more information, see Adding a Gateway server.
   b. If you have two or more Gateway servers and you are using an external load balancer, add the Gateway server to the load balancer again.

Results

The upgrade is applied to your DataPower Gateway.

What to do next

1. Verify the upgrade from the command line by using the **--version** command.
   - The following message is an example of the **--version** command output from a DataPower XI52 appliance:

```
xi52# show version

         Serial: 0000000
        Version: XI52.7.1.0.3
          Build: 256232
     Build Date: 2015/01/30 10:39:05
 Watchdog Build: XI52.7.1.0.3
 Installed DPOS: XI52.7.1.0.3
   Running DPOS: XI52.7.1.0.3
XML accelerator: embedded
   Machine Type: 5725
     Model Type: J91
```

   - The following message is an example of the **--version** command output from a DataPower XG45 virtual appliance:

```
xg45# show version

         Serial: 0000000
        Version: XG45.7.1.0.14
```

```
              Build: 283791
         Build Date: 2017/01/12 13:45:10
     Watchdog Build: XG45.7.1.0.14
     Installed DPOS: XG45.7.1.0.14
       Running DPOS: XG45.7.1.0.14
    XML accelerator: embedded
       Machine Type: 7198
         Model Type: 32X
```

2. With the Version information provided, verify the following data:
   - Verify that the version value matches the value for the installed image.
   - Verify that the version value and the DPOS values that are running are identical.

Related tasks

- Upgrading your API Connect solution

Related information

- Cloud Console overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Upgrading your API Connect solution

Apply a product image upgrade for each server in the Management service of your IBM® API Connect cloud environment. Then refresh servers in the Gateway services to apply the corresponding maintenance level.

Before you begin

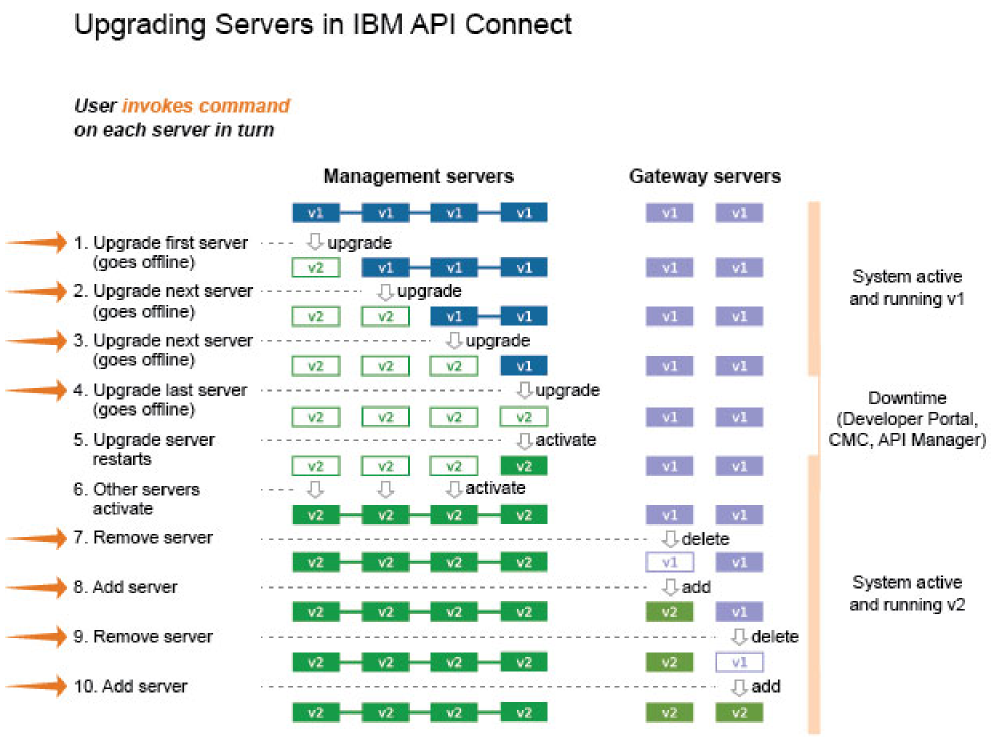
Note: For any service, the servers must be at the same version level. If you are upgrading from v50810-ifix2, see Upgrading your API Connect solution from v50810-ifix2 to v50811.
Downgrading a Management server to an earlier fix pack level is not supported.

Before upgrading, complete the following steps:

- Back up your API Connect cloud and data, see Preserve your cloud data.
- If the Cloud Manager on-premises cloud contains only one Management server, ensure that the API Connect cloud is not being used while you upgrade the server because the Cloud Manager is inactive during parts of the upgrade process. If you have more than one Management server, you can continue to use the API Connect cloud while you upgrade each server.
- You must have 16GB of RAM to ensure that an upgrade to IBM API Connect Version 5.0 or later, is possible.

About this task

To successfully upgrade the API Connect solution, you must perform operations on both the Management service(s) and Gateway service(s). The following diagram illustrates how the sequence of events unfold as you apply maintenance to the various servers in the API Connect solution.


Upgrading Servers in IBM API Connect
User invokes command
on each server in turn
Management servers
Gateway servers
1. Upgrade first server (goes offline)
2. Upgrade next server (goes offline)
3. Upgrade next server (goes offline)
4. Upgrade last server (goes offline)
5. Upgrade server restarts
6. Other servers activate
7. Remove server
8. Add server
9. Remove server
10. Add server
upgrade
activate
delete
add
v1
v2
System active and running v1
Downtime (Developer Portal, CMC, API Manager)
System active and running v2

In the Upgrading Servers in the IBM API Connect diagram, there are four Management servers in the Management service and two Gateway servers in the Gateway service.
Steps 1-8 illustrate the upgrade sequence for the Management service. Upgrade the Management servers one at a time. As each Management server is upgraded, the server is temporarily removed from the cloud and remains inactive until the last server is upgraded. You can begin with any non-primary Management server in the Management service, and upgrade the Primary Management server last. Verify that the upgrade is successful before initiating the upgrade for the next Management server. After the successful upgrade of the last Management server, which should be the Primary Management server, the Primary Management server automatically activates and reactivates all the other upgraded servers in the Management service. The role of the server (for example, Primary, RSS) can be viewed under server details for each server in the Services tab of the Cloud Manager.

The steps that follow illustrate the sequence in which the Gateway service should be refreshed after upgrading the Management services. You might also need to upgrade your DataPower® product. For more information about upgrading DataPower, see Upgrading your IBM DataPower Gateways in API Connect.

To refresh the IBM API Connect Gateway service, delete and then add each Gateway server in the service, one at a time.

Lastly, verify that all the servers in the Management and Gateway services are running the same version, see Verifying maintenance level.
Note: When comparing versions, verify that the entire build number matches. For example:

```
5.0.0.0.20160321-1354_H9_64
5.0.0.0.20160321-1354_H9_64
```

not

```
5.0.0.0.20160321-1354_H9_64
5.0.0.0.20160209-1615_H9_64
```

When you apply maintenance, the new level of the product overwrites the existing level. Your user configuration, APIs, and Products are retained.
Important: You must have 16GB of RAM to ensure that the upgrade to IBM API Connect Version 5.0 or later, is possible.

Procedure

This procedure provides the steps to upgrade an individual Management server. Repeat this procedure for each Management server in your Management service.

1. Required: (5.0.8.7 iFix 4 or later) Add new certificates to your DataPower Gateway servers.
   You must complete this step once before upgrading to API Connect 5.0.8.7 iFix 4 (or a later), to prevent the loss of analytics events data during the upgrade. For instructions, see Add certificates to gateways before upgrading API Connect.

   If you skip this task, the upgrade will be successful but you will lose analytics event records spanning the time when the management servers start up at the upgraded level until each Gateway server is removed and re-added after the upgrade.

   Attention: This is a one-time task and does not need to be repeated with subsequent upgrades.

2. Download the appropriate firmware image from IBM Fix Central. See IBM Fix Central and Downloading the images to upgrade IBM API Connect servers.
   The Management server is a virtual appliance. The firmware image file name includes ManagementAppliance and the file extension is .vcrypt2.
   For example,

   ```
   version-APIConnect-ManagementAppliance-timestamp_H9_64-CUMUIFIX-003.vcrypt2
   ```

   Host the downloaded firmware on an FTP server that is accessible from the Management server that you want to upgrade. "Manually manipulating the load balancer is not necessary because upgraded servers will be down until upgrade process is complete"
3. In your virtual machine hosting software, for all of your Management servers that you are upgrading, increase the disk size to 10GB. A newly deployed IBM API Connect Version 5.0 appliance has a disk size of 10GB. Upgrading an IBM API Management Version 4.0 appliance requires a disk size increase. The only disk that must be increased is the code disk. The new disk size comes into effect after the upgrade to IBM API Connect Version 5.0 is complete.
   V5.0.1+ For more information, see Increasing code disk size for appliances

4. Log on to the CLI on the Management server through a Secure Shell (SSH).
5. Using the firmware image that is hosted on an FTP server, update the Management server by entering the following **system update firmware** command:
   Note: FTP is shown here as an example. If you prefer, you can select another protocol (SFTP, HTTP or HTTPS) that is more convenient for your environment.

   ```
   Syntax :

           system update firmware from ftp <hostname|ip> user <username> file <filename>
   ```

   Where:
   - *hostname|ip* is the FTP server where the firmware image is stored
   - *username* is the user name that is used to log in to the FTP server
   - *filename* is the absolute path to the firmware image .vcrypt2 file that is stored on the FTP server

   When prompted, enter the password to connect to the FTP server.
   The firmware upgrade is applied to the Management server. After the successful completion of the firmware installation, the Management server remains inactive until the Primary Management server in the service is upgraded. To monitor the upgrade status, use `system show status`.
6. After the upgrade is complete, verify that the correct version is installed by logging in to the CLI and issuing the `system show version` command.
   Note: The status of the Management server that you are upgrading shows as Inactive in the Cloud Manager until all of the other servers in the cloud are upgraded to the same version.
7. Repeat the previous steps until every Management server has been successfully upgraded and you have verified by using the CLI that all the Management servers in the Management service are running the same exact version.
8. After all of the Management servers are upgraded, verify that all of the servers resume normal operations by launching the Cloud Manager and completing the following steps:
   Note: It might take several minutes for all of the servers to come back online.
   a. Log in to the Cloud Manager of the upgraded Management server and select the Services page.
   b. Refresh the browser to ensure that the latest information about the services and servers is displayed.
   c. Click the Server Details icon (i) of each server to view the server details and verify that the upgraded version is installed.
   d. Verify that each Management server is in Active status.
      Note: After a management server upgrade, the Gateway servers are displayed as out of sync on the Services tab in the Cloud Manager. For more information, see Gateway resynchronization.

After maintenance to the Management server, or the Gateway server, each Gateway server in a service must be refreshed.

Removing the Gateway server from the Cloud Manager removes it from the load balancer group (or groups) and also ensures that when you re-add the Gateway server, it obtains the latest DataPower configuration.

Do not delete the Gateway services, just refresh (remove and add) the Gateway servers within them.

9. Remove the Gateway server.
   When you remove a DataPower Gateway server, historical monitoring information relating to CPU, memory and disk usage, and server load is lost. However, analytics data relating to API usage is retained.
   a. If you have two or more Gateway servers and you are using an external load balancer, remove the Gateway server from the load balancer and allow time for any in-flight API transactions to complete.
   b. Remove the Gateway server by using the Cloud Manager, see Removing servers.
10. Add the Gateway server to a service.
    a. Add the Gateway server by using the Cloud Manager. For more information, see Adding a Gateway server.
    b. If you have two or more Gateway servers and you are using an external load balancer, add the Gateway server to the load balancer again.

11. Verify that the Configuration version displayed for each Gateway server matches the version of the Management servers, see Verifying maintenance level.
12. Upgrade your Developer Portal appliance by following the procedure in Applying an IBM fix pack and upgrading all sites to use the new distribution.

Results

The upgrade is applied to the Management service of your cloud environment.

- **Upgrading your API Connect solution from v50810-ifix2 to v50811**
  Apply a product image upgrade for each server in the Management service of your IBM API Connect cloud environment. Then, refresh servers in the Gateway services to apply the corresponding maintenance level.
- **Add certificates to gateways before upgrading API Connect**
  Add new certificates to your DataPower Gateway servers before upgrading IBM API Connect.

Related tasks

- Upgrading your IBM DataPower Gateways in API Connect

Related information

- Cloud Console overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Upgrading your API Connect solution from v50810-ifix2 to v50811

Apply a product image upgrade for each server in the Management service of your IBM® API Connect cloud environment. Then, refresh servers in the Gateway services to apply the corresponding maintenance level.

Before you begin

Note: For any service, the servers must be at the same version level.
To upgrade to v50811, the management servers must be upgraded to v50810-ifix2 first. For more information, see Upgrading your API Connect solution.

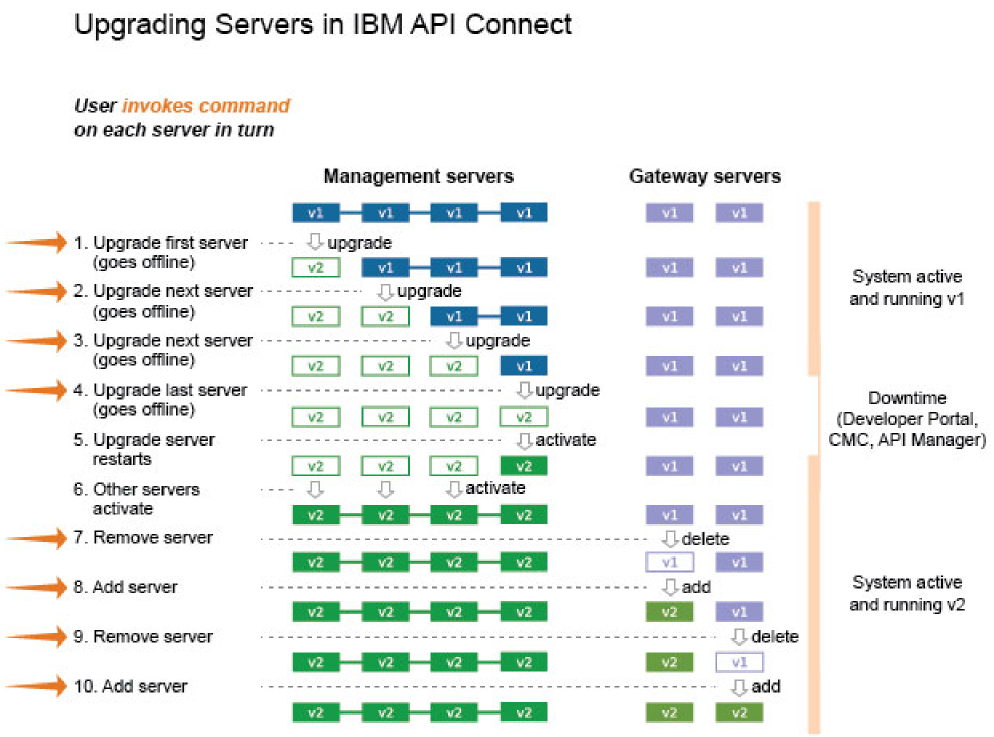
Downgrading a Management server to an earlier fix pack level is not supported.

Before you upgrade, complete the following steps:

- Back up your API Connect cloud and data, see Preserve your cloud data.
- If the Cloud Manager on-premises cloud contains only one Management server, ensure that the API Connect cloud is not being used while you upgrade the server because the Cloud Manager is inactive during parts of the upgrade process. If you have more than one Management server, you can continue to use the API Connect cloud while you upgrade each server.
- You must have 16 GB of RAM to ensure that an upgrade to IBM API Connect Version 5.0 or later, is possible.
- Login to vcenter and take a snapshot of the virtual machine.
- SSH to the vm and run the cli command `config autodiskadd disable`.
- Add a 10 GB free hard disk to the VM.
  Note: There must be only 1 free hard disk of exactly 10 GB for the upgrade to proceed.

About this task

To successfully upgrade the API Connect solution, you must complete operations on both the Management services and Gateway services. For more information, see Upgrading your API Connect solution.


Upgrading Servers in IBM API Connect
User invokes command
on each server in turn
Management servers
Gateway servers
1. Upgrade first server (goes offline)
upgrade
2. Upgrade next server (goes offline)
upgrade
3. Upgrade next server (goes offline)
upgrade
4. Upgrade last server (goes offline)
upgrade
5. Upgrade server restarts
activate
6. Other servers activate
activate
7. Remove server
delete
8. Add server
add
9. Remove server
delete
10. Add server
add
v1
v2
System active and running v1
Downtime (Developer Portal, CMC, API Manager)
System active and running v2

In the Upgrading Servers in the IBM API Connect diagram, there are four Management servers in the Management service and two Gateway servers in the Gateway service.
Steps 1-8 illustrate the upgrade sequence for the Management service. Upgrade the Management servers one at a time. As each Management server is upgraded, the server is temporarily removed from the cloud and remains inactive until the last server is upgraded. You can begin with any non-primary Management server in the Management service, and upgrade the primary Management server last. Verify that the upgrade is successful before you start the upgrade for the next Management server. After the successful upgrade of the last Management server, which should be the primary Management server, the primary Management server automatically activates and reactivates all the other upgraded servers in the Management service. The role of the server, for example, primary or RSS, can be viewed under server details for each server in the Services tab of the Cloud Manager.

The steps that follow illustrate the sequence in which the Gateway service should be refreshed after you upgrade the Management services. You might also need to upgrade your DataPower® product. For more information about upgrading DataPower, see Upgrading your IBM DataPower Gateways in API Connect.

To refresh the IBM API Connect Gateway service, delete and then add each Gateway server in the service, one at a time.

Lastly, verify that all the servers in the Management and Gateway services are running the same version, see Verifying maintenance level.
Note: When you compare versions, verify that the entire build number matches. For example:

```
5.0.0.0.20160321-1354_H9_64
5.0.0.0.20160321-1354_H9_64
```

not

```
5.0.0.0.20160321-1354_H9_64
5.0.0.0.20160209-1615_H9_64
```

When you apply maintenance, the new level of the product overwrites the existing level. Your user configuration, APIs, and Products are retained.

Procedure

This procedure provides the steps to upgrade an individual Management server. Repeat this procedure for each Management server in your Management service.

1. Add new certificates to your DataPower Gateway servers.
   You must complete this step once before you upgrade to API Connect 5.0.8.7 iFix 4 or later, to prevent the loss of analytics events data during the upgrade. For instructions, see Add certificates to gateways before upgrading API Connect.

   If you skip this task, the upgrade will be successful but you lose analytics event records that span the time when the management servers start at the upgraded level until each Gateway server is removed and readded after the upgrade.

Attention: This is a one-time task and does not need to be repeated with subsequent upgrades.

2. Download the appropriate firmware image from IBM Fix Central. See IBM Fix Central and Downloading the images to upgrade IBM API Connect servers.
   The Management server is a virtual appliance. The firmware image file name includes ManagementAppliance and the file extension is .vcrypt2.
   For example,

   ```
   version-APIConnect-ManagementAppliance-timestamp_H9_64-CUMUIFIX-003.vcrypt2
   ```

   Host the downloaded firmware on an FTP server that is accessible from the Management server that you want to upgrade. "Manually manipulating the load balancer is not necessary because upgraded servers are down until the upgrade process is complete"
3. In your virtual machine hosting software, for all of your Management servers that you are upgrading, increase the disk size to 10 GB. A newly deployed IBM API Connect Version 5.0 appliance has a disk size of 10 GB. Upgrading an IBM API Management Version 4.0 appliance requires a disk size increase. The only disk that must be increased is the code disk. The new disk size comes into effect after the upgrade to IBM API Connect Version 5.0 is complete.
   V5.0.1+ For more information, see Increasing code disk size for appliances
4. Log on to the CLI on the Management server through a Secure Shell (SSH).
5. Using the firmware image that is hosted on an FTP server, update the Management server by entering the following **system update osfirmware** command:
   Note: FTP is shown here as an example. If you prefer, you can select another protocol (SFTP, HTTP, or HTTPS) that is more convenient for your environment.

   ```
   Syntax :
   
           system update osfirmware from ftp <hostname|ip> user <username> file <filename>
   ```

   Where:
   - *hostname|ip* is the FTP server where the firmware image is stored
   - *username* is the user name that is used to log in to the FTP server
   - *filename* is the absolute path to the firmware image .vcrypt2 file that is stored on the FTP server

   When prompted, enter the password to connect to the FTP server.
   The firmware upgrade is applied to the Management server. After the successful completion of the firmware installation, the Management server remains inactive until the primary Management server in the service is upgraded. To monitor the upgrade status, use `system show status`.
6. After the upgrade is complete, verify that the correct version is installed by logging in to the CLI and issuing the `system show version` command. Check that is shows the version 50811, and does not report any problems that are related to the upgrade.
   Note: The status of the Management server that you are upgrading shows as Inactive in the Cloud Manager until all of the other servers in the cloud are upgraded to the same version.
7. Repeat the previous steps until every Management server has been successfully upgraded and you verify that all the Management servers in the Management service are running the same exact version, by using the CLI.
8. After all of the Management servers are upgraded, verify that all of the servers resume normal operations by starting the Cloud Manager and completing the following steps:
   Note: It might take several minutes for all of the servers to come back online.
   a. Log in to the Cloud Manager of the upgraded Management server and select the Services page.
   b. Refresh the browser to ensure that the latest information about the services and servers is displayed.
   c. Click the Server Details icon (i) of each server to view the server details and verify that the upgraded version is installed.
   d. Verify that each Management server is in Active status.
      Note: After a management server upgrade, the Gateway servers are displayed as out of sync on the Services tab in the Cloud Manager. For more information, see Gateway resynchronization.

After maintenance to the Management server, or the Gateway server, each Gateway server in a service must be refreshed.

Removing the Gateway server from the Cloud Manager removes it from the load balancer group, or groups, and also ensures that when you re-add the Gateway server, it obtains the latest DataPower configuration.

Do not delete the Gateway services, instead, refresh (remove and add) the Gateway servers within them.

9. Remove the Gateway server.
   When you remove a DataPower Gateway server, historical monitoring information that relates to CPU, memory and disk usage, and server load is lost. However, analytics data that relates to API usage is retained.
   a. If you have two or more Gateway servers and you are using an external load balancer, remove the Gateway server from the load balancer and allow time for any in-flight API transactions to complete.
   b. Remove the Gateway server by using the Cloud Manager, see Removing servers.
10. Add the Gateway server to a service.
    a. Add the Gateway server by using the Cloud Manager. For more information, see Adding a Gateway server.
    b. If you have two or more Gateway servers and you are using an external load balancer, add the Gateway server to the load balancer again.

11. Verify that the Configuration version displayed for each Gateway server matches the version of the Management servers, see Verifying maintenance level.
12. Upgrade your Developer Portal appliance by following the procedure in Applying an IBM fix pack and upgrading all sites to use the new distribution.

Results

The upgrade from v50810-ifix2 to v50811 is successfully applied to the Management service of your cloud environment.

Related tasks

- Upgrading your IBM DataPower Gateways in API Connect

Related information

- Cloud Console overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Add certificates to gateways before upgrading API Connect

Add new certificates to your DataPower® Gateway servers before upgrading IBM® API Connect.

About this task

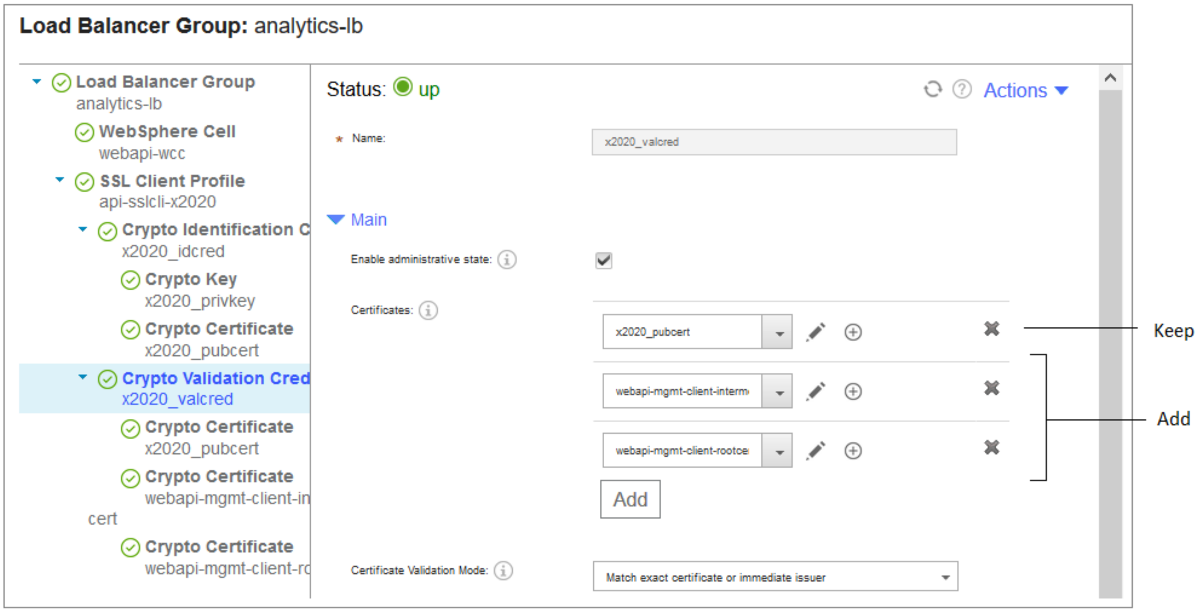
Before upgrading your API Connect deployment to 5.0.8.7 iFix 4 or later, you must add two new certificates to all of your DataPower Gateway servers to ensure that analytics events data is not lost during the upgrade. If you skip this step, the upgrade will be successful but you will lose analytics event records spanning the time when the management servers start up at the upgraded level until each Gateway server is removed and re-added after the upgrade.

Attention: This is a one-time task and does not need to be repeated with subsequent upgrades.

Procedure

1. In API Connect, use the Cloud Manager to locate your DataPower domain.
   a. Log in to the Cloud Manager console and click Services.
   b. In the DataPower Services pane, click ⚙ to open the Service Settings page, and note the DataPower domain.
2. For each Gateway server, select a method (UI or CLI) for adding the certificates to the server.
   Remember: Add the new certificates to every Gateway server in the domain.
   UI: Use the DataPower console to complete the following steps:

   a. Log in to the DataPower Gateway console.
      For the Domain field, select the domain that you obtained in Step 1. For the Graphical Interface field, select Blueprint Console.
   b. In the main navigation list, click to open the Network page.
   c. In the Network navigation list, expand Other and click Load Balancer Group.
   d. In the list of load balancer groups, click analytics-lb.
   e. In the Load Balancer Group analytics-lb tree, expand SSL Client Profile api-sslcli-x2020 and select Crypto Validation Credentials 2020_valcred.
   f. In the "Certificates" section, retain the x2020_pubcert certificate and add the following new certificates:
      - webapi-mgmt-client-intermediatecert
      - webapi-mgmt-client-rootcert



Load Balancer Group: analytics-lb

- Load Balancer Group analytics-lb
  - WebSphere Cell webapi-wcc
  - SSL Client Profile api-sslcli-x2020
    - Crypto Identification C x2020_idcred
      - Crypto Key x2020_privkey
      - Crypto Certificate x2020_pubcert
    - Crypto Validation Cred x2020_valcred
      - Crypto Certificate x2020_pubcert
      - Crypto Certificate webapi-mgmt-client-in cert
      - Crypto Certificate webapi-mgmt-client-rc

Status: ● up

Name: x2020_valcred

Main

Enable administrative state: ☑

Certificates:
- x2020_pubcert
- webapi-mgmt-client-interm
- webapi-mgmt-client-rootce

Certificate Validation Mode: Match exact certificate or immediate issuer

g. Click Apply.
h. Review your configuration and save your changes.

CLI: Use the DataPower command-line interface to complete the following steps:

a. Log in to the DataPower CLI.
b. Execute the following command to add two new certificates to the server:
   Use the domain from Step1 as the value for the **switch** setting.

```
co; switch APIMgmt_BC0F2A1833 ; crypto; valcred x2020_valcred; certificate webapi-mgmt-client-
intermediatecert; certificate webapi-mgmt-client-rootcert; exit; exit
```

c. Review your configuration and then save your changes with the following command:

```
write memory
```

d. Run the following command to end the CLI session:

```
exit; exit
```

Remember: Add the new certificates to every Gateway server in the domain.

What to do next

Proceed to upgrade your API Connect deployment as explained in Upgrading your API Connect solution.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Verifying maintenance level

After applying an IBM® API Connect upgrade, verify that the installation is complete and the correct version is applied to all servers.

Procedure

1. Log in to the Cloud Manager of a new Management server and select the Services page.
2. Refresh the browser to ensure that the latest information about the services and servers is displayed.
3. Click the Server Details icon (i) to view the server details for each server.
4. Verify that the version of each Management server is the same new version. For example: 5.0.0.0.20140801-1354_H9_64.
   Note: For management servers, you can also log in to the CLI and enter the **system show version** CLI command to verify the version.
5. Verify that the Configuration version displayed for each Gateway server matches the version of the Management servers.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Upgrade the Developer Portal

You maintain the Developer Portal by applying an upgrade to take advantage of the most recent fixes and minor enhancements. If you have already installed the latest fix pack, you can also upgrade the Drupal code and contributed modules as well as the Linux system and associated packages.

V5.0.2 + By upgrading the Developer Portal, the encryption of database traffic is enabled. For each node, you are asked to confirm the encryption of database traffic by responding with **Y**. To bypass the question when you apply to the rest of the nodes, you can add **-y** to the command line of the fix pack script.

V5.0.2 + Applying the fix pack causes the database on each node to shutdown, and after all of the fix packs are applied, they start again.

- **Applying an IBM fix pack and upgrading all sites to use the new distribution**
  Upgrade all of your Developer Portal sites at the same time by applying an IBM fix pack.
- V5.0.8 + **Migrating your Developer Portal OVAs from Debian V7 to Ubuntu V16.04**
  It is strongly recommended that you migrate your Developer Portal to the Ubuntu V16.04 base now because support for Debian V7 upgrades was withdrawn in May 2018. To migrate your Developer Portal OVAs from Debian V7 to Ubuntu V16.04, you need to upgrade your existing Debian OVAs, if not already on the latest version, deploy and configure the new Ubuntu OVAs to replace them, and then shutdown all the Debian OVAs.
- V5.0.8 + **Migrating your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04**
  It is strongly recommended that you migrate your Developer Portal to Ubuntu V18.04 because support for Ubuntu V16.04 is being withdrawn in March 2021. To migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04, you need to perform a backup and restore method of migration. This method involves backing up your current Ubuntu V16.04 Developer Portal content, standing up new OVAs on Ubuntu V18.04, and then restoring the Developer Portal content onto the new OVAs.
- **Testing the upgrade on a standalone node**
  You can test your Developer Portal upgrade on a standalone node.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Applying an IBM fix pack and upgrading all sites to use the new distribution

Upgrade all of your Developer Portal sites at the same time by applying an IBM fix pack.

Before you begin

Downgrading your Developer Portal to an earlier fix pack level is **not** supported.

Download the fix pack to your local machine. The fix pack is labeled in the following format depending on which version of API Connect you are upgrading to.

V5.0.7 and earlier

```
version-APIConnect-Portal-yyyymmdd_hhmm.bin
```

V5.0.8 +

Debian V7 upgrade file:

```
version-APIConnect-Portal-Debian7-yyyymmdd-hhmm.bin
```

Ubuntu V16.04 upgrade file:

```
version-APIConnect-Portal-Ubuntu16-yyyymmdd-hhmm.bin
```

V5.0.8 + Ubuntu V18.04 upgrade file:

V5.0.8 +

```
version-APIConnect-Portal-Ubuntu18-yyyymmdd-hhmm.bin
```

Note that the version number and time stamp will vary.

V5.0.8 + Attention:

- The Linux distribution for the Developer Portal OVA has moved from a Debian V7 base to an Ubuntu base. Support for the Debian V7 OVA was withdrawn in May 2018. You are strongly encouraged to migrate your Developer Portal to the Ubuntu base, as support for Debian V7 upgrades has been removed. For more information, see Migrating your Developer Portal OVAs from Debian V7 to Ubuntu V16.04.
- From API Connect version 5.0.8.10 iFix 1, it is strongly recommended that you migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04, because support for Ubuntu V16.04 is being withdrawn in March 2021. For more information, see Migrating your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04.

It is recommended that you back up your Developer Portal. For more information, see Backing up and restoring the Developer Portal.

Alternatively, you can test your upgrade on a standalone node before upgrading your live cluster. For more information, see Testing the upgrade on a standalone node.

About this task

Complete the following steps to upgrade your Developer Portal; see upgrading to V5.0.6.3 and later or upgrading to V5.0.6.2 and earlier. If you are upgrading your Developer Portal to IBM® API Connect Version 5.0.3 specifically, and your machines are in a cluster, there are specific substeps that you must follow.

Procedure

- **(Upgrading to V5.0.6.3 and later)** To apply an IBM API Connect on a node:
  1. If your platform is UNIX or Macintosh, enter the following commands:
     Note: You need apply the fix pack only to a single machine in the portal cluster and it will upgrade all the machines and then all the portal sites in the cluster.
     V5.0.7 and earlier

     ```
     scp version-APIConnect-Portal-yyyymmdd-hhmm.bin admin@my.portal.com:
     ```

     ```
     ssh admin@my.portal.com
     ```

     V5.0.8 + Upgrading on a Debian base:

     ```
     scp version-APIConnect-Portal-Debian7-yyyymmdd-hhmm.bin admin@my.portal.com:
     ```

     ```
     ssh admin@my.portal.com
     ```

     V5.0.8 + Upgrading on an Ubuntu V16.04 base:

     ```
     scp version-APIConnect-Portal-Ubuntu16-yyyymmdd-hhmm.bin admin@my.portal.com:
     ```

     ```
     ssh admin@my.portal.com
     ```

     V5.0.8 + Upgrading on an Ubuntu V18.04 base:

     ```
     scp version-APIConnect-Portal-Ubuntu18-yyyymmdd-hhmm.bin admin@my.portal.com:
     ```

     ```
     ssh admin@my.portal.com
     ```

     If your local platform is Windows then use PuTTY, or a similar application, to secure copy the .bin to the Developer Portal and then secure shell log in as the admin user to the Developer Portal.
  2. To apply the fix pack in interactive mode on a single machine in your cluster (or the single standalone machine), run the following command:
     V5.0.7 and earlier

     ```
     bash version-APIConnect-Portal-yyyymmdd-hhmm.bin
     ```

     V5.0.8 + Upgrading on a Debian base:

     ```
     bash version-APIConnect-Portal-Debian7-yyyymmdd-hhmm.bin
     ```

     V5.0.8 + Upgrading on an Ubuntu V16.04 base:

     ```
     bash version-APIConnect-Portal-Ubuntu16-yyyymmdd-hhmm.bin
     ```

V5.0.8 + Upgrading on an Ubuntu V18.04 base:

```
bash version-APIConnect-Portal-Ubuntu18-yyyymmdd-hhmm.bin
```

To automatically answer yes to any questions, run the command with the `-y` option. For example:

```
bash version-APIConnect-Portal-Ubuntu16-yyyymmdd-hhmm.bin -y
```

- **(Upgrading to V5.0.6.2 and earlier)** Apply an IBM API Connect fix pack.
  1. If your platform is UNIX or Macintosh, enter the following commands:
     Note: If your machines are in a cluster, be sure to apply the fix packs one by one to all machines in that cluster. Ensure that the scripts finish before moving onto the next machine.

     ```
     scp version-APIConnect-Portal-yyyymmdd-hhmm.bin admin@my.portal.com:
     ```

     ```
     ssh admin@my.portal.com
     ```

     If your local platform is Windows then use PuTTY, or a similar application, to secure copy the *version*-APIConnect-Portal-*yyyymmdd-hhmm*.bin to the Developer Portal and then secure shell log in as the admin user to the Developer Portal.
  2. When the file has copied, complete one of the following steps. The steps are dependent on whether you have a standalone node, or a cluster of nodes:
     - If you have a standalone machine, enter the following command:

       ```
       bash version-APIConnect-Portal-yyyymmdd-hhmm.bin -a -i
       ```

     - If you have a cluster of machines, enter the following command for all machines except the last machine in the cluster:

       ```
       bash version-APIConnect-Portal-yyyymmdd-hhmm.bin -a
       ```

       then enter the following command for the final machine in the cluster:

       ```
       bash version-APIConnect-Portal-yyyymmdd-hhmm.bin -a -i
       ```

If you are upgrading your Developer Portal to IBM API Connect Version 5.0.3 specifically, and your machines are in a cluster, complete the following sub-steps:

-
  1. Run the following command individually on each machine in the cluster, ensuring that the command finishes on a machine before running the command on the next machine:

     ```
     bash 5.0.3.0-APIConnect-Portal-20160803-0132.bin -ly
     ```

     There is a short period of down time on each machine when you run the `-ly` command, as the new database software is installed and the database is restarted.
  2. After you have run the command on each machine, run the following command on all of the machines apart from the last machine, ensuring that the command finishes on a machine before running the command on the next machine:

     ```
     bash 5.0.3.0-APIConnect-Portal-20160803-0132.bin -say
     ```

     Each machine stops serving web traffic after the `-say` command has run on it.
  3. On the last machine, run the following command:

     ```
     bash 5.0.3.0-APIConnect-Portal-20160803-0132.bin -sayi
     ```

     All machines restart and begin serving traffic after the `-sayi` command has run on the last machine

- **Switching themes after upgrading**
  The theme of the Developer Portal does not affect functionality, but you can choose to manually switch to the latest Developer Portal theme after upgrading. You can upgrade a custom theme that is based on an IBM API Management Version 4.0 theme to one that is based on the IBM API Connect Version 5.0 theme.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Switching themes after upgrading

The theme of the Developer Portal does not affect functionality, but you can choose to manually switch to the latest Developer Portal theme after upgrading. You can upgrade a custom theme that is based on an IBM® API Management Version 4.0 theme to one that is based on the

IBM API Connect Version 5.0 theme.

Before you begin

You must have administrator access to complete this task.

About this task

A Developer Portal is switched to the latest theme if it has not had a custom theme previously.

Note: Manually switching the theme is a task that can occur only after the upgrade from IBM API Management Version 4.0 to IBM API Connect Version 5.0.
Switching themes after upgrading is optional and does not affect the functionality of the Developer Portal if it does not occur.

Procedure

To switch to the latest Developer Portal theme, proceed with one of the following steps:

1. On the Developer Portal administrator dashboard,
   a. Click Appearance, then click Enable and set default for the latest Developer Portal theme.
      The latest Developer Portal theme is labeled as IBM API Connect 7.x-*API_Connect_version_number*.
   b. Click Disable for the previous Developer Portal theme. The previous Developer Portal theme is labeled as IBM APIM 7.x-*API_Connect_version number*.
   c. Click Save configuration.
      The latest Developer Portal theme is enabled and displayed.
   d. Click Structure > Pages, then click edit adjacent to the welcome page that you are no longer using.
   e. Click Edit for Path.
   f. In the Path text field, change the value from home to any alias that is not home.
      For example:

      ```
      pre_v5_home
      ```

   g. Click Update and save.
   h. On the administrator dashboard, click Structure > Pages, then click edit adjacent to v5welcome_page or welcome-page.
   i. Click Edit for Path.
   j. Enter home in the Path text field, then click Update and save.
      The latest Developer Portal front page is displayed.
2. You can create a sub-theme. For more information, see Using a sub-theme.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Migrating your Developer Portal OVAs from Debian V7 to Ubuntu V16.04

It is strongly recommended that you migrate your Developer Portal to the Ubuntu V16.04 base now because support for Debian V7 upgrades was withdrawn in May 2018. To migrate your Developer Portal OVAs from Debian V7 to Ubuntu V16.04, you need to upgrade your existing Debian OVAs, if not already on the latest version, deploy and configure the new Ubuntu OVAs to replace them, and then shutdown all the Debian OVAs.

Before you begin

Upgrade to the 5.0.8.3-APIConnect-Portal-Debian7-20180508-1349 Debian interim fix that is available from IBM® Support Fix Central: https://www-945.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~WebSphere&product=ibm/WebSphere/IBM+API+Connect&release=5.0.8.3&platform=All&function=all.

For upgrade instructions, see Applying an IBM fix pack and upgrading all sites to use the new distribution.

Also, download the 5.0.8.3-APIConnect-Portal-Ubuntu16-20180508-1349 Ubuntu OVA file, and the latest 5.0.8.x fix pack/interim fix (to upgrade the Ubuntu servers after the migration).

Debian V7 upgrade file

```
5.0.8.3-APIConnect-Portal-Debian7-20180508-1349.bin
```

Ubuntu V16.04 OVA file

```
5.0.8.3-APIConnect-Portal-Ubuntu16-20180508-1349.ova
```

It is recommended that you back up your Developer Portal before you start this process. For more information, see Backing up and restoring the Developer Portal.

Pre-migration checks

Ensure that your Debian portal servers successfully upgraded to the most recent IBM API Connect Version 5.0.8.3 interim fix by completing the following checks:

1. Run the **status** command, and confirm that everything is reported as up and that no errors are shown. The command returns the following values:

   ```
   System version: 7.x-5.0.8.3-yyyymmdd-hhmm
   ```

   The date and time of the System version should match that of the upgrade file that was installed.

   ```
   Distribution version: 7.x-5.0.8.3-yyyymmdd-hhmm
   ```

   The date in the Distribution version, but not the time, should be no more than one day older than the upgrade file that was installed. Make a note of this value.
2. Run the command **list_sites -p**. Confirm that all sites are listed as INSTALLED, and that all have the platform setting that matches the Distribution version from step 1; for example, `platform_devportal_7_x_5_0_8_3_yyyymmdd-hhmm`. If any sites are listed as being on an earlier platform, they should be upgraded to match the Distribution version, by using the **upgrade_devportal** command.
3. Run the command **list_platforms** and confirm that a platform that matches the Distribution version from step 1 is listed; for example, `platform_devportal_7_x_5_0_8_3_yyyymmdd-hhmm => devportal-7.x-5.0.8.3-yyyymmdd-hhmm : Template Exists (default)`. Any older platforms can be deleted by using the **delete_platform** command; deleting these platforms saves time later when the portal replicates with other cluster members.
4. Run the command **check_site -a** and confirm that all sites are up and are returning `200` HTTP codes.
5. Run the command **ready_to_migrate** to confirm that all sites are in a valid state to migrate.

If you encounter any problems when you do the pre-migration checks, do not proceed with the migration. Instead, open a support request and state that your Debian nodes failed the pre-migration checks and attach the output file from the **generate_logs** command.

About this task

The Linux® distribution for the Developer Portal OVA moved from a Debian V7 base to an Ubuntu V16.04 base. Support for the Debian V7 OVA was withdrawn in May 2018. The following steps describe how to upgrade your Developer Portal platform software to upgrade your sites to use the new distribution, and then to migrate the OVA to the Ubuntu base. After the Developer Portal is migrated to Ubuntu, future upgrades can be performed by using the standard instructions for applying an IBM fix pack, see Applying an IBM fix pack and upgrading all sites to use the new distribution.

Important: You can complete the migration by using one of the following methods:

- **(Preferred)** Cluster the new Ubuntu nodes with the existing Debian nodes. After you complete the clustering, you remove the Debian nodes from the cluster.
- Migrate by using a backup file. Use this method if you have only one active Developer Portal virtual machine that is running at a time. The backup file method should be done on a stand-alone Debian node to a stand-alone Ubuntu node only.

Procedure

- Migrating by clustering new Ubuntu nodes with the existing Debian nodes.
  1. Ensure that you complete the pre-migration checks described previously, and that the Distribution version of your Debian nodes matches that of the Ubuntu nodes that you are migrating to.
  2. If your Developer Portal is a stand-alone Debian machine, you must set that machine to be in a cluster of one member. Run the following command:

     ```
     [[ $(dbstatus) == "Standalone" ]] && set_cluster_members -c
     ```

3. Deploy the Ubuntu V16.04 OVAs. You must deploy as many Ubuntu OVAs as there are Debian OVAs in your Developer Portal cluster. Deploy the portal OVAs as described in Deploying the Developer Portal OVA file, and then configure them as described in Installing the Developer Portal; you must complete **all** the steps described, ensuring that the Management cluster that you specify is the same as that used by the Debian nodes.
4. Ensure that the Ubuntu nodes report the same time and NTP server as the Debian nodes. All nodes must also have their operating system timezone set to UTC; this is the default setting so you need to change this only if it has been changed previously.
5. On the first configured Ubuntu portal node, run the following command:

   ```
   touch /var/aegir/config/nginx.conf
   ```

   and then add this node to the cluster by running the following command:

   ```
   set_cluster_members IP_address_of_an_existing_Debian_7_ova
   ```

   Repeat the `set_cluster_members` command for every Ubuntu OVA that you are adding to the cluster.
6. Run the `status` command.
   Check that every server is marked as `SUCCESS`. It might take several hours for large deployments to fully replicate across all servers.
7. If the sites on the Debian nodes are still being used, you need to restart nginx on the sites so that they can use the updated nginx config file that was replicated from the Ubuntu node. To do this, run `sudo service nginx restart` on all of the Debian nodes.
8. Add the new Ubuntu OVAs into the existing load balancing configuration for the Developer Portal user interfaces, and for the SSH communication from the Cloud Manager. For more information, see Load balancing in IBM API Connect and Load balancing the requests the Cloud Manager sends to the Developer Portal.
   If the configuration does not use a load balancer, edit the IP address for the Cloud Manager SSH communication to match one of the new Ubuntu OVAs (modify the Portal Management Address field in the Cloud Manager UI, see Installing the Developer Portal).
9. Shut down all of the existing Debian V7 OVAs, by logging in to the CLI for each Debian machine and running the following command:

   ```
   queue_run -qp && sudo halt -p
   ```

10. Log in to one of the Ubuntu V16.04 machines, and run the following commands depending on whether you are configuring a cluster of machines, or a stand-alone machine.
    To recluster with the currently visible Ubuntu machines, run the following command:

    ```
    set_cluster_members -v
    ```

    For a stand-alone Ubuntu machine, run the following command to de-cluster the single machine and convert it to stand-alone mode:

    ```
    set_cluster_members -v1
    ```

    Your cluster of Ubuntu machines, or stand-alone machine, is set up.
11. Apply the latest 5.0.8.x fix pack/interim fix that is available on IBM Support Fix Central.

- Migrating by using a backup file. If you can have only one virtual machine at a time in your environment, you must complete the following instructions to migrate your Developer Portal OVA from Debian V7 to Ubuntu V16.04. The backup file method should be done on a stand-alone Debian node to a stand-alone Ubuntu node only.
  1. Complete the pre-migration checks described previously.
  2. If the Debian deployment is part of a cluster, reduce it down to a single node by using the `set_cluster_members -d` command.
  3. Back up your Debian Developer Portal by using the `backup_devportal` command, and then save the backup to a separate FTP/SFTP server.
     For more information, see backup_devportal.
     Important: You must save your backup to a separate server. Otherwise the backup is lost when you delete your Debian Developer Portal in the next step.
  4. Delete your Debian Developer Portal virtual machine.
  5. Deploy the Ubuntu Developer Portal OVA in place of the deleted Debian one, and set the same IP address and host name as the deleted Debian OVA.
     For detailed instructions, see Deploying the Developer Portal OVA file.
  6. Configure the Ubuntu Developer Portal to connect to the management server. For detailed steps, see Installing the Developer Portal.
  7. Copy the Debian Developer Portal backup that you created in Step 3 to the newly deployed Ubuntu server by using FTP, SFTP, or SCP. Then, restore the site by using the `restore_devportal -scn` command.
     For example:

     ```
     restore_devportal -scn backup_file_absolute_path
     ```

where ***`backup_file_absolute_path`*** is the location of your backup file. For more information, see restore_devportal.

8. Run the **`status`** command.
   Check that the server is marked as **`SUCCESS`**.
9. Apply the latest 5.0.8.x fix pack/interim fix that is available on IBM Support Fix Central.

Results

You have migrated your Developer Portal OVAs from Debian V7 to Ubuntu V16.04.

What to do next

When you next upgrade to Developer Portal releases in the API Connect Version 5.0.x stream, you must use the *version*-APIConnect-Portal-Ubuntu16-*yyyymmdd*-*hhmm*.bin upgrade file.
Note: From API Connect version 5.0.8.10 iFix 1, it is strongly recommended that you migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04, because support for Ubuntu V16.04 is being withdrawn in March 2021. For more information, see Migrating your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04.

Related concepts

- Useful commands for use with a running node

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Migrating your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04

It is strongly recommended that you migrate your Developer Portal to Ubuntu V18.04 because support for Ubuntu V16.04 is being withdrawn in March 2021. To migrate your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04, you need to perform a backup and restore method of migration. This method involves backing up your current Ubuntu V16.04 Developer Portal content, standing up new OVAs on Ubuntu V18.04, and then restoring the Developer Portal content onto the new OVAs.

Before you begin

Ubuntu V18.04 is available from Developer Portal version 5.0.8.10 iFix 1, for example 5.0.8.10-iFix-APConnect-Portal-Ubuntu18-20201217-2319.ova. Note that the version number and time stamp will vary depending on the version that you download.

Ensure that your Developer Portal is at version 5.0.8.8 or later. This release and all later versions are available from IBM® Support Fix Central: https://www.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~WebSphere&product=ibm/WebSphere/IBM+API+Connect&release=5.0.8.8&platform=All&function=all.

For upgrade instructions, see Applying an IBM fix pack and upgrading all sites to use the new distribution.

Also, download the latest Developer Portal OVA file (minimum version 5.0.8.10 iFix 1), which will contain Ubuntu V18.04, and will be used to deploy onto the new OVAs.

Pre-migration checks

Verify that your Ubuntu V16.04 portal servers are at version 5.0.8.8 or later, by completing the following checks:

1. Run the **status** command, and confirm that everything is reported as up and that no errors are shown. The command returns the following values:

   ```
   System version:  7.x-5.0.8.8-yyyymmdd-hhmm
   ```

   The date and time of the System version should match that of the upgrade file that was installed.

   ```
   Distribution version:  7.x-5.0.8.8-yyyymmdd-hhmm
   ```

The date in the Distribution version, but not the time, should be no more than one day older than the upgrade file that was installed. Make a note of this value.

2. Run the command **list_sites -p**. Confirm that all sites are listed as `INSTALLED`, and that all have the platform setting that matches the Distribution version from step 1; for example, `platform_devportal_7_x_5_0_8_8_yyyymmdd-hhmm`. If any sites are listed as being on an earlier platform, they should be upgraded to match the Distribution version, by using the **upgrade_devportal** command.
3. Run the command **check_site -a** and confirm that all sites are up and are returning `200` HTTP codes.

If you encounter any problems when you do the pre-migration checks, do not proceed with the migration. Instead, open a support request and state that your Ubuntu V16.04 nodes failed the pre-migration checks, and attach the output file from the **generate_logs** command.

About this task

The Linux® distribution for the Developer Portal OVA moved from an Ubuntu V16.04 base to an Ubuntu V18.04 base within Developer Portal V5.0.8.10 iFix 1. Support for the Ubuntu V16.04 OVA is being withdrawn in March 2021. The following steps describe how to backup your Developer Portal platform, and then to standup new OVAs on which to restore your Developer Portal content. After the Developer Portal is migrated to Ubuntu V18.04, future upgrades can be performed by using the standard instructions for applying an IBM fix pack, see Applying an IBM fix pack and upgrading all sites to use the new distribution.

Procedure

Complete the following steps for both single-node and multi-node migrations.

1. Ensure that you complete the pre-migration checks described previously.
2. Back up your entire Developer Portal by following the instructions in Backing up your whole Developer Portal. In a multi-node environment, you need to back up the Developer Portal only on one node.
   Important: You must save your backup to a remote server, so that the backup can be restored onto the new Ubuntu V18.04 OVA.
3. Optional: If you want to keep the hostname details of your old OVAs, for each OVA run the following command and make a note of each hostname:

   ```
   hostname -f
   ```

   Then, before restoring the Developer Portal onto each OVA, you can use the returned values to set the hostname for each new Ubuntu V18.04 OVA.
4. You must now shut down all of the legacy Ubuntu V16.04 OVAs before continuing with the restore.
5. Deploy the Ubuntu V18.04 portal OVAs as described in Deploying the Developer Portal OVA file. You must deploy as many Ubuntu V18.04 OVAs as there are Ubuntu V16.04 OVAs in your Developer Portal cluster.
   Warning: Upon deployment, if you are experiencing trouble with a static IP configuration where your machine is not being assigned the given static IP address, then this can be resolved by flushing the DNS cache and restarting your Developer Portal Machine.
6. On the new Ubuntu V18.04 OVA, download the backup from the remote server that you created earlier. In a multi-node deployment, download the backup onto one machine only.
7. On each new OVA, set the hostname of the machine by running the following command:

   ```
   set_hostname hostname
   ```

   Where ***`hostname`*** is either the value that you saved from the equivalent Ubuntu V16.04 node, or a new hostname for this upgraded machine.
8. If you are using a static IP configuration for each Developer Portal machine, then ensure that the IP details are correctly set before you continue. If you are using a DHCP IP configuration, ensure that your DHCP server correctly allocated an IP address for each Developer Portal Machine. If the IP address is incorrect, you might find that your Developer Portal sites are restored, but they are not accessible by the API Manager or through a web browser. For more information, see Installing the Developer Portal.
9. If you didn't use the same hostname details for the new OVAs, and you're using a load balancer to manage traffic between the Cloud Manager and the Developer Portal, then you must ensure that your load balancer points to the new set of hostnames. You can obtain the hostnames for each of the new OVAs by running `hostname -f`.
   If you're not using a load balancer, then ensure that the Cloud Manager configuration for the Developer Portal is changed to the new hostname. For detailed instructions, see Step 2 in Installing the Developer Portal.

   Note:
   For Developer Portal version 5.0.8.10-iFix or later, on a Ubuntu18.04 base, if you did not change your Developer Portal IP address, but now want to, ensure that you sync your new IP address with the API Manager. Use the command:

   ```
   resubscribe_webhooks
   ```

   If you do not run this command, you might not receive webhooks from the API manager as its configuration is set to be the old Developer Portal IP address.
10. Restore the Developer Portal by running the following command on the new Ubuntu V18.04 OVA. In a multi-node deployment, run the command on the machine that you downloaded the backup onto.

```
restore_devportal -c -s backup_file_absolute_path
```

where ***backup_file_absolute_path*** is the location of your backup file. For more information about this command, see restore_devportal.
Note: If you're not reusing the old hostnames for the new OVAs, but these old hostnames are used within the URL of your Portal sites, then it's best practice to update the Developer Portal site URLs for each site within the API Manager to use the new hostnames.

11. Optional: For standalone environments only, turn on SSL for MySQL.
    For some V5 standalone environments, you might get an SSL error when you try to upgrade to later releases post-migration to Ubuntu18. This error is because SSL is not turned on for MySQL for standalone, and the MySQL server was not configured correctly. Run these commands:

    ```
    mysql_certs -fgl
    stop_db
    start_db
    ```

    For a single-node migration, the migration procedure is now complete.

    For a multi-node migration, you need to create the cluster membership by continuing with the following steps.

12. On the OVA that you just restored the Developer Portal onto, run the following command to create the cluster membership:

    ```
    set_cluster_members -c
    ```

13. After the command has completed, run the **status** command and check that the server is marked as **SUCCESS**.
14. When the cluster membership has been successfully created, add the remaining machines to the cluster by running the following command on each of the other new OVAs:

    ```
    set_cluster_members hostname_of_first_member
    ```

    Where ***hostname_of_first_member*** is the hostname of the first OVA that you ran **set_cluster_members -c** on.
    It's recommended that you wait for each invocation of **set_cluster_members** to finish successfully, before running the command on the next machine. You should run the **status** command on each OVA to check that the server is marked as **SUCCESS** before moving on to the next machine.

    Note that it might take several hours for large deployments to fully replicate across all servers.

Results

You have migrated your Developer Portal OVAs from Ubuntu V16.04 to Ubuntu V18.04.

What to do next

When you next upgrade to Developer Portal releases in the API Connect Version 5.0.x stream, you must use the *version*-APIConnect-Portal-Ubuntu18-*yyyymmdd*-*hhmm*.bin upgrade file.

Related concepts

- Useful commands for use with a running node

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Testing the upgrade on a standalone node

You can test your Developer Portal upgrade on a standalone node.

About this task

You can add a new node to an existing cluster, synchronize the node, make the node standalone, and test the upgrade on it. The task enables greater confidence in the upgrade working on the live cluster as the upgrade is tested on a standalone machine that has the same Drupal sites as the live cluster.
Note: It is recommended to backup all of the nodes.

Procedure

With a cluster of 3 machines, proceed with the following steps:

1. Deploy a new machine from the same OVA version as the existing 3 cluster members, then run the following command:

   ```
   set_hostname myuniquerhostname
   ```

   where *myuniquerhostname* is a unique host name.
2. Add the new machine to the cluster using the following command:

   ```
   set_cluster_members IP_or_hostname_of_existing_cluster_member
   ```

   Note: You can deploy and configure the machine in advance, but you can only run `set_cluster_members` when you are ready to test the upgrade.
3. Disable the queue on the fourth machine to ensure that it is not running tasks when it is de-clustered by running the following command:

   ```
   queue_run -q
   ```

4. Continually run the `status` command until you see the following message from all four machines:
   `SUCCESS: All services are Up and the cluster timestamps are in sync.`
   Note: You must wait for the file synchronization to finish to ensure that the fourth machine retains all of its files.
5. De-cluster the fourth machine by running the following command:

   ```
   set_cluster_members -d
   ```

6. To return the other machines to a cluster of three, run the following command on the original three machines:

   ```
   set_cluster_members machine1 machine2 machine3
   ```

7. Upgrade the fourth de-clustered machine by using the fix pack.
   For more information on applying a fix pack, see Applying an IBM fix pack and upgrading all sites to use the new distribution
8. Test whether the Drupal sites work to your specifications after the upgrade has finished.
9. Power down the fourth machine as it is no longer needed.
   The upgrade process is complete.
   Note: You can power down the fourth machine regardless of whether the upgrade was successful or not. If the upgrade fails and you want to test the upgrade again, return to step 1 and proceed with the steps.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Troubleshooting the upgrade process to IBM API Connect Version 5.0 or later

You can use commands and look for errors to help solve any issues that you face during the upgrade process to IBM® API Connect Version 5.0 or later. In addition to solving issues in an upgrade to an existing system, you can apply the same answers to a test upgrade.

Useful commands

After running the upgrade command in the CLI, your appliance reboots. During the rebooting process, the UI might not display for a prolonged period of time. This can be due to a failure in the automatic conversion of your configured data from API Manager to API Connect. To check if the UI display issues are a result of failed data conversion, you can run the following CLI commands:

-
  ```
  debug tail file /var/log/migration.log
  ```

  You might see an error in the log, with recommended actions that you can follow.

-
  ```
  debug tail file /var/log/cmc.out
  ```

  You might see an error in the log, with recommended actions that you can follow. You are recommended to monitor `cmc.out` for a few minutes to check whether the output changes. If the output does not change, the upgrade has failed to complete correctly.

-
  ```
  debug postmortem generate fulllogs
  ```

Then, run the following command:

```
debug postmortem export ftp ftpserver_hostname user my_user_name file file_path_name.zip
```

After you run the command, you are prompted to enter your password.

All of the logs are extracted from your appliance. The logs are used if you contact IBM Support. The logs can also be used to help solve issues for systems that have completed the upgrade. For more information, see Debugging commands.

Analytics information is not displayed

If your analytics information is not displayed in the API Manager or in the Cloud Management Console immediately after an upgrade, the following information might be useful:

- Analytics data is not migrated from version 4.x to version 5.x. If you are migrating from version 4.x, your previous analytics data is no longer available in the upgraded environment.
- If you are upgrading from version 5.x to a later version of 5.x, restart your management nodes to resolve the issue.

Obtaining further information

Depending on the issue that you are trying to solve, IBM Support might require further information.

- OpenAPI (Swagger 2.0) representation of an API is useful if your API definition has OpenAPI (Swagger 2.0) errors, does not invoke successfully, or does not provide the correct response when it is invoked.
- If you cannot invoke an API successfully, log files from DataPower® appliances might be used in addition to OpenAPI (Swagger 2.0) representation to help solve the issue.
- If your APIs are displayed incorrectly in the Developer Portal, log files from the Developer Portal site might be used to help solve the issue.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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, manage, and monitor the API Connect on-premises cloud.

You can use the Cloud Manager to define the API Connect cloud by:

- Connecting up the relevant physical or virtual servers that are required for the cloud
- Configuring the appropriate security certificates
- Connecting to an existing SMTP mail server
- Configuring load balancing settings (if applicable)
- Uploading an IBM WebSphere® DataPower® Extension file to extend the capability of the Gateway servers.

You can manage the API Connect cloud by using the Cloud Manager to complete the following tasks:

- Adding, editing, and removing servers
- Restarting and rebooting servers
- Gathering postmortem information

You can monitor the API Connect cloud by using the Cloud Manager to complete the following tasks:

- Monitoring the status of the servers that comprise the cloud
- Viewing the health metrics that are relevant to the different types of server in the API Connect cloud; for example, processor usage, memory usage, disk space, and transaction rate
- Monitoring system alerts and user interface activity

You can use the Cloud Manager to configure the existing API Connect cloud to update settings at the cloud level; for example, security certificates, load balancing options, and data storage settings.

You can use the Cloud Manager to manage organizations.

- **Accessing the cloud console user interface**
  Navigate to and log in to the Cloud Manager user interface.
- **Defining the cloud**
  Define the on-premises API Connect cloud so that your company can create, promote, and track APIs.
- **Managing the cloud**
  Manage the servers in your cloud.
- **Monitoring the cloud**
  You can check the status of your servers, view detailed metrics about server usage, and see activities and alerts. V5.0.7+ You can also configure destination targets for the analytics data that is captured for your API Connect on-premises cloud.
- **Administering provider organizations**
  Manage the provider organizations that share in the use of your API Connect cloud.
- **Administering user access**
  If you have either the Cloud Administrator or Cloud Owner role in the Cloud Manager, you can add additional administrators and users, and assign them roles, enabling them to perform administrative duties and manage provider organizations.
- **Authentication**
  As the Cloud Manager administrator, you can add new user registries to securely authenticate your Catalogs and APIs and create SSL profiles that secure transmission of data through web sites.
- **Changing your Cloud Manager password**
  You can change your Cloud Manager password.
- **Changing the expiration settings for Cloud Manager user accounts**
  You can change the expiration settings for your Cloud Manager user accounts by using REST API calls.
- **Troubleshooting**
  Troubleshooting topics for the Cloud Manager.

Related information

- Firewall requirements

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Accessing the cloud console user interface

Navigate to and log in to the Cloud Manager user interface.

Before you begin

The first time that you access the Cloud Manager user interface, you must change the Cloud Administrator password and configure the API Connect cloud.
To access the Cloud Manager user interface after you install IBM® API Connect, you must log in.

Procedure

To log in to the Cloud Manager, complete the following steps:

1. In a web browser, enter the URL of the Management server.
   For example, `https://`*`ManagementHostname.domain`*`/cmc` where *ManagementHostname.domain* is the fully qualified host name or IP address of the initial Management server. For example:
   - MySubdomain.MyDomain.com
   - 9.51.111.121

   The cloud console login window opens.
2. Enter the Cloud Administrator user name and password. If you did not change the user name and password during the installation process, enter the default values of `admin` for the user name and `!n0r1t5@C` for the password.
   Note: 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, please ensure that you have typed the password correctly.
3. 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.

Results

The Cloud Manager user interface opens.

What to do next

Now you can define your API Connect on-premises cloud. For more information, see Defining the cloud.

Related tasks

- Changing your Cloud Manager password
- Defining the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Defining the cloud

Define the on-premises API Connect cloud so that your company can create, promote, and track APIs.

About this task

After you define your API Connect cloud, you can invite other users to access the API Manager user interface to create APIs. You can then expose these APIs to the development community through the Developer Portal user interface.

Define the API Connect cloud by completing the following tasks:

- **Configuring the Management service**
  IBM API Connect automatically detects the host name, user name, and password combination to use to log on to the first Management server.
- **Configuring the initial Gateway service**
  You must define your initial Gateway service before you configure the rest of your API Connect on-premises cloud. The Gateway service is a cluster of one or more of the gateway servers.
- **Configuring your topology**
  To provide load balancing and fail over in your environment, you can use the cloud console to configure two or more servers in your Management and Gateway services.
- **Specifying the cloud settings**
  Before you can add a provider organization to your cloud, you must first specify your cloud settings. A default setting is used for the DNS scheme and sandbox DataPower Service. However, you must specify the email server configuration and also review the user registry options. API Connect supports LDAP as an external IDP.
- **Configuring your Gateway server extensions**
  You can add extra IBM DataPower® enforcement capabilities to a Gateway service by uploading an IBM DataPower exported configuration .zip file.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the Management service

IBM® API Connect automatically detects the host name, user name, and password combination to use to log on to the first Management server.

Before you begin

You must complete the following task:

- Accessing the cloud console user interface

Ensure that the Management server is running and available on your network.

About this task

Your initial Management server is automatically added to your cloud, so that you can manage the overall operations of the various servers in the API Connect cloud.

The Management service consists of the first Management server that you defined during the API Connect installation.

Within the API Connect cloud, Management servers store all of the cloud configuration and control communication between the other servers.

One Management server is automatically defined in the cloud console when you install API Connect. If required, you can add further Management servers to your cloud.

In the Cloud Manager, each server is added as a member of a service.

If you plan to use more than one Management server in your on-premises cloud, you must complete the following steps:

Procedure

1. Set up an external load balancer with only one Management server in the load balancer group.
   You use an external load balancer to provide load balancing for the user interfaces.
2. Enter the load balancer FQDN in to the Management service configuration (see Defining the Management service).
3. Add the second Management server to the Management service by using the API Connect cloud console (see Adding more Management servers).
4. Add the second Management server to the load balancer.
5. Repeat the previous steps for all other Management servers that you want to add to your on-premises cloud.

Defining the Management service

Procedure

The Management service is a cluster of one or more of the Management servers. To configure your Management service, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the Management Services pane, click the Service Settings icon ⚙.
3. If you are using only one Management server in your Management service, check that the Address field contains the FQDN (fully qualified domain name) of the server, for example, *mgmt01.acme.com*. If you plan to add additional servers, change the value of this field to the FQDN of the external load balancer.

4. Optional: Define an internal load balancer by completing the following steps:
   a. Select Use a different hostname for internal communication..
   b. Enter the load balancer FQDN in the Hostname for internal communication field, and click Save Service.
      For more information on using an internal load balancer, see Using a load balancer to optimize internal communications.
5. Optional: To change the name of the Management service, enter a new value in the Display Name field.
6. When you are finished, click Save Service.

Results

The Management service cloud settings are configured.

What to do next

Complete the following task:

- Configuring the initial Gateway service

Related tasks

- Defining the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the initial Gateway service

You must define your initial Gateway service before you configure the rest of your API Connect on-premises cloud. The Gateway service is a cluster of one or more of the gateway servers.

Before you begin

You must complete the following task:

- Configuring the Management service

About this task

You must configure all of the Cloud Manager settings for the initial Gateway service.
In the Cloud Manager, each server is added as a member of a service.

If you want to change these settings after you define the cloud, you must remove all of the Gateway servers first.

Procedure

To configure the Cloud Manager settings for the initial Gateway service, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the DataPower Services pane, click the Service Settings icon ⚙.
3. Optional: To change the name of the Gateway service, enter a new value in the Display Name field.
4. In the Address field, enter the virtual IP address or host name that is to be used for inbound API calls, or of an external load balancer if one is being used.
   V5.0.6+ If you have configured your API Connect cloud to use Dynamic DNS, you can specify the same host name in the Address field for two or more Gateway services, to give the appearance that the APIs that are deployed to the separate Gateway services are on the same Gateway. For more information, see Configuring multiple Gateway services to have the same host name.
5. In the Port field, enter the API data port for inbound API calls.
6. In the Port Base field, enter the reserved port base for internal traffic. Note that the Gateway service's Port Base value should represent a set of 10 ports and should not overlap with another cluster's set of 10 ports or with any ports in use by other non-API Management applications on the DataPower appliance.
7. V5.0.7+ In the Supported Application Types field, select the type of application traffic that can flow through this DataPower® Gateway service. The options are as follows:
   - Production: this Gateway service is for use with production applications only.
   - Development: this Gateway service is for use with development applications only.
   - Both (default): this Gateway service supports both development and production applications.

   For more information, see Managing the application lifecycle.
8. Optional: You can select a TLS profile from the TLS Profile list to provide an SSL Private Key and Certificate to replace the self-signed certificate that has been automatically generated by API Connect. These are used to identify the HTTP/S Server that receives API calls made to this service. API Connect supports PCKS #12 files containing the key and certificate and those may be protected by a password.
   By default, a self-signed certificate is used.
9. Optional: Configure TLS to secure API calls that are made to this Gateway service. You can specify a single TLS profile to secure all API calls, or you can use Server Name Indication (SNI) to specify which TLS profile should be used depending on the host name that the API is attempting to connect to. 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.
   To specify a single TLS certificate, complete the following steps:
   a. Select TLS Profile.

b. Select the required TLS profile from the list.

To use SNI, complete the following steps:

a. Select Server Name Indication Profile.
b. In the Hostname field enter a host name, and select the TLS Profile to be used when API calls are made to that host name.
c. Click Add to map further host names to TLS profiles.
d. Use the arrow keys to reorder the mappings as required. When an incoming API call is received, the list of mappings is searched in order until the first match is found.

If the incoming request contains a TLS SNI extension that consists of a host name, it is matched against the SNI hostname map, and the credentials of the TLS profile that corresponds to the first matching entry is returned. If no match is found in the SNI host name map, then the connection is terminated with an error. To avoid such a connection error, specify a wildcard (*) as the final entry in the host name map, mapped to a TLS profile that serves as a catchall, then if none of the other host names in the SNI host name map match, the wildcard (*) entry serves as a fallback.

If the incoming request does not contain a TLS SNI extension, the first wildcard (*) entry is chosen, and the credentials associated with that TLS profile are used for communication. If no wildcard (*) entry exists, the connection is terminated due to a lack of a TLS SNI extension in the request.

For example, consider the following SNI mapping configuration:

Table 1. SNI mapping example 1

Host name	TLS profile
abc.domain.com	TLS profile 1
*.domain.com	TLS profile 2
*	Default profile

If an API call to `abc.domain.com` is received, TLS profile 1 is used. If an API call to `xyz.domain.com` is received, TLS profile 2 is used. If the host name does not match `abc.domain.com` or `*.domain.com`, the wild card (*) entry is used. Specify a wild card (*) as the final entry in the table to explicitly handle any TLS client that does meet any of the previous mapping requirements.

With the following mapping table, a request from a TLS client that uses a host name of `acme.com` is rejected, because there is no matching wildcard (*) to handle that client:

Table 2. SNI mapping example 2

Host name	TLS profile
abc.domain.com	TLS profile 1
*.domain.com	TLS profile 2

Important: You cannot change the SNI configuration for an existing Gateway service. If you need to change your SNI configuration you must recreate the Gateway service.

10. Optional: You can add extra enforcement capabilities to your Gateway server by uploading an IBM DataPower exported configuration .zip file. For more information, see Configuring your Gateway server extensions.
11. You can use the automatically generated cryptographic material, or provide your own. The OAuth 2.0 secret must begin with the characters 0x (zero, lowercase x) followed by at least 64 hexadecimal characters.
12. Optional: If you configure two or more Gateway servers, you can choose which load balancing option to use. In the Load balancing section, select one of the following load balancing choices.
    - Select External or no load balancer if you configure more than one Gateway server but you do not want to use the API Connect load balancing function. This option is selected by default.
      Important: You must configure your external load balancer separately. If you do not configure an external load balancer, no load balancing is done for inbound API calls.
    - Select DataPower Gateway load balancer to use the API Connect load balancing function. Enter the unique (for the local network) Load Balancing group number for the Gateway service.
      Note: The load balancing settings for internal traffic are reused to load balance inbound API calls.

    By setting the load balancing options for the Gateway service, the workload is distributed across multiple servers and ensures that the defined configuration runs as efficiently as possible. For more information, see Load balancing in IBM API Connect.
    Restriction: VLAN is not supported if you are using the Gateway as a load balancer option.
13. When you are finished, click Save.

Results

The Gateway service cloud settings are configured.

What to do next

Add one or more Gateway servers to your Gateway service; see Adding a Gateway server.

Related tasks

- Defining the cloud
- Adding more Gateway services

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring your topology

To provide load balancing and fail over in your environment, you can use the cloud console to configure two or more servers in your Management and Gateway services.

Before you begin

You must complete the following task:

- Configuring the initial Gateway service

About this task

You can define a development environment with just a single server in each service. In a production environment, it is preferable to have at least two servers in each service. If you define two or more Management and Gateway servers and a single server fails, the failure does not interrupt the availability of the cloud.

When the Services page opens, it already contains a Management service that contains one Management server, and an initial Gateway service that contains no servers. You must add at least one Gateway server to the initial Gateway service before the environment is valid. You can add any other servers that you require at this stage, or you can add them later. For resilience, you must have at least two servers per service. You can also add further Gateway services to partition your Gateway servers across deployment environments.

For more information about servers, see IBM® API Connect overview.

Procedure

To configure your topology, complete the following steps:

1. In the Cloud Manager, click Services.
2. Add at least one Gateway server. For more information, see Adding a Gateway server.
3. Optional: Add a second Management server. For more information, see Adding more Management servers.
4. Optional: Configure additional Gateway services. For more information, see Adding more Gateway services.

Results

The API Connect servers are defined in your cloud.

- **Adding a Gateway server**
  Add a Gateway server so that you can process and manage security protocols and store relevant user and appliance authentication data.
- **Adding more Gateway services**
  When you install IBM API Connect, an initial Gateway service is created automatically. However, you can use the cloud console to create one or more additional Gateway services.
- **Adding multiple Gateway services that share a single DataPower appliance**
  You can use the Cloud Manager to create additional Gateway services that share an IBM DataPower® appliance.
- **Adding more Management servers**
  One Management server is automatically defined when you install IBM API Connect. Management servers manage the overall operations of the API Connect cloud and provide the tools to interface with the other defined servers.
- **Using a load balancer to optimize internal communications**
  You can use an internal load balancer to optimize internal communications between the Gateway servers and Management servers.

This is typically used when you want to gain finer control over load distribution, particularly if your IBM API Connect cloud spans two or more different geographical locations.

- V5.0.6 + **Configuring multiple Gateway services to have the same host name**
  If you have configured your API Connect cloud to use Dynamic DNS, you can give the same host name to two or more Gateway services. Use this capability to give the appearance that the APIs that are deployed to the separate Gateway services are on the same Gateway.

Related concepts

- Configuring and managing your server environment
- Managing the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a Gateway server

Add a Gateway server so that you can process and manage security protocols and store relevant user and appliance authentication data.

Before you begin

You must complete the following tasks:

- Configuring the Management service
- Configuring the initial Gateway service
- Optional: Adding more Gateway services

About this task

Within the API Connect on-premises cloud, Gateway servers act as proxies. Gateway servers receive inbound API traffic and route the requests to the relevant endpoints within your organization's firewall. Gateway servers also process security protocols and enforce user and appliance authentication processes.

You must add at least one Gateway server to the initial Gateway service. You can also create additional Gateway services and add servers to those services.
Note:

- For API Connect versions lower then 5.0.8.1: When adding a DataPower to a Gateway server, the AppOpt (AO) license will not be activated automatically on DataPower. To activate the AO license on DataPower, contact IBM Support for a script that will complete the operation. Another option is to upgrade to API Connect version 5.0.8.1, which does activate the AO license on DataPower.
- Adding a DataPower appliance to API Connect as a Gateway server might cause the appliance to be reloaded. As a result, in a production environment, any other services on the appliance might be unavailable for a short period until the appliance reloads.

In the Cloud Manager, each server is added as a member of a service.

For more information, see IBM API Connect overview.

Procedure

To add a Gateway server to a Gateway service, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the Services pane, click Add, and select to use a new DataPower® Gateway server.
3. Enter a name for the server.
4. V5.0.6 and earlier Enter the user name and password of the DataPower appliance, used for authentication of the SOMA requests.
   The user must be an admin or a privileged-user.
   Note: If the user is a newly created privileged user, that user must be active on DataPower. The user account is activated on first login, at which point the temporary password is changed to a permanent one.
5. V5.0.6 and earlier Enter the network interface, on the DataPower appliance, that receives the API requests from clients.
   Important:

- The actual IP address is determined from the interface definition on the DataPower appliance, and is typically accessible on the Internet. This address is also generated into the front side protocol handler of the API Connect service that mediates the client messages. If this DataPower appliance is part of a clustered configuration, this Ethernet interface is used for configuring self-balancing and the virtual IP address. This interface might be the same as the one used for the XML Management interface. If the server is added to a service that uses DataPower self-balancing, a Standby Control configuration is added automatically.
- If you are adding a Gateway VLAN interface address to a clustered configuration, the VLAN interface on the DataPower appliance must be configured to have standby control enabled. Otherwise, the error message `invalid interface name` is displayed when the interface is added to the cluster.

6. When you are finished, click V5.0.7 + Create V5.0.6 and earlier Create Server.

Results

The new Gateway server is added to the list of available servers.

What to do next

Repeat these steps to add further Gateway servers to your cloud.
Complete the following task:

- Selecting the DNS scheme

Related tasks

- Defining the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding more Gateway services

When you install IBM® API Connect, an initial Gateway service is created automatically. However, you can use the cloud console to create one or more additional Gateway services.

About this task

By creating additional Gateway services, you can configure separate API deployment environments with Gateway servers dedicated to each environment, thereby partitioning your Gateway servers across deployment environments. For example, you can configure one service of Gateway servers for your development environment, and a separate service of Gateway servers for your production environment.

Procedure

To add a Gateway service, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the Gateway Services pane, click Add DataPower Service.
3. In the Display Name field, enter a name for the Gateway service.
4. In the Address field, enter the virtual IP address or host name that is to be used for inbound API calls, or of an external load balancer if one is being used.
   V5.0.6 + If you have configured your API Connect cloud to use Dynamic DNS, you can specify the same host name in the Address field for two or more Gateway services, to give the appearance that the APIs that are deployed to the separate Gateway services are on the same Gateway. For more information, see Configuring multiple Gateway services to have the same host name.
5. In the Port field, enter the API data port for inbound API calls.
6. In the Port Base field, enter the reserved port base for internal traffic. Note that the Gateway service's Port Base value should represent a set of 10 ports and should not overlap with another cluster's set of 10 ports or with any ports in use by other non-API Management applications on the DataPower appliance.
7. V5.0.7 + In the Supported Application Types field, select the type of application traffic that can flow through this DataPower® Gateway service. The options are as follows:
   - Production: this Gateway service is for use with production applications only.

- Development: this Gateway service is for use with development applications only.
- Both (default): this Gateway service supports both development and production applications.

For more information, see Managing the application lifecycle.

8. Optional: You can select a TLS profile from the TLS Profile list to provide an SSL Private Key and Certificate to replace the self-signed certificate that has been automatically generated by API Connect. These are used to identify the HTTP/S Server that receives API calls made to this service. API Connect supports PCKS #12 files containing the key and certificate and those may be protected by a password.
   By default, a self-signed certificate is used.
9. Optional: Configure TLS to secure API calls that are made to this Gateway service. You can specify a single TLS profile to secure all API calls, or you can use Server Name Indication (SNI) to specify which TLS profile should be used depending on the host name that the API is attempting to connect to. 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.
   To specify a single TLS certificate, complete the following steps:
   a. Select TLS Profile.
   b. Select the required TLS profile from the list.

   To use SNI, complete the following steps:
   a. Select Server Name Indication Profile.
   b. In the Hostname field enter a host name, and select the TLS Profile to be used when API calls are made to that host name.
   c. Click Add to map further host names to TLS profiles.
   d. Use the arrow keys to reorder the mappings as required. When an incoming API call is received, the list of mappings is searched in order until the first match is found.

   If the incoming request contains a TLS SNI extension that consists of a host name, it is matched against the SNI hostname map, and the credentials of the TLS profile that corresponds to the first matching entry is returned. If no match is found in the SNI host name map, then the connection is terminated with an error. To avoid such a connection error, specify a wildcard (*) as the final entry in the host name map, mapped to a TLS profile that serves as a catchall, then if none of the other host names in the SNI host name map match, the wildcard (*) entry serves as a fallback.

   If the incoming request does not contain a TLS SNI extension, the first wildcard (*) entry is chosen, and the credentials associated with that TLS profile are used for communication. If no wildcard (*) entry exists, the connection is terminated due to a lack of a TLS SNI extension in the request.

   For example, consider the following SNI mapping configuration:

   Table 1. SNI mapping example 1

Host name	TLS profile
abc.domain.com	TLS profile 1
*.domain.com	TLS profile 2
*	Default profile

   If an API call to `abc.domain.com` is received, TLS profile 1 is used. If an API call to `xyz.domain.com` is received, TLS profile 2 is used. If the host name does not match `abc.domain.com` or `*.domain.com`, the wild card (*) entry is used. Specify a wild card (*) as the final entry in the table to explicitly handle any TLS client that does meet any of the previous mapping requirements.

   With the following mapping table, a request from a TLS client that uses a host name of `acme.com` is rejected, because there is no matching wildcard (*) to handle that client:

   Table 2. SNI mapping example 2

Host name	TLS profile
abc.domain.com	TLS profile 1
*.domain.com	TLS profile 2

   Important: You cannot change the SNI configuration for an existing Gateway service. If you need to change your SNI configuration you must recreate the Gateway service.
10. Optional: You can add extra enforcement capabilities to your Gateway server by uploading an IBM DataPower exported configuration .zip file. For more information, see Configuring your Gateway server extensions.
11. Optional: If you configure two or more Gateway servers, you can choose which load balancing option to use. In the Load balancing section, select one of the following load balancing choices.
    - Select External or no load balancer if you configure more than one Gateway server but you do not want to use the API Connect load balancing function. This option is selected by default.
      Important: You must configure your external load balancer separately. If you do not configure an external load balancer, no load balancing is done for inbound API calls.
    - Select DataPower Gateway load balancer to use the API Connect load balancing function. Enter the unique (for the local network) Load Balancing group number for the Gateway service.
      Note: The load balancing settings for internal traffic are reused to load balance inbound API calls.

    By setting the load balancing options for the Gateway service, the workload is distributed across multiple servers and ensures that the defined configuration runs as efficiently as possible. For more information, see Load balancing in IBM API Connect.
    Restriction: VLAN is not supported if you are using the Gateway as a load balancer option.

12. When you are finished, click Create.

Results

The new Gateway service is added to your configuration.

What to do next

Add one or more Gateway servers to your Gateway service; see Adding a Gateway server.

Related concepts

- Configuring and managing your server environment

Related tasks

- Configuring the initial Gateway service

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding multiple Gateway services that share a single DataPower appliance

You can use the Cloud Manager to create additional Gateway services that share an IBM® DataPower® appliance.

About this task

Adding multiple Gateway services on a single DataPower appliance can have the following benefits:

- provides increased utilization of DataPower appliances
- allows an external facing gateway service and an internal facing gateway service to each use a different network interface
- allows separation of different environments and data traffic, such as Development, Test, and Production, based on different network interfaces or ports
- allows a different DataPower domain to be set up per network zone on a DataPower appliance that spans multiple zones

The following tables provide settings information and examples of how to add multiple Gateway services on a single DataPower appliance. Note that the interfaces are installed in different subnets to avoid the communication issues described in the IBM Technote Multiple interfaces in the same subnet on DataPower should never be used.

Table 1. Sample DataPower appliance configuration settings used in the following scenarios

Network Interface	IP Address
eth0 (XML management interface)	1.101.3.4
eth1	1.102.3.4
eth2	1.103.3.4
eth3	1.104.3.4

Table 2. Using different network interfaces to differentiate domains on the same DataPower appliance

Cluster parameters	First service	Second service	Third service
Cluster name	Gateway service GS1	Gateway service GS2	Gateway service GS3
Cluster address	1.101.3.4	1.102.3.4	1.103.3.4
Cluster port	443	443	443
Auto-generated domain	APIMgmt_1234567890	APIMgmt_2345678901	APIMgmt_3456789012

Cluster parameters	First service	Second service	Third service
Server name	Gateway server GS1	Gateway server GS2	Gateway server GS3
Server address	1.101.3.4	1.101.3.4	1.101.3.4
Server port	5550	5550	5550
Server network interface	eth0	eth1	eth2

Table 3. Using different Gateway service ports to differentiate domains on the same network interface on the same DataPower appliance

Cluster parameters	First service	Second service	Third service
Cluster name	Gateway service GS4	Gateway service GS5	Gateway service GS6
Cluster address	1.104.3.4	1.104.3.4	1.104.3.4
Cluster port	443	444	445
Auto-generated domain	APIMgmt_4567890123	APIMgmt_5678901234	APIMgmt_67890122345
Server name	Gateway server GS4	Gateway server GS5	Gateway server GS6
Server address	1.101.3.4	1.101.3.4	1.101.3.4
Server port	5550	5550	5550
Server network interface	eth3	eth3	eth3

You can determine the auto-generated domain name for a gateway service by completing the following steps:

1. Log in to the Cloud Manager user interface.
2. Select the Services tab.
3. In the DataPower Services section, click the Service Settings icon ⚙ for the required gateway service. The auto-generated domain name is displayed in the DataPower Domain field.

Note: The Gateway server address and port parameters are used by the Management server to communicate to the Gateway server for management purposes. The management server configures the DataPower appliance using the XML management interface. The default server port is 5550, but the port depends on the XML management interface setup in the DataPower appliance. The Gateway server network interface is used for data traffic to that Gateway server as part of the Gateway service that the server is added to.
Before you add multiple Gateway services to a single DataPower appliance, consider the following requirements and restrictions:

- The combination of a network interface and a port must be unique for each domain on a given DataPower appliance.
- If there are other domains defined on the DataPower appliance that you plan to use, ensure that the setup for those domains does not conflict in terms of network interfaces or range of ports with the new domains that you plan to deploy in this API Connect cloud. You can contact your DataPower administrator to check the current configuration of network interfaces and ports.
- If required, you can create more network interfaces. For more information, see IBM DataPower version 6.0.1 product documentation.
- Do not add more than one Gateway server representing the same DataPower appliance to the same Gateway service. Since a given Gateway service has a single DataPower domain associated with it, a domain cannot be deployed to a given DataPower appliance more than once.
- The Gateway Cluster's Port Base value should represent a set of 10 ports and should not overlap with another service's set of 10 ports or with any ports in use by other non-API Management applications on the DataPower appliance.
- To enable the Gateway load balancer option in the Gateway service settings, ensure that each Gateway server representing a given DataPower appliance uses a different network interface. It is not sufficient for these different Gateway servers to only be differentiated by API data port used on the same network interface. The actual network interfaces must be different. The preferred approach is to use the External or no load balancer option. For more information about how to set up load balancing, see Load balancing in IBM API Connect.

Procedure

To share a DataPower appliance across Gateway services, complete the following steps:

1. Plan the Gateway topology that you want to configure in your cloud. Decide how many Gateway services and Gateway servers that you want to add. Ensure that your plan accounts for the requirements and restrictions outlined in the previous section.
2. Optional: Contact your DataPower administrator to check if there might be any usage conflict for the network interfaces and port numbers that you are planning to use. You might also need to request extra network interfaces.
3. Add the number of Gateway services that you want to use. For more information, see Adding more Gateway services.
4. Add Gateway servers to Gateway services to match your required topology. Make sure to wait for addition of a given gateway server to complete fully before using the Cloud Manager for any other tasks. For more information, see Adding a Gateway server.
5. Verify that each added Gateway server is reported as active in the Cloud Manager.
6. Create Catalogs that use the new Gateway services, and stage and publish Plans with APIs to these Catalogs.

Related concepts

- Configuring and managing your server environment

Adding more Management servers

One Management server is automatically defined when you install IBM® API Connect. Management servers manage the overall operations of the API Connect cloud and provide the tools to interface with the other defined servers.

About this task

Within the API Connect on-premises cloud, Management servers store all of the cloud configuration and control communication between the other servers.

Although the initial Management server is automatically defined, you can add further Management servers to your cloud if they do not have prior configuration in place. To clear a server configuration, run the CLI command **system clean apiconfig** from the Management server you wish to add to the Management service.

Note: Changing the configuration of a server can disrupt user activities, so, ensure that all users are disconnected from the user interface before you change a Management server. For example, by adding or removing servers, or updating host names.
In the Cloud Manager, each server is added as a member of a service.

Procedure

To add a Management server, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the Management Services pane, click Add Server.
3. Enter a name for the server.
4. Enter the IP address or host name of the server.
5. Enter the user name and password for the Management server, which were specified during the initialization of the server.
   Note: Specify the password you use to log into the Cloud Manager of the Management server you named in the previous step. This password can be different from the password you use to log into the CLI of the Management server you named in the previous step.
6. When you are finished, click Create Server.

Results

The new Management server is added to the list of available servers.

What to do next

Repeat these steps to add further Management servers to your cloud.

Related concepts

- Configuring and managing your server environment

Related tasks

- Adding a Gateway server

Using a load balancer to optimize internal communications

You can use an internal load balancer to optimize internal communications between the Gateway servers and Management servers. This is typically used when you want to gain finer control over load distribution, particularly if your IBM® API Connect cloud spans two or more different geographical locations.

About this task

By default, gateway servers in a given cloud communicate with all active Management servers in that cloud to receive configuration updates and post analytics data. To optimize server communications, you can choose to configure DNS and load balancing rules at each location to resolve a single internal load balancing address only to the Management servers that are in the same location. This configuration allows you to obtain the best load distribution for your enterprise.

A typical approach, for best performance, is to keep configuration and analytics communications from Gateway servers to Management servers within the same geographical location by preference, while still retaining the ability to fail over to a different location.

Procedure

To configure IBM API Connect to use an internal load balancer, complete the following steps:

1. Decide on a single common host name to be used for the internal load balancer at each geographical location.
2. At each location, configure DNS and load balancing rules to resolve the internal load balancer host name to the Management servers at that location. Configure the load balancing rules to forward requests on ports 2443 and 9443, and to not terminate SSL connections at the internal load balancer.
3. Ensure that the firewall is configured to allow communication from the Gateway servers to the internal load balancer, and from the internal load balancer to the Management servers on the required ports; for more information, see Firewall requirements.
4. Configure IBM API Connect to use the internal load balancer by completing the following steps:
    a. In the Cloud Manager, click Services.
    b. In the Management Services pane, click the Service Settings icon ⚙.
    c. Select Use a different hostname for internal communication.
    d. Enter the internal load balancer host name in the Hostname for internal communication field, and click Save Service.
    The internal load balancer settings are propagated to each of the Gateway servers in the API Connect cloud.
    Note: It can take several minutes for the propagation of the settings to complete.
5. Verify your internal load balancer configuration by completing the following steps:
    a. In the Gateway Services pane, click the Server details icon
    ⓘ
    for each Gateway server, and confirm that the managementLoadBalancingAddresses and analyticsLoadBalancingAddresses properties are set to the internal load balancer host name.
    If you observe either of the following conditions, it is likely that the internal load balancer is incorrectly configured, or is not reachable:
        - After a period of time, the managementLoadBalancingAddresses list indicates "softdown" status for the internal load balancer address.
        - The analyticsLoadBalancingAddresses list contains the earlier addresses of each of the management servers in the cloud instead of the new internal load balancer address.
    b. Create, deploy, and publish a new API to each of the gateway servers, and confirm that the following conditions are satisfied:
        i. You can successfully call the new API on each Gateway server.
        ii. The new API calls are correctly reflected in the analytics information displayed in the API Manager UI.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

Configuring multiple Gateway services to have the same host name

If you have configured your API Connect cloud to use Dynamic DNS, you can give the same host name to two or more Gateway services. Use this capability to give the appearance that the APIs that are deployed to the separate Gateway services are on the same Gateway.

When your API Connect cloud is configured to use Dynamic DNS, API Connect prepends the Gateway service host name with the provider organization and catalog to form the fully qualified host name of the gateway address that is used for API calls. The format of the fully qualified host name is as follows:

```
api.provider_organization.catalog.gateway_service_host_name
```

Therefore, if you give the same host name to two or more Gateway services, the URLs that are used to call APIs that are deployed to those Gateway services differ only in the *provider_organization* and *catalog* components of the fully qualified host name.

The following table provides an illustrative example:

Table 1. Gateway services with the same host name

Gateway service name	Host name	Port	Provider organization	Catalog	Host name for API calls
Service 1	gateway.mybank.com	443	accounts	test	api.test.accounts.gateway.mybank.com
Service 2	gateway.mybank.com	443	accounts	production	api.production.accounts.gateway.mybank.com

You must ensure that your DNS configuration maps the fully qualified host names appropriately.

For information on configuring your cloud to use Dynamic DNS, see Selecting the DNS scheme.

For information on adding Gateway services to your cloud, see Configuring the initial Gateway service and Adding more Gateway services.

For information on provider organizations, see Administering provider organizations.

For information on Catalogs, see Working with Catalogs.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Specifying the cloud settings

Before you can add a provider organization to your cloud, you must first specify your cloud settings. A default setting is used for the DNS scheme and sandbox DataPower Service. However, you must specify the email server configuration and also review the user registry options. API Connect supports LDAP as an external IDP.

Before you begin

You must complete the following tasks:

- Configuring the Management service
- Configuring the initial Gateway service
- Adding a Gateway server

About this task

You can also integrate API Connect with LDAP and use it as the source to authenticate users to access the API Manager and the API Connect Developer Portal.

If you choose not to authenticate by using corporate LDAP, by default, the API Connect Internal Identity provider automatically provides authentication. In addition to LDAP, you can also use an authentication URL to authenticate users.

In the following task, you use the Settings page to specify your cloud settings.

Procedure

1. In the Cloud Manager, click Settings.
2. In the Email section, specify your email server connection details. For more information, see Connecting to an existing email server.
3. In the User Registries section, specify the user registry settings for managing access to the Cloud Manager and API Manager user interfaces. Note that in the Cloud Manager, the Cloud Administrator role is unique and the administrator can always access the Cloud

Manager, even if a Cloud Manager registry is changed. You can also define your own user registries, which you can make public and share with provider organizations. For steps on defining your own user registries, see Working with user registries.

   a. Optional: If you authenticate users with case-sensitive user names, move the case-sensitive usernames slider to the On position.
      Attention: This action cannot be undone. Ensure that you do not require case insensitive authentication credentials before setting.
   b. Choose a registry for the Cloud Manager and API Manager; select the required registry from the list. You can choose the local user registry (default) or if desired, you can select other registries.

4. In the TLS Profiles section, configure the following connections:
   a. In the Cloud Manager section, specify the port number and TLS profile for connecting to the Cloud Manager user interface.
   b. In the API Manager section, specify the port number and TLS profile for connecting to the API Manager user interface.
   c. In the Developer Portal APIs section, specify the port number and TLS profile for connecting to the Developer Portal site.
      Important: There is no support for presenting a client certificate when a call to /v1/portal/ is made.
5. V5.0.8 + In the Ciphers section, view the list of available ciphers for each TLS Protocol version. Enable or disable ciphers as needed.
6. In the Developer Portals section, move the Enable the developer portal slider to the On position to use the API Connect Developer Portal as the portal for any Catalogs that are created. Then, specify the connection details for the Developer Portal user interface. For more information, see Installing the Developer Portal..
7. In the DNS Scheme section, select the appropriate Dynamic DNS setting. For more information, see Selecting the DNS scheme.
8. In the Sandbox DataPower Service section, select the default DataPower gateway service for the development Catalog from the list.
9. V5.0.7 + In the Analytics section, configure destination targets for the analytics data that is captured for your on-premises cloud.
   For more information, see Configuring destination targets for API Connect analytics data.
10. V5.0.7 + Set the length of time that data is retained in the Analytics section.
    The default value is set for 90 days. Captured data that is older than the length of time that is specified here is deleted.
    Tip: Set this value as low as possible to avoid using extra disk space to maintain older data that you do not need.
11. V5.0.8 + Set the method by which the number of replicas is determined on your system.
    Replicas provide extra protection against loss of data if one of your nodes fails. The cluster can use the replicas to recompile the information. The following options are available:

    Dynamic
    : The number of replicas is automatically set as one less than the total number of nodes. If nodes are added or removed from the cluster, the number of replicas is automatically adjusted. For example, if you have three nodes in your cluster, selecting the Dynamic method sets your number of replicas to 2 (1 less than three total nodes).

    Fixed
    : The number of replicas is manually determined by the number that you enter in the Number of Replicas field. This number must be at least one fewer than the number of nodes in your cluster. If you add nodes to or remove nodes from your cluster, this number is not automatically updated. For example, if you have three nodes and set the number of replicas to 2, you must manually change this number to 1 if you remove one of the nodes.

12. V5.0.8 + Beginning with version 5.0.8.3, select Edit in the Analytics Fields section to identify the analytics fields that you want to be included with the analytics output.
    All of the available fields are selected by default, but you can deselect the fields that you do not want to capture to save storage space.
13. In the Advanced section, specify a configuration database failover timeout. Also configure Syslog to accept audit events from the Management node and write them to an external datastore.
    For more information, see Syslog configuration.
14. When you are finished, click Save in each section where you have made changes.

Results

Your cloud settings are specified.

What to do next

Complete the following task:

- Creating a provider organization account

- **Selecting the DNS scheme**
  Select the domain name server (DNS) scheme that you want to use in your API Connect on-premises cloud. The DNS scheme that you choose is used to define the endpoints within your cloud.
- **Connecting to an existing email server**
  Provide the details of an existing SMTP server and configure API Connect to generate emails, where required. You must configure the email server before you add any provider organizations. Otherwise, the owner of the organization does not receive the email with the details that they require to access the API Manager user interface.

Related tasks

- Updating a provider organization account
- Deleting a provider organization account

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Selecting the DNS scheme

Select the domain name server (DNS) scheme that you want to use in your API Connect on-premises cloud. The DNS scheme that you choose is used to define the endpoints within your cloud.

Before you begin

You must complete the following tasks:

- Configuring the Management service
- Configuring the initial Gateway service
- Adding a Gateway server

About this task

Important:

- If you modify the Developer Portal endpoint, it changes the address by which application developers access the portal page of your accounts. You must notify the users of the new address for your Developer Portal pages.
- To modify the inbound API calls endpoint, you must first remove all of the Gateway servers from the environment.

When you create your cloud, you must select the DNS scheme that you want to use for the Developer Portal and inbound API calls.

Procedure

To configure the endpoint settings, complete the following steps:

1. In the Cloud Manager, click Settings.
2. Select whether to use dynamic DNS.
   The URL format is used, by your users, to make API calls and also to access the Developer Portal.
   - Do not use Dynamic DNS. The URL format is as follows:

     ```
     https://<host>/<org>/<catalog>/...
     ```

   - Use Dynamic DNS. The URL formats are as follows:
     For API calls:

     ```
     https://api.<catalog>.<org>.<host>/...
     ```

     For accessing the Developer Portal:

     ```
     https://api.<catalog>.<org>.<host>/...
     ```

   Where:
   - *org* is the name of the provider organization.
   - *catalog* is the Catalog name. If one of the Catalogs has been configured to be the default, the *catalog* portion of the URL can be omitted for API calls, in which case the default Catalog is assumed; when accessing the Developer Portal, the *catalog* portion of the URL must always be included.
   - The value of *host* is as follows:

     For API calls
     : The address of the gateway service.

     For accessing the Developer Portal
     : The host name of the management server, or the management virtual IP host name if an external load balancer is used.

For dynamic DNS only, APIs in an organization use a subdomain of the domain name; the subdomain is named after the organization. In the following example, a default Catalog is being used and the Catalog name is omitted in the URL. For example, if the domain name given is `acme.example.com` and you have an organization name of sales, APIs in this organization use the domain name `api.sales.acme.example.com`.

The same concept applies when there are multiple Catalogs that are deployed for a single organization. The Catalog can be entered as another subdomain of the domain name; the subdomain is named after the Catalog too. Using the previous example, for the `Sales` organization and `Production` catalog (`prod`), the format of the URL is `api.sales.prod.acme.example.com`.

Your DNS must be configured with a wild card DNS mapping or an individual DNS entry for each organization. The DNS mapping for the example that is given is `api.*.acme.example.com` or individual entries, such as `api.sales.acme.example.com` and `api.marketing.acme.example.com`.

3. When you are finished, click Save in the Select DNS Scheme panel.

Results

The endpoint settings are configured.

What to do next

Complete the following task:

- Specifying the cloud settings

Related tasks

- Defining the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Connecting to an existing email server

Provide the details of an existing SMTP server and configure API Connect to generate emails, where required. You must configure the email server before you add any provider organizations. Otherwise, the owner of the organization does not receive the email with the details that they require to access the API Manager user interface.

Before you begin

You must complete the following tasks:

1. Configuring the Management service
2. Configuring the initial Gateway service
3. Adding a Gateway server
4. Specifying the cloud settings

About this task

Set up the details of your email server so that, when requested, emails can be sent to users; for example, when they are invited to join an API Connect organization.

Note: If your SMTP server requires SSL authentication, IBM API Connect uses the pre-supplied default SSL profile when connecting to the SMTP server, so you must upload the necessary trust store certificate into the default SSL profile. For details of uploading a trust store certificate, and of the certificate file formats supported by SSL profiles in IBM API Connect, see TLS profiles.
Restriction: Your email server might require you to authenticate before it can connect. Some email servers override either the sender address, sender descriptive name, or both, with the details of the authenticated user. Therefore, emails that are generated do not reflect the values that you configure in this panel.

Procedure

To connect to your email server, complete the following steps:

1. In the Cloud Manager, click Settings.
2. Navigate to the Email section.
3. In the Hostname field, enter the name of the host.
4. In the Port field, enter the number of the port.
5. Optional: If your email server requires you to authenticate, in the Username field, enter a user name to use for authorization with the SMTP server.
6. Optional: If your email server requires you to authenticate, in the Password field, enter the password for the entered user name. This password is used for authorization with the SMTP server.
7. In the Sender Address field, provide an email address for the sender of the email.
   Emails that are generated by yourAPI Connect cloud are sent from this address. Any replies to these emails are sent to this address. For example, you might decide to use the email address for the IT support department in your organization.
8. In the Sender Descriptive Name field, provide a human readable description of the email sender.
   For example, `IT Department Support`.
9. Optional: If you want to test that your settings are correct, click Test configuration, complete the details in the Test configuration window, and click Send.
   An email is sent to the email address that you specify.
10. When you are finished, click Save.

Results

Your SMTP server settings are configured.

What to do next

You can now complete the following task:

- Creating a provider organization account

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring your Gateway server extensions

You can add extra IBM® DataPower® enforcement capabilities to a Gateway service by uploading an IBM DataPower exported configuration .zip file.

About this task

Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), the capability described in this topic is **not** available. Depending on your company requirements, you can extend the behavior of the Gateway servers in a Gateway service to provide extra enforcement capabilities. For more information, see Extending the Gateway server behavior.

Procedure

1. In the Cloud Manager, click Services.
2. Navigate to the required Gateway service, and click the Service Settings icon ⚙.
   The DataPower Gateway Service window opens.
3. In the API Gateway Extension section, click Browse, if no extensions configuration file has yet been uploaded, or Replace this extension, if a file has been previously uploaded.
4. Select the IBM DataPower exported .zip file that you want to upload.

Note: The export.xml file created by DataPower in the .zip must have a <file> element that declares extensions.xml as part of the configuration. You must manually add this <file> element if it is not in the export.xml file.

5. V5.0.8 + **(From API Connect version 5.0.8.4 onward)** Use the Automatically refresh extension on gateway servers check box to specify whether you want the servers in the Gateway service to be automatically refreshed, or whether you want to refresh the servers manually. By refreshing the servers manually, you can determine the timing and order of the updates and can more easily coordinate the activity with an external load balancer, giving you more control over potential downtime of API runtime traffic on gateway servers.
   If you select the check box, server refresh is automatic. If you clear the check box, you must manually delete and re-add each server to refresh it; for details, see Gateway resynchronization. With the manual option selected, within several minutes of an update to the gateway extension, the status of gateway servers in this service will include an `Out of Sync` indicator until the servers are manually refreshed.

6. When you are finished, click Save.

Results

The extensions are 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.
V5.0.8 + **(From API Connect version 5.0.8.3 onward)** The gateway servers in the updated gateway service are refreshed based on the ascending order of the Address field of each gateway server. The Address (either hostname or IP address) of the gateway server was configured when adding it in Cloud Manager. The status of the server changes from REFRESH to ACTIVE as the extensions on each gateway server are applied.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing the cloud

Manage the servers in your cloud.

Following are the tasks for managing your cloud:

- **Testing Management servers**
  You can test the health of your Management server database by using the Command Line Interface.
- **Modifying servers**
  You can update your server configuration in the Cloud Manager.
- **Moving a Gateway server to another Gateway service**
  If you have created two or more Gateway services, you can move a Gateway server directly from one Gateway service to another, without having to first remove the server from the source service and then separately add it to the target service.
- **Gathering postmortem information about your servers**
  Server postmortem information is useful in problem determination.
- **Restarting and rebooting servers**
  You can restart or reboot servers to resolve operational errors.
- **Rebooting a Management cluster**
  To reboot a Management cluster, you must reboot each server individually, following a strict sequential procedure.
- **Removing and deleting servers**
  If you no longer require a server, you can remove or delete it.
- **Setting the maximum number of concurrent Gateway server additions**
  From IBM API Connect version 5.0.8.8 onward, you can set a limit on the maximum number of Gateway servers that can be added to a Gateway service concurrently.
- **Configuration database failover timeout**
  The configuration database failover timeout setting specifies how many seconds a secondary management server should wait before taking over as the primary when the primary server cannot be reached.
- **SNMP Items Available Using SNMP Get**
  Presents a table of OID trees that you can poll using SNMP Get.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

Testing Management servers

You can test the health of your Management server database by using the Command Line Interface.

About this task

Before you upgrade your Management servers to a new release, and periodically while they are running in your system, you should test your servers to ensure that they are running correctly. This test can be run by using the IBM® API Connect Command Line Interface.

Procedure

To test the health of your Management server database, complete the following steps:

1. Open a terminal window.
2. Log in to the Management server that you want to test with an account that has administrator privileges.
3. Enter the following command using the Command Line Interface: **`stat show apiconfig`**.
   The information that is returned provides statistics about the health of the database. This includes the following information:

   Internal check summary
   : Lists PASS/FAIL information from running Informix verification utilities. All checks must report a status of **`Pass`**.

   Memory segments
   : Shows the amount of RAM, both as a percentage and as an aggregate amount, that the Informix engine is using on the current server. The percentage is compared to its maximum allowed memory that is provided to the virtual machine. Review this to make sure that the Informix engine has enough available RAM.

   Database spaces
   : Shows how much disk space the Informix engine is using on the current server. Review this to ensure the database engine has enough available disk space. Note that only Primary servers use **`tempsblobsp`**, so its output is suppressed on other servers.

   Online server roles
   : Lists online servers, roles, and database transaction log replication state. Database transaction log replication state is represented by a combination of the LogID and LogPos values. LogID is an indication of which transaction log is currently being used; LogPos is the most recently used position within that log. If you watch these values over time LogPos grows to a preconfigured limit, then LogID is incremented by one and LogPos is reset to zero and begins growing again. Database content changes appear first on the primary servers, and then are copied to HDR and RSS servers. In a healthy configuration, the values for LogID/LogPos on the primary server are quickly reflected on the other servers.

   The following lines show an example of the output:

   ```
   dir/APIConnect> stat show apiconfig
   ```

   ```
   Reserved pages:   PASS
   Extent(apimdbs):  PASS
   Extent(sblobsp):  PASS
   ```

   ```
                     Total(MB)   Used(MB)   Free(MB)  %Used  %Free
   Memory Segments:   12000.00     770.25   11229.75   6.42  93.58
   ```

   ```
           apimdbs:    1000.00      35.67     964.33   3.57  96.43
           rootdbs:     292.96      90.37     202.59  30.85  69.15
           sblobsp:    4796.00     323.16    4472.84   6.74  93.26
          tempdbsp:     220.00       0.13     219.87   0.06  99.94
       tempsblobsp:      20.00       1.50      18.50   7.48  92.52
   ```

   ```
   Server              Serial Number         Role            LogID      LogPos
   192.168.0.1         VMWAA000A00KML00      HDR/AA            148         147
   192.168.0.2         VMWAQ1KB9UMARJFN      PRIMARY           148         147
   192.168.0.3         VMWF8GU0QYSTD6AK      RSS               148         147
   ```

If the checks fail, a line is added to the end of the output that warns you about the situation. The text is similar to the following message:

```
WARNING: Errors detected, see 'stat show apiconfig verbose'.
```

If you see that a secondary server has very different values than Primary, without sign of converging, this could indicate a synchronization problem.

Results

If the test results pass, then your Management server database appears to be running correctly.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Modifying servers

You can update your server configuration in the Cloud Manager.

Before you begin

Before you make any updates to the servers in the Cloud Manager (for example, adding or removing servers, or updating the host names), ensure that no other users are logged in to the Cloud Manager user interface, and that no users are logged in to the API Manager user interface. Otherwise, the users might experience unusual behavior.

About this task

If you change any of the following settings on a Management appliance or Gateway appliance in your IBM® API Connect cloud, you must reflect that change in the Cloud Manager user interface as soon as possible after updating the appliance to restore correct operation:

- Host name
- User name
- Password

In a cloud where there are two Management servers, the host name change breaks the connection between the two servers until you update the host name in the Cloud Manager user interface. Also, check that no users are connected to the user interfaces when you make this type of change to reduce the risk of loss of data, which might be written while the Management servers are disconnected from each other. When there are two or more Management servers in your topology, consider changing the host name of each appliance and update that host name in the Cloud Manager one at a time.

If you change the user name or password on a gateway server, communication is lost from the management servers to the gateway server until you update the username or password in the Cloud Manager user interface.

Procedure

To modify your server configuration, complete the following steps:

1. In the Cloud Manager, click Services.
2. Navigate to the server that you want to update and click the Server Actions icon ⋯, then click Edit server.
3. Change the configuration parameters and click Save server.

Results

The server configuration is updated.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

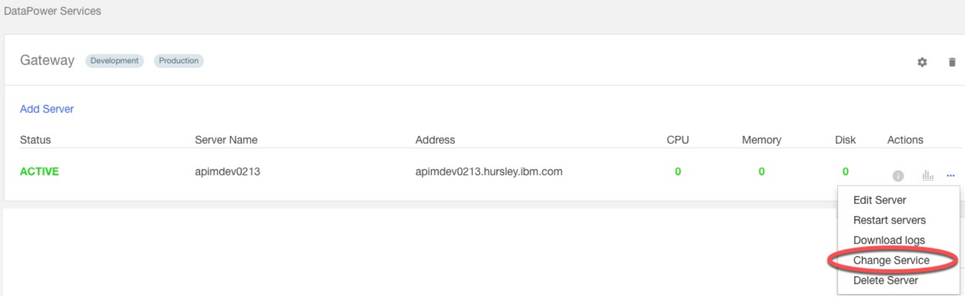
Moving a Gateway server to another Gateway service

If you have created two or more Gateway services, you can move a Gateway server directly from one Gateway service to another, without having to first remove the server from the source service and then separately add it to the target service.

Procedure

To move a Gateway server to another Gateway service, complete the following steps:

1. In the Cloud Manager, click Services.
2. In the DataPower Services section, locate the Gateway server that you want to move and click the Server Actions icon ⋯ , then click Change Service.
3. Select the name of the Gateway service to which you want to move the Gateway server, then click Change Service.



DataPower Services

Gateway Development Production

Status	Server Name	Address	CPU	Memory	Disk	Actions
ACTIVE	apimdev0213	apimdev0213.hursley.ibm.com	0	0	0	

Results

The Gateway server is moved to the target Gateway service.

Related concepts

- Configuring and managing your server environment

Related tasks

- Adding more Gateway services

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Gathering postmortem information about your servers

Server postmortem information is useful in problem determination.

About this task

If you contact IBM® Support, you are typically required to provide postmortem information.

Procedure

To gather postmortem information about your servers, complete the following steps:

1. In the Cloud Manager, click Services.
2. Navigate to the server for which you want to gather postmortem information, and click the Server Actions icon ⋯, then click Download logs.
3. Click Generate and Download.
4. If prompted, choose to save the compressed (.zip) file.
   The file might also be automatically downloaded depending on your browser.
   The file is saved to the download location that is configured for your browser, and is named apim-logs-*service*-*server*-*timestamp*.zip by default, where *service* represents the service type (`management` or `gateway` depending on the type of service that the server belongs to) and *server* represents the server name.
   You can extract the contents of the .zip file in order to review the postmortem files.

What to do next

If required, you can now restart or reboot your server, see Restarting and rebooting servers.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restarting and rebooting servers

You can restart or reboot servers to resolve operational errors.

Before you begin

If you are restarting or rebooting a server in response to operational errors, first gather the postmortem information; see Gathering postmortem information about your servers.

About this task

If you have an operational error, restarting or rebooting a server might solve the problem. Restarting a server closes all the processes that are running and starts them again. Rebooting a server closes all running processes and reboots the server. Rebooting a server is more intrusive than restarting a server and takes longer, and is typically used if restart does not resolve the issue.

Note: If you restart or reboot a Gateway server, the server is reloaded. Reload is a DataPower® term that means the same as restart.
Attention:

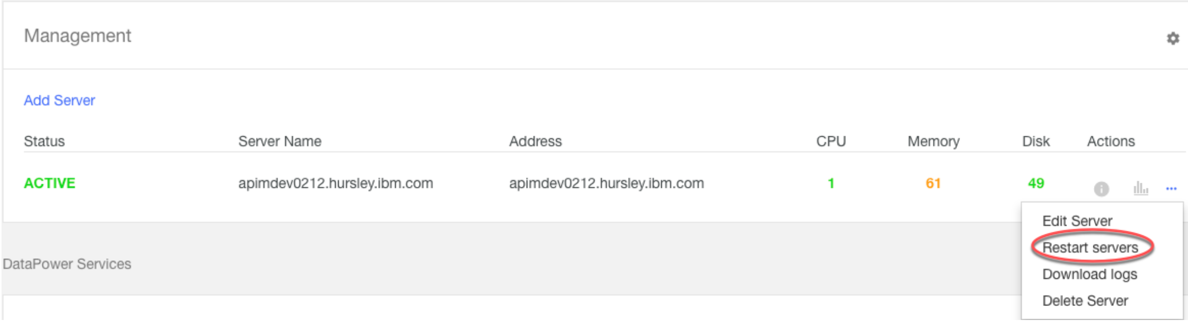
- If you configure two Management servers, for example for high availability or workload management, the persistent record of the defined APIs is replicated to both of the Management servers in case one of the servers fails. If one Management server fails or is stopped, when it restarts it uses the information, as provided by the server that remained active.
- If both Management servers are taken offline in one order and put back online in the opposite order, any changes that are made during that time period might be lost because two differing views of what represents the most recent configuration are contained in the persistent storage. Therefore, ensure that one Management server is always active.
- When you reboot two or more servers in a Management cluster, you must reboot each server individually, following a strict sequential procedure; for details, see Rebooting a Management cluster.

Restriction: You cannot restart or reboot a server if it is still joining or integrating with the API Connect on-premises cloud. Otherwise, the server might remain in an inconsistent state.

Procedure

To restart or reboot a server, complete the following steps:

1. In the Cloud Manager, click Services.
2. Navigate to the server that you want to restart and click the Server Actions icon ···, then click Restart Servers.



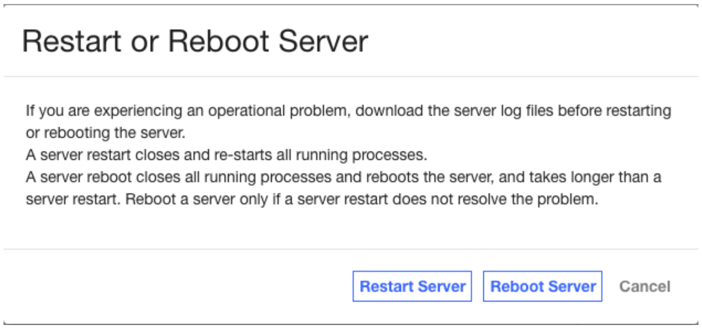
Management

Add Server

Status	Server Name	Address	CPU	Memory	Disk	Actions
ACTIVE	apimdev0212.hursley.ibm.com	apimdev0212.hursley.ibm.com	1	61	49	···

DataPower Services

The Restart/Reboot Server window opens.



Restart or Reboot Server

If you are experiencing an operational problem, download the server log files before restarting or rebooting the server.
A server restart closes and re-starts all running processes.
A server reboot closes all running processes and reboots the server, and takes longer than a server restart. Reboot a server only if a server restart does not resolve the problem.

3. To restart the server, click Restart Server. To reboot the server, click Reboot Server.

Results

The server is restarted or rebooted.

Related concepts

- Configuring and managing your server environment

Related tasks

- Gathering postmortem information about your servers

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Rebooting a Management cluster

To reboot a Management cluster, you must reboot each server individually, following a strict sequential procedure.

About this task

Important: When you reboot two or more servers in a Management cluster, you **must** ensure that the following conditions are met:

- No two servers in the cluster are rebooting at the same time.
- No two servers in the cluster are down at the same time.

You must therefore ensure that a rebooted server is fully recovered before proceeding to reboot another server, as described in the following procedure.

Procedure

1. Reboot one server in the cluster.
2. Log in to the Cloud Manager user interface on the rebooted server.
3. On the Analytics page, ensure that the CPU, memory, and disk usage data is displayed.
4. On the Services page, ensure that the status of the rebooted server is ACTIVE Note that it may take a few minutes before the server displays ACTIVE status after it is rebooted.
5. Ensure that you can successfully log in to the API Manager user interface on the rebooted server.
6. Repeat steps 1 to 5 for each server that you want to reboot.
   If any server fails to reboot successfully, open a support request.

Related tasks

- Restarting and rebooting servers
- Monitoring the health of your cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing and deleting servers

If you no longer require a server, you can remove or delete it.

Before you begin

You can delete a Gateway server or a Management server from your API Connect cloud completely. From API Connect version 5.0.8.8 onward however, for a Gateway server you also have the option to remove it from a Gateway service but retain it in your API Connect cloud.

Before you make any server updates (for example, adding or removing servers, or updating the host names), ensure that no other users are logged in to the Cloud Manager user interface, and that no users are logged in to the API Manager user interface. Otherwise, the users might experience unusual behavior.

About this task

To be valid, an API Connect cloud must contain at least one Management server and one Gateway server.

Note: You must always have at least one Management server, therefore you cannot remove the last Management server.

Procedure

To remove or delete a server, complete the following steps:

1. In the Cloud Manager, click Services.
2. Navigate to the server that you want to remove, and click the Server Actions icon ⋯.
   a. To delete the server from your API Connect cloud completely, click Delete Server.
   b. If the server is a Gateway server, you have the option to remove the server from the Gateway service but retain it in the API Connect cloud. Click Remove from Service.
      After removal, the server will be listed under Unused Servers, from where it can be added to a Gateway service in the future by using the Join Service option.
3. To remove or delete the server, click OK in the confirmation window.

Results

The server is removed or deleted from your API Connect cloud.

Important: Clean a deleted Management server by logging in to the command-line interface (CLI) of the deleted server, through a secure shell (SSH), and entering the **system clean apiconfig** command.
This command resets a deleted server to an empty state so that it can be reused in a new cloud if required. Alternatively, you might want to delete the appliance completely from your virtualization infrastructure. If the delete operation was unsuccessful, follow the instructions in the activity feed.

What to do next

If you want to reimage a deleted server that you removed, see Reverting servers to stand-alone appliances.

Related concepts

- Configuring and managing your server environment

Related tasks

- Adding a Gateway server
- Adding more Management servers
- Reverting servers to stand-alone appliances

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Setting the maximum number of concurrent Gateway server additions

From IBM® API Connect version 5.0.8.8 onward, you can set a limit on the maximum number of Gateway servers that can be added to a Gateway service concurrently.

About this task

By default, Gateway servers can be added to a Gateway service only one at a time. However, you can raise the value of the Concurrent Gateway Addition Limit setting to allow multiple servers to be added concurrently. In particular, this reduces the time taken to refresh Gateway servers after an upgrade.

Note:

- The Concurrent Gateway Addition Limit is a single cloud wide setting but applies to each Management server individually.
- Each concurrently added Gateway server must be a **unique** DataPower® appliance with a unique name. For more information about DataPower appliance names, see Defining information specific to the DataPower Gateway in the DataPower documentation.

The maximum number of total concurrent Gateway server additions across all Management servers should be the number of CPUs on a Management server.

If there are multiple Management servers in the Management cluster and, for example, each server has 8 CPUs, then you can choose to add Gateway servers in either of the following ways:

- 8 Gateway servers on a single Management server concurrently. In this case, set the Concurrent Gateway Addition Limit to 8.
- 4 Gateway servers on each of 2 Management servers concurrently; this option further improves performance. In this case, set the Concurrent Gateway Addition Limit to 4.

In either case, when adding the gateway servers you must log in to each Management server separately rather than using the Management server load balancer URL, because the Concurrent Gateway Addition Limit applies to each individual Management server.

Procedure

1. In the Cloud Manager, click Settings.
2. On the left hand side of the Settings page, click Advanced.
3. From the Concurrent Gateway Addition Limit list, select the required Gateway server limit.

4. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuration database failover timeout

The configuration database failover timeout setting specifies how many seconds a secondary management server should wait before taking over as the primary when the primary server cannot be reached.

About this task

Increasing the timeout value allows for more time to reestablish communication, reducing the possibility of the configuration becoming inconsistent with more than one server becoming primary. Increasing the failover timeout period may be necessary if the network connection between the management servers is not reliable or is prone to delays.
Note: Note: Change this advanced setting with caution and only if necessary.
For more information, see Dissociation and your cloud and Define the main site in your API Connect cloud.
Increasing the failover timeout reduces the high availability of the management servers in the cloud. (During the failover timeout period, the secondary management servers is not able to process updates if they cannot communicate with the primary server).

Procedure

To change the configuration database failover timeout setting, complete the following steps. The default failover timeout is set to 60 seconds. The timeout cannot be set to lower than 10 seconds. Typically, you would not set this failover timeout to higher than 300 seconds (5 minutes).

1. In the Cloud Manager, click Settings.
2. On the left hand side of the Settings page, click Advanced.
3. For the Configure Database Failover Timeout setting, enter the new failover timeout setting in seconds.
4. Click Save.

Results

The timeout setting is changed.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

SNMP Items Available Using SNMP Get

Presents a table of OID trees that you can poll using SNMP Get.

Note:

- 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.

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 table below 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

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	
.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 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	

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Monitoring the cloud

You can check the status of your servers, view detailed metrics about server usage, and see activities and alerts. V5.0.7+ You can also configure destination targets for the analytics data that is captured for your API Connect on-premises cloud.

About this task

The Cloud Manager shows the following types of monitoring information:

- Key metrics about CPU, memory, and disk usage for each cluster (shown on the Analytics page)
- Server availability (shown on the Services page)
- V5.0.5+ Specific server monitoring metrics about CPU, memory, and disk usage (shown on the Analytics page)
- System activity covering user actions in the Cloud Manager and lower-level server information and alerts (shown in the Notifications view)

You can monitor your API Connect cloud and configure destination targets for your analytics data by using the following tasks:

- **Monitoring the health of your cloud**
  You can monitor the following aspects of the health of your on-premises cloud: server availability, processor usage, memory usage, and disk space usage.
- **Viewing information about activities**
  You can use the Notifications view to track changes to the configuration or status of the cloud infrastructure, and changes to all user activity in the Cloud Manager; for example, the creation of server instances or provider organizations, the addition of servers to a cluster, or when users log in or out of the Management server.
- **Syslog auditing and your cloud**
  Syslog is a widely used standard for message logging. You can use syslog for system management and security auditing as well as general informational, analysis, and debugging messages. Devices such as printers and routers and message receivers across many platforms rely on the RFC 5424 syslog standard to enable consolidation of logging data from different types of systems in a central repository.
- **Syslog configuration**
  To support the retrieval of audit events from the management node, you configure Syslog to accept messages and write them to an external datastore.
- **Dissociation and your cloud**
  A properly functioning API Connect cloud has one management server acting as the *primary* server and additional servers acting as *secondary* or *RSS* servers. In a healthy cloud, the primary server is the only server able to write new data, while RSS servers must

forward all write requests to the primary server. Cloud dissociation occurs when two or more management servers independently decide to become primary servers and expect the others to become secondary. This divergence results in different sets of information being maintained by each server. In this topic, you learn how to resolve incidents of cloud dissociation and are presented with some common scenarios for recognizing incidents of dissociation.

- **Gateway resynchronization**
  If you make certain configuration changes to your IBM API Connect system, your Gateway servers might be displayed as out of sync on the Services tab, in the Cloud Manager.
- V5.0.8 + **Analytics split-brain**
  An Elasticsearch engine of API Connect analytics that is running correctly should have a single master, but sometimes a management cluster has multiple masters. This condition is called analytics split-brain. Multiple masters results in different log information being maintained by each server.
- V5.0.7 + **Configuring destination targets for API Connect analytics data**
  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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Monitoring the health of your cloud

You can monitor the following aspects of the health of your on-premises cloud: server availability, processor usage, memory usage, and disk space usage.

About this task

For each cluster, and for each server in a cluster, you can see the key metrics for CPU, memory, and disk usage.

V5.0.5 + You can also view the metrics for individual servers on an analytics dashboard.

V5.0.7 + 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 has been switched off. API Connect Analytics is switched off from the Settings tab in the Cloud Manager user interface. For more information, see Configuring destination targets for API Connect analytics data.
For more information about viewing key metrics, see:

- Monitoring the overall heath of your clusters
- Monitoring the health of the individual servers in your clusters

Monitoring the overall heath of your clusters

You monitor the metrics for CPU, memory, and disk usage for the cluster from the Analytics tab, which is the default tab you see when you first access the Cloud Manager.

About this task

The Analytics tab provides an analytics dashboard for the cluster, and displays the following metrics for the Management and Gateway services in the form of *visualizations* that are represented as charts:

- A line chart that collectively tracks the average processor usage, memory usage, and disk space usage of the cluster of Management servers, measured as a percentage
- A line chart that collectively tracks the average processor usage, memory usage, and disk space usage of the cluster of Gateway servers, measured as a percentage

The charts track the usage rates at three-hour time intervals (represented by the area between two consecutive data points), over the last seven days. When you hover over any line in the charts, a large black tooltip displays the usage percentage computed for that particular date and time. You can apply filters to change the time range of the data displayed and can set an auto-refresh rate for the data. You can also resize or rearrange the charts.

Note: Different data collection frequencies apply for the Management and Gateway services. On the Management node, the interval between successive data samples is 30 seconds, whereas on the Gateway node, the interval is five minutes. Consequently, denser graphs with more data points are produced for the Management service.

The analytics dashboard for the cluster is similar in layout to the analytics dashboards for Catalogs, and includes a subset of the analytics functions that are available for Catalogs in the API Manager. For a description of the layout of the screen elements, see The default Overview dashboard.

Procedure

To monitor the overall heath of your clusters and manipulate the data for further analysis, complete the following steps:

1. In the Cloud Manager, click Analytics if it is not already selected.
2. Review the charts and change the view of the data displayed if required.
   - Change the time period against which the data in the charts is scoped:
     - V5.0.7 + From the dashboard, in the Services Overview banner, click the Time Picker icon .
       V5.0.6 and earlier From the dashboard, in the Services Overview banner, click the Time Picker icon .
     - 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 above 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 in the same banner as the Time Picker icon.

     - If you want to additionally specify a frequency at which the data should automatically be refreshed in your charts, click Auto-refresh and then select a predefined refresh interval.
     - V5.0.7 + 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 charts are automatically refreshed to show the matching data. The specified quick, relative, or absolute filter setting is shown to the right of the Time Picker icon . If set, the auto-refresh interval is shown next to the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

       V5.0.6 and earlier Close the Time Picker by clicking the caret ^. The search query is resubmitted and the charts are automatically refreshed to show the matching data. The specified quick, relative, or absolute filter setting is shown to the right of the Time Picker icon . If set, the auto-refresh interval is shown next to the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

       Tip: V5.0.7 + 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.
       V5.0.6 and earlier To switch off the auto-refresh capability, click the auto-refresh value next to the Time Picker icon, and click Off.
   - Zoom in on a specific time range on the chart. This is equivalent to applying a filter for an "absolute" time period.
     - 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 both charts in the dashboard, and the start and end time range is shown to the right of the Time Picker icon (V5.0.7 + V5.0.6 and earlier).

     - To remove the filter, click the browser Back button. Alternatively, use the Time Picker icon to select the previous, or a different time range.
   - V5.0.6 and earlier Reposition the charts by using the container headers to drag and drop the charts in the preferred location in the dashboard.
   - Resize the charts by dragging outwards or downwards from the lower right corner to increase the size, or dragging inwards or upwards to decrease the size.

   Note: Changes you make are not permanently retained. When you move to another tab or close the session, the time filter defaults to Last 7 days again, and the charts revert to their original size and location.

Monitoring the health of the individual servers in your clusters

Metrics for individual servers are available on the Services tab.

Procedure

To monitor the health of the individual servers in your clusters and manipulate the data for further analysis, complete the following steps:

1. In the Cloud Manager, click the Services tab.
   The following information is displayed for each server in the Management services and Gateway services:
   - The status of the server
   - The server name and address
   - The maximum processor usage, memory usage, and disk usage for the server, measured as a percentage
2. V5.0.5+ To view and manipulate the analytics data for any of the servers, click the Analytics icon 📊 for that server.
   You are redirected to the Analytics tab, which displays an analytics dashboard for the server, with a line chart visualization that collectively tracks the average processor usage, memory usage, and disk space usage of the server, measured as a percentage. The chart tracks the usage rates at three-hour time intervals (represented by the area between two consecutive data points), over the last seven days. When you hover over any line in the chart, a large black tooltip displays the usage percentage computed for that particular date and time.

3. V5.0.5+ Review the chart and change the view of the data displayed if required.
   - Change the time period against which the data in the chart is scoped:
     - V5.0.7+ From the dashboard, click the Time Picker icon 🕘.
       V5.0.5 V5.0.6 From the dashboard, click the Time Picker icon 📅.
     - 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 above 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 in the same banner as the Time Picker icon.

     - If you want to additionally specify a frequency at which the data should automatically be refreshed in your chart, click Auto-refresh and then select a predefined refresh interval.
     - V5.0.7+ 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 chart is automatically refreshed to show the matching data. The specified quick, relative, or absolute filter setting is shown to the right of the Time Picker icon 🕘. If set, the auto-refresh interval is shown next to the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

       V5.0.5 V5.0.6 Close the Time Picker by clicking the Caret icon ^. The search query is resubmitted and the chart is automatically refreshed to show the matching data. The specified quick, relative, or absolute filter setting is shown to the right of the Time Picker icon 📅. If set, the auto-refresh interval is shown next to the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

       Tip: V5.0.7+ 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.
       V5.0.5 V5.0.6 To switch off the auto-refresh capability, click the auto-refresh value next to the Time Picker icon, and click Off.
   - Zoom in on a specific time range on the chart. This is equivalent to applying a filter for an "absolute" time period.
     - 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 the chart in the dashboard, and the start and end time range is shown to the right of the Time Picker icon (V5.0.7+ 🕘 V5.0.5 V5.0.6 📅).

     - To remove the filter, click the browser Back button. Alternatively, use the Time Picker icon to select the previous, or a different time range.
   - Resize the chart by dragging outwards or downwards from the lower-right corner to increase the size, or dragging inwards or upwards to decrease the size.

   Note: Changes you make are not permanently retained. When you move to another tab or close the session, the time filter defaults to Last 7 days again, and the chart reverts to its original size.

4. V5.0.5+ To close the analytics dashboard for a server, either refresh your browser tab or window to return to the original Analytics page in the Cloud Manager, or click one of the other tabs (for example, Services).

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing information about activities

You can use the Notifications view to track changes to the configuration or status of the cloud infrastructure, and changes to all user activity in the Cloud Manager; for example, the creation of server instances or provider organizations, the addition of servers to a cluster, or when users log in or out of the Management server.

About this task

You use the Notifications icon 🔔 to view notifications. This icon is displayed in the primary banner of every page in the Cloud Manager user interface.
V5.0.7+ Restriction: If the Notifications view displays no data or shows 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. For more information, see Configuring destination targets for API Connect analytics data.

Procedure

To view details of notifications, complete the following steps:

1. Click the Notifications icon 🔔.
   A log of activities is displayed in the Notifications view. On initial access, all activities are listed. On subsequent access, the level of detail shown might change based on the last set of activities accessed.
2. View the activities by category as follows:
   - To view all the configuration and status details of the cloud infrastructure, all user activity, and all alerts, click the Show all activity icon ☰. The activities are listed in reverse chronological order, with the most recent first.
   - To view all of the user activity, click the Show user activity icon 👤.
   - To view notifications of issues, click the Show alerts icon ⚠.
3. To check for new notifications while the Notifications page is open, click the Refresh icon ↻.
   The page refreshes to show the most recent activity.
4. When your review is complete, exit the Notifications view by clicking the Back button on your browser.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Syslog auditing and your cloud

Syslog is a widely used standard for message logging. You can use syslog for system management and security auditing as well as general informational, analysis, and debugging messages. Devices such as printers and routers and message receivers across many platforms rely on the RFC 5424 syslog standard to enable consolidation of logging data from different types of systems in a central repository.

Retrieving audit events with Syslog

To programmatically support the retrieval of audit events from the management node, the API Connect management node emits audit events as syslog messages. This allows a syslog collector to accept the messages and write them to an external datastore for further processing and/or archiving. The implementation supports both UDP and TCP-based transports, and you can protect the audit event data with SSL encryption. Note that audit events continue to be recorded in the analytics engine as well and are available to view on the Notification tab in the API Connect user interfaces.

The message format conforms to the syslog specification detailed in RFC 5424. It contains the following elements:

- Priority – Value will always be 110 to indicate a facility of 13 (audit) and a severity of 6 (informational).
- Version – The syslog specification version. Always set to a value of 1.
- Timestamp – Specifies when the event took place in ISO8859-1 format. The time is accurate to a millisecond and indicates the completion of the activity being logged.
- Hostname – Hostname or IP address of the management node.
- App Name – This will always be “ibmapimanagement”.
- Proceed – Contains the dash character ('-') to indicate a nil value.
- MsgId – Contains up to 32 characters of the provider or developer organization name for messages that originate from a provider or developer organization, and contains the constant string *cmc* for messages that originate from the Cloud Manager. Any spaces in the organization name are replaced by the underscore character ("_"). If the origin cannot be determined, the value of the MsgId is the dash character ('-') to indicate a nil value.
- Structured Data – The structured data field is not used and the value is always the dash character ('-') to indicate a nil value.
- Message Text – Describes the event that took place and includes information relevant to the event, for example. API name, user name, etc.

The following example shows a message representing the creation of an API:

```
<110>1 2017-10-21T10:44:20.529Z apimsampledev.ibm.com ibmapimanagement - macsshack - API myDemoApi
version 1.0.0 was created from a Swagger document by username@example.com.
```

The syslog configuration is performed in the Settings page of the Cloud Manager. To configure a connection to a syslog collector, select the type of connection (UDP or TCP) and enter the hostname or IP address of the syslog collector. The default ports are 514 for UDP and 601 for TCP. Optionally, you can specify a port other than the default port if necessary.

Only one syslog collector can be specified per cluster. All management nodes on the cluster connect to the same syslog collector.

The connection to the syslog collector is dynamic. To change the syslog collector, simply change the hostname or IP address of the collector, then save the settings. You can also switch between a UDP transport and a TCP transport dynamically using the same mechanism.

If syslog messages are not being received in the collector after configuration is complete, the logs for the servers can be downloaded and file /var/log/cmc.out examined for messages related to the connection state. The file indicates a successful connection and contains messages for connection failures. These messages can be used to validate that the connection was successful or to further diagnose issues for unsuccessful connections.

For details on configuring IBM® API Connect to log Syslog auditing messages, see Syslog configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Syslog configuration

To support the retrieval of audit events from the management node, you configure Syslog to accept messages and write them to an external datastore.

About this task

In this task, you learn how to configure Syslog using the Settings page of the Cloud Manager.

For details on the structure of an audit message, see Syslog auditing and your cloud.

Procedure

1. In the Cloud Manager, click Settings.
2. Click **Advanced**.
3. In the Audit log section, enter the Host name or IP address of the Syslog collector.
   Note: Because all management nodes on the cluster connect to the same Syslog collector, only one collector can be specified per cluster.
4. In the Port text field, enter the Port number.
5. For Protocol, select either NONE, UDP or TCP. Default ports are 514 for UDP and 601 for TCP.
   Note: The connection to the Syslog collector is dynamic. To change the Syslog collector to connect to, simply change the hostname or IP address of the collector. You can also switch between a UDP transport and a TCP transport dynamically using the same

mechanism.
6. In the SSL Profiles section, select a profile defined in the SSL Profiles page of the CMC. The selected profile communicates with the audit log host.
7. Click **Save**.
   Note: If Syslog messages are not being received in the collector after configuration is complete, download the server logs and examine the /var/log/cmc.out file for messages related to the connection state. The file indicates a successful connection or contain messages for failures in the connection process and you can use these to validate that the connection was successful or further diagnose issues.

Results

Syslog is configured to accept messages and write them to an external datastore.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Dissociation and your cloud

A properly functioning API Connect cloud has one management server acting as the *primary* server and additional servers acting as *secondary* or *RSS* servers. In a healthy cloud, the primary server is the only server able to write new data, while RSS servers must forward all write requests to the primary server. Cloud dissociation occurs when two or more management servers independently decide to become primary servers and expect the others to become secondary. This divergence results in different sets of information being maintained by each server. In this topic, you learn how to resolve incidents of cloud dissociation and are presented with some common scenarios for recognizing incidents of dissociation.

Detecting cloud dissociation

Cloud dissociation can occur when network communication between servers is severed. As a result, a group of servers with no primary server selects a new primary and begins to report to it, triggering the dissociation. The servers in each group then report servers from other groups as inactive or with an empty role. Once the network outage is resolved, an alert is displayed in the Management clusters section of the Clusters page. A dissociation icon is displayed adjacent to the dissociated servers and email notification of the dissociation is sent to users with the following roles: Cloud Owner, Cloud Administrator and Topology Administrator.

V5.0.8 + Beginning with version 5.0.8.3, you can also determine whether you have a cloud dissociation situation by using the health check REST APIs. See Obtain health check data of Management servers by using REST API calls for more information about this method.

Reducing the likelihood of cloud dissociation

You can reduce the likelihood of cloud dissociation due to an interruption in network communication in the following ways:

- Define a main site in your API Connect cloud
- Increase the database failover timeout

Recovering from cloud dissociation

In the following section, you are provided with steps for recovering from cloud dissociation. However, before you begin, be sure to:

- Back up all management servers to avoid any possible mistakes. To do this, log directly into the Cloud Manager of each dissociated server and determine which server (or servers) are designated as primary. For more information on backing up your servers, see Creating a backup of an API Connect configuration.
- Back up any important Products and APIs from the respective dissociated primary servers. This guards against choosing the wrong server and erasing important data.
- Choose the server you want to retain as the management server. To choose your management server, check your Cloud Manager settings and log into API Manager to compare your APIs and Products.

Perform the following procedures to resolve cloud dissociation. In the following steps, *M1* and *M2* represent the servers to be retained as the primary and secondary (RSS) servers with the "good" data to retain. *M3* and *M4* are dissociated servers to be removed and re-added. In all steps, the Cloud Manager is represented by CMC.

1. Back up the configuration from all management servers:

    a. On the command line interface (CLI) of each management server (M1, M2, M3 and M4), execute `config save apiconfig` to capture the configuration backups from each server.
    b. On dissociated servers M3 and M4, export or back up any Products, APIs or other artifacts not reflected in the "good" servers (M1 and M2) that you want to retain.
2. Delete all dissociated management servers, checking after each removal to ensure that indices are intact:
   By default, Analytics is configured to use one primary shard containing the data for each Analytics index, and 1 replica shard containing the same data (described in step 11 of Specifying the cloud settings). These shards are distributed among the nodes in your cluster such that for each index, one node contains the primary shard and another node contains the replica. When you have more than 2 nodes in the cluster, removing multiple servers from the cluster in quick succession can result in removing the servers that contained both the primary shard and the replica shard for a particular index--which results in the index reporting as red, due to the loss of data.

   To avoid data loss, you should check the indices before deleting any servers from the cluster, and again after each removal.

    a. Run the **`stat show analytics indices`** command as a baseline to understand the current state of the indices.
       For information on the **`stat show analytics indices`** command, see **`stat show analytics`** commands.

    b. Delete a dissociated server from the cluster.
       For example, to delete server M3:

        i. Log into the M1 server's Cloud Manager and remove the M3 server.
        ii. From the CLI of M3, monitor the log using the debug tail file /var/log/cmc.out command to confirm that M3 cleans itself and fully restarts. This may take 5-10 minutes.
        iii. Log into the M3 server's Cloud Manager UI (with default admin password *!n0r1t5@C*) to ensure that M3 is displayed as the only management server in the Management Cluster section of the Clusters page.
        iv. If M3 did not clean itself and you are still able to log into M3's Cloud Manager with your old password *or* if M3's Cloud Manager lists more than one management server, execute system clean apiconfig on the M3 server's CLI. After the server restarts, log into M3's Cloud Manager to ensure M3 is displayed as the only management server in the Management Cluster section of the Clusters page.
    c. Check the health of the indices for that cluster by running the **`stat show analytics indices`** command again.
       A status of green indicates for an index means that all shards are assigned. If an index has a health of yellow, there is at least one unassigned shard for that index.

    d. Before you delete any more dissociated servers, wait until any previously unassigned shards have been assigned to a server that is still associated with the cluster and no indices show a status of red.
    e. Repeat this process to delete each of the remaining dissociated servers (for example, to delete server M4).
3. Add the fully cleaned management servers back:
    a. In the M1 server's CMC, add the M3 server, then wait for M3 to restart and synchronize. Following this, both the M1 and M3 servers should be active and the Management Cluster section of the Clusters page should show M1 as Primary and M3 as RSS.
    b. Confirm that both M1 and M3 servers are fully operational and are populated with the expected configuration. For example, log into the Cloud Manager of both M1 and M3 directly, expand the details of all the management servers and ensure that the role displayed for each server matches both perspectives. Next, log into the API Manager on M3 to confirm that the information displayed there matches M1's API Manager.
    c. Repeat the addition of any remaining formerly dissociated (now clean) management servers. For example, re-add M4 and confirm its proper addition to the cloud.
4. Apply individual changes from formerly dissociated servers:
    a. If the information on M1/M2 and M3/M4 servers diverged during the cloud dissociation, apply any changes to the M1 server that were made to M3 and/or M4 and not reflected on M1 due to cloud dissociation. For example, in the API Manager of M1, import the earlier exported Products or APIs from M3 or M4 that may have been added when those servers were in a dissociated state.

Related concepts

- Configuring and managing your server environment
- Managing the cloud

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Gateway resynchronization

If you make certain configuration changes to your IBM® API Connect system, your Gateway servers might be displayed as out of sync on the Services tab, in the Cloud Manager.

About this task

Your Gateway servers can display that they are out of sync due to the following reasons:

- You have updated the TLS profile that is associated with a Gateway service containing the servers that are now out of sync. Updates to TLS profiles are not automatically propagated to Gateway servers.
- You have upgraded your management servers to a new version.
- V5.0.8 + **(From API Connect version 5.0.8.4 onward)** A gateway extension for the Gateway service that contains the server has been added, modified, or deleted with the Automatically refresh extension on gateway servers check box explicitly unchecked to request manual refresh. For more information, see Configuring your Gateway server extensions.

Procedure

To resynchronize any of your out of sync Gateway servers, complete the following steps:

1. Delete an out of sync server from your Gateway service. For more information, see Removing servers.
2. Add the server back to your Gateway service. For more information, see Adding a server.
3. Repeat steps 1 and 2 for any other servers that are out of sync.
   You can avoid downtime for API requests by deleting and re-adding back each Gateway server individually, instead of all at the same time. This allows the remaining Gateway servers to service API requests.

Results

By removing and adding the Gateway servers back to the Gateway service, the Cloud Manager no longer displays that the Gateway servers are out of sync as they are using the latest configurations.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Analytics split-brain

An Elasticsearch engine of API Connect analytics that is running correctly should have a single master, but sometimes a management cluster has multiple masters. This condition is called analytics split-brain. Multiple masters results in different log information being maintained by each server.

Identifying analytics split-brain

Analytics split-brain occurs when there is a temporary network disruption, and now there is more than one master server for the analytics data. There are three ways to identify analytics split-brain:

Check to see if the analytics data reporting rate decreased on the master server

In an analytics split-brain condition, the analytics data that is normally indexed on a single Elasticsearch cluster is indexed across multiple clusters. Because the analytics data is indexed on different clusters, one of the first indicators of the split-brain condition is a significant decrease in the rate of analytics data on what was originally the only cluster.

See if you received an email notification from API Connect

Approximately 15 minutes after the network disruption is resolved and the analytics split-brain condition begins, API Connect sends an email notification to the cloud administrator, cloud owner, and topology administrator of the affected API Connect management node. The email contains the following information:

- The management node where the split-brain condition was detected, including information about the URLs for which servers need to be restarted.
- The time when the condition was first detected.
- A link to this topic to provide instructions for resolving the issue.

If you receive email notifications about other issues that the network disruption caused, resolve those issues before resolving the analytics split-brain issue. Recovering one of those issues might contain steps that also resolve the analytics split-brain condition. After the initial notification email, two reminder emails are sent out per day until the condition is resolved.

Invoke the REST API or run the command to view details about the nodes

- You can identify analytics split-brain by invoking the `get _cat node` REST API on each of the servers to get detailed statistics about each node.
- You can identify analytics split-brain by entering the `stat show analytics nodes` in the command-line interface. This also returns the details about the nodes.

If you notice that there are a fewer number of nodes than there are the number of cluster members, this might be a sign of a split-brain condition. You can confirm split-brain when you see two or more unique master nodes running in the cluster. On a healthy cluster, there is only one master.

Resolving the analytics split-brain condition

During the analytics split-brain condition, the unique analytics data is sent to different Elasticsearch masters. You cannot fully merge the unique information from the multiple masters together during the recovery. This causes you to lose data that was written to all of the masters during the split-brain state, except the one that you select to continue using as the master. The data that is on the selected master becomes the basis for all of the analytics data in the future. The faster you resolve the analytics split-brain condition, the less analytics data is lost.

To resolve the condition, it is important to determine which nodes to restart. The analytics Elasticsearch cluster membership must match the number of management cluster members, and it generally minimizes the analytics data loss when you restart the fewest number of nodes. When you restart a system, avoid restarting the primary node. If you need to restart multiple nodes, restart them as soon as you can after one another, starting with secondary master. See the example below for more details.

After you determine that your system is in an analytics split-brain condition, complete one of the following procedures to resolve it:

Restart the management server

An analytics split-brain condition is automatically resolved when you restart the management node to fix another issue that occurred as a result of the network disruption, such as cloud dissociation. If you restart the management node, no additional action is required.

Use the REST API to restart the Elasticsearch server

If you do not want to restart your management server, you can restart only the Elasticsearch server by invoking the REST API to restart by completing the following steps:

1. Use the analytics Elasticsearch REST API to restart just the Elasticsearch process on the management node. This requires you to have the cloud administrator, cloud owner, or the topology administrator role. If you do not specify an IP query parameter, the node that you are connected is restarted. enter the following `curl` command to restart the server:

   ```
   curl -XPOST https://mgmt_server_hostname/v1/analytics_ops/es_restart?ip=ip_address -u
   cmc/user:password
   ```

   Where:
   - *mgmt_server_hostname* is the host name and domain of your server.
   - *ip_address* is the IP address of the server that you are restarting.
   - *user* is the Cloud Manager user name that has cloud administrator, cloud owner, or topology administrator permissions assigned to it.
   - *password* is the password for the administrator account for that server.

   Remember: You can use the email notification that you received about the analytics split-brain condition to identify the URL paths to the nodes that you need to restart.
2. Repeat step 1 for any other servers that you need to restart.

To verify that your analytics configuration is working correctly after the recovery, complete the following procedure:

1. Confirm that new analytics data is being stored.
   a. Confirm that new analytics data is flowing into the Cloud Manager analytics views.
      Note: Monitoring event storage must be enabled for API Connect for new analytics data to be stored and to be displayed in the Cloud Manager analytics views.
   b. Confirm that new analytics data is flowing into the API Manager analytics views.
      Note: API event storage must be enabled for API Connect for new analytics data to be stored and to be displayed in the API Manager analytics views.
2. Use the REST API or the command-line interface to confirm that the analytics subsystem is healthy.
   a. On each server, either call the `get _cat node` REST API, or enter `stat show analytics nodes` in the command-line interface, and ensure that each server reports the same number of nodes as there are cluster members, and that each server reports the same master.
   b. On each server, call the `_cluster/health` REST API, and check that the `status` property does not have the value `red`.

Sample analytics split-brain notification email

The following text contains a sample of the email notification that you receive when you have an analytics split-brain condition on your system:

```
Manager Server: 9.20.153.94
  Master: 9.20.153.94
  Nodes in the cluster:  9.20.153.94 9.20.153.96
  Elasticsearch rest API restart URL (use POST request):  /v1/analytics_ops/es_restart?ip=9.20.153.94
Manager Server: 9.20.153.95
  Master: 9.20.153.95
  Nodes in the cluster:  9.20.153.95  9.20.153.97 9.20.153.98
  Elasticsearch rest API restart URL (use POST request):  /v1/analytics_ops/es_restart?ip=9.20.153.95
Manager Server: 9.20.153.96
  Master: 9.20.153.94
  Nodes in the cluster:  9.20.153.94 9.20.153.96
  Elasticsearch rest API restart URL (use POST request):  /v1/analytics_ops/es_restart?ip=9.20.153.96
Manager Server: 9.20.153.97
  Master: 9.20.153.95
  Nodes in the cluster:  9.20.153.95  9.20.153.97 9.20.153.98
  Elasticsearch rest API restart URL (use POST request):  /v1/analytics_ops/es_restart?ip=9.20.153.97
Manager Server: 9.20.153.98
  Master: 9.20.153.94
  Nodes in the cluster:  9.20.153.95  9.20.153.97 9.20.153.98
  Elasticsearch REST API restart URL (use POST request):  /v1/analytics_ops/es_restart?ip=9.20.153.98
```

In this example, you can see the following five nodes in the management cluster: 9.20.153.94, 9.20.153.96, and 9.20.153.98. You can see that both 9.20.153.94 and 9.20.153.95 are listed as masters. This scenario shows an analytics split-brain condition. In this example, complete the following steps to resolve this condition with 9.20.153.95 as the master:

1. Restart 9.20.153.94. This resolves the issue of having dual masters. When it restarts, it is no longer identified as a master node.
2. Restart the following nodes in any order:
   - 9.20.153.96
   - 9.20.153.98

   Because these Elasticsearch nodes recognized 9.20.153.94 as the master, you must restart them so they can be reconfigured with the correct master.

Related concepts

- Configuring and managing your server environment
- Managing the cloud

Related information

- Obtaining cluster health information by using REST API calls

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Configuring destination targets for API Connect analytics data

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

Event data might typically 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 can enable users to draw correlations with other data sources on a single platform, and help 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.

For a number of API Connect event types, you can configure a combination of destination systems to which data is forwarded for analysis and display. For each event type, you can choose one of the following options:

- Enable access to analytics data in the API Connect user interfaces. This is the default setting for each API Connect event type.
- 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.
- Completely disable access to analytics data.

Use the following information to configure destination targets for your analytics data:

- V5.0.7+ **Supported event types for analytics offload**
  Your API Connect offering is configured to emit events that provide API invocation metrics, performance monitoring metrics for the on-premises cloud, and alerts and notifications of system and user activities. When these events are generated, the event data is captured, and by default is typically presented for analysis or as notifications within the user interfaces. The captured event data can also be offloaded to third-party systems if required.
- V5.0.7+ **Supported third-party systems for analytics offload**
  You can choose to offload the analytics data for one or more API Connect event types to a number of third-party systems. The supported systems include HTTP servers, Elasticsearch clusters, Kafka clusters, and Syslog servers.
- V5.0.7+ **Enabling or disabling access to analytics event data in API Connect**
  The Gateway logs and reports all API interactions to the API Connect analytics engine, for real-time and historical analytics and reporting. Performance-related data about the Management and Gateway servers, and other data relating to system and user status and activities are also captured for display and analysis. By default, this analytics data is accessible from the API Connect user interfaces. You can, if preferred, switch off access to analytics within the API Connect user interfaces.
- V5.0.7+ **Configuring the offload of analytics event data to third-party systems**
  You can forward the analytics data that is captured for API Connect events to a number of third-party systems as a real-time data stream. This capability is useful if you want to consolidate data from multiple sources, if you require enhanced monitoring, or if you want to enrich your analytics data. Target systems to which you can offload data include HTTP, Elasticsearch, Kafka, and Syslog servers.
- V5.0.7+ **Sample test events for analytics offload**
  When you configure the offload of analytics data to third-party systems, you can use the Send Test Event button to verify that your configuration settings are valid before you save your settings. This button generates and sends a test event to the target system.
- V5.0.7+ **Troubleshooting your analytics offload**
  Review this guidance to help you resolve issues if your data offload fails. Common causes might be connection or security-related.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Supported event types for analytics offload

Your API Connect offering is configured to emit events that provide API invocation metrics, performance monitoring metrics for the on-premises cloud, and alerts and notifications of system and user activities. When these events are generated, the event data is captured, and by default is typically presented for analysis or as notifications within the user interfaces. The captured event data can also be offloaded to third-party systems if required.

The following event types can be configured for display in API Connect or can be offloaded to supported third-party systems:

- API events
- Monitoring events
- Log events
- Audit events

API events

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. For more information about the fields that are displayed in an API event record, see API event record fields. 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:

- Within analytics dashboards in the API Manager user interface.
  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 Catalog dashboards, see API Analytics. 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. For more information about accessing the debug function, see Testing an API with the API Manager test tool.

- By using API Manager REST API calls.
  The REST API calls return JSON objects that contain API invocation metrics for a Catalog and API provider organization. These JSON objects contain the same information as the API event records that are displayed when the API Manager user interface is used to view the raw data for visualizations. For information about issuing REST API calls, see Obtaining analytics data by using REST API calls.

By default, API consumers who are members of a developer 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 developer 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:

- When you disable access to analytics data within API Connect, data streaming from the Gateway to the UIs is immediately terminated. API providers and consumers will still be able to view the analytics dashboards in the API Manager and Developer Portal UIs. However, the visualizations will either display no data, or might show data only for a time period in the past, depending on when Analytics was disabled.
- When you disable access to analytics data within API Connect, no API data will be displayed when API developers use the debug function while testing their APIs locally in the API Manager user interface.
- Disabling access to analytics data within API Connect has no effect on the REST API. Calls can still be issued to return API invocation metrics for a Catalog and API provider organization.

Monitoring events

Monitoring events provide performance-related data about your Management and Gateway servers, and enable you to monitor the health of your on-premises cloud.

By default, API providers with the relevant access permission can view key metrics about processor usage, memory usage, and disk space usage for each Management and Gateway cluster of servers, and for each server in a cluster, within an analytics dashboard in the Cloud Manager user interface. For more information, see Monitoring the health of your cloud.

Effect of switching off IBM API Connect Analytics for monitoring events: When you disable access to analytics data within API Connect, data streaming of performance-related data to the Cloud Manager UI is immediately terminated. Users with relevant access to the Cloud Manager will still be able to view the analytics dashboards for the Management and Gateway services and servers. However, the visualizations will either display no data, or might show data only for a time period in the past, depending on when Analytics was disabled.

Log events

The Gateway can be configured to forward log events to the Management server for storage. These events are not currently viewable in any of the API Connect user interfaces. (If required, the events can be viewed by using the Gateway's log event viewer, or by using the offload mechanism to route them to a third-party system for display and analysis.)

Effect of switching off IBM API Connect Analytics for log events: When you disable access to the analytics data for log events within API Connect, these events will no longer be stored in the Management server, but users will observe no discernible difference since these events are not visible in the UIs.

Audit events

Audit events provide notifications of any changes to the configuration or status of the cloud infrastructure, and also provide details about API developer user activity, API activity, and alerts in the Management server. For example, notifications can be generated when server instances, provider organizations, or users are created, when servers are added to a cluster, when users log in or out of the Management server, or when Products are published.

By default, API providers can view audit events within the Notifications view in the Cloud Manager and API Manager user interfaces. The Notifications view can be accessed by clicking the Notifications icon 🔔, which is displayed in the primary banner of every page in the API

Manager and Cloud Manager UIs. For more information about using the Notifications view, see Cloud Manager: Viewing information about activities and API Manager: Viewing notifications of activities and alerts.

Effect of switching off IBM API Connect Analytics for audit events: When you disable access to analytics data within API Connect, streaming of notifications data to the Cloud Manager and API Manager UIs is immediately terminated. Users will still be able to view the Notifications view in the UIs. However, this view might either display no notifications, or might show notifications only for a time period in the past, depending on when Analytics was disabled.

Related concepts

- Supported third-party systems for analytics offload

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Supported third-party systems for analytics offload

You can choose to offload the analytics data for one or more API Connect event types to a number of third-party systems. The supported systems include HTTP servers, Elasticsearch clusters, Kafka clusters, and Syslog servers.

For details about offloading data to these systems, see Configuring the offload of analytics event data to third-party systems.

- HTTP
- Elasticsearch (Version 5)
- Apache Kafka (Versions 0.9 and 0.10)
- 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. If preferred, you can configure your data offload to forward the real-time event stream to other systems that provide enhanced search, analytics, and visualization capabilities. Note that if the offload HTTP target is unreachable, the events will be lost.

For example, you can forward your API Connect event data to a Splunk deployment, which is set up to receive, index, search, analyze, and visualize IT streaming, machine, and historical data that is collected from diverse websites, applications, sensors, and devices. Or, you can forward your API Connect event data to an in-memory data structure store like Redis, which is set up for use as a database, cache, and message broker.

The following steps provide an example of how you can configure your API Connect HTTP output to offload event data to Splunk's HTTP Event Collector. The HTTP Event Collector provides an endpoint in your Splunk deployment to which you can forward event data using HTTP or HTTPS.

1. From your Splunk deployment, complete the following steps:
    a. Set up the HTTP Event Collector for use by enabling the HTTP Event Collector and ensuring that an Event Collector token has been created and enabled for use with API Connect. Event Collector tokens are generated as globally unique identifiers (GUIDs). Depending on your infrastructure setup and role assignments, you might need to work with a designated Splunk administrator to configure the HTTP Event Collector and obtain a token. For more information, see Set up and use HTTP Event Collector and HTTP Event Collector token management in the Splunk documentation.
    Tip: The API Connect Management server will need to use this Event Collector token to authenticate to the Splunk server on which the HTTP Event Collector is running. When configuring the data offload, you can insert the token in the authorization header of each HTTP request, as described in step 2.
    b. Generate a unique channel ID (as a GUID) by using your preferred method.
    Tip: This ID will be used to automatically create a channel for the API Connect event data that is sent to the HTTP Event Collector, and is required because API Connect transmits raw events. When configuring the data offload, you can include the channel ID with each HTTP request that contains raw events by specifying a channel identifier header, as described in step 2. For more information, see Format events for HTTP Event Collector and Raw event parsing in the Splunk documentation.
2. From the Cloud Manager user interface, configure the data offload for HTTP, for your required event type. For more information, see Configuring the offload of analytics event data to third-party systems.

a. When you specify the HTTP server URL, include the Splunk "raw" endpoint (`services/collector/raw`) that enables one or more raw events to be sent in a single request. For example:

```
https://server_host_name:port/services/collector/raw
```

By default, the "raw" endpoint uses HTTPS and listens on port 8088.

b. Add the following custom HTTP headers and corresponding values, which will enable the Management server to connect to the HTTP Event Collector and forward raw events:

Table 1. HTTP header configuration for connection to Splunk

Header name	Header value	Description

`Authorization`	`Splunk` *`event_collector_token`*
	Example:
	`Splunk 4DC8896A-6F9E-47A4-9DA6-A3FEF9F2A1A3`
		Use an authorization header to specify a valid Event Collector token that can be used by API Connect. (See step 1 for details.)

`x-splunk-request-channel`	*`GUID`*
	Example:
	`19674C68-B28B-4550-9CF0-6E7345CA60CA`
		Use a channel header to specify a unique and valid channel ID, which ensures that the API Connect data forwarded to the HTTP Event Collector is kept separate from any other event sources that are configured in the same Splunk instance.

Elasticsearch (Version 5)

You can offload API Connect analytics data to an Elasticsearch target system that is used for real-time distributed search and analysis of data. The target system can be an Elasticsearch cluster that comprises a single running instance (or node) on a server, or a collection of nodes that typically run on individual servers and work together to share their data and workload.

The data that you offload will be stored as schema-free JSON documents in one or more indexes that are created using an Elasticsearch template that API Connect supplies, and which are distributed across a number of shards hosted on the cluster nodes. The naming syntax that you specify for your indexes determines how many indexes are created and how they are characterized. For example, you can define indexes that are created based on a date pattern (for example, daily or weekly) or indexes that are created to store data by provider organization or by API name.

Apache Kafka (Versions 0.9 and 0.10)

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.

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 the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP)-based transports. The Syslog server might 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.

Related concepts

- Supported event types for analytics offload

Related information

- Splunk
- Redis
- Elasticsearch
- Kafka

V5.0.7+

Enabling or disabling access to analytics event data in API Connect

The Gateway logs and reports all API interactions to the API Connect analytics engine, for real-time and historical analytics and reporting. Performance-related data about the Management and Gateway servers, and other data relating to system and user status and activities are also captured for display and analysis. By default, this analytics data is accessible from the API Connect user interfaces. You can, if preferred, switch off access to analytics within the API Connect user interfaces.

Before you begin

To enable or disable access to analytics data within API Connect, you must be assigned one of the following roles: Cloud Owner, Cloud Administrator, Topology Administrator, or System User. These roles provide access to the Settings tab in the Cloud Manager user interface from where you can configure security, analytics, and other capabilities. For more information about roles and permissions, see Adding users and assigning roles and API Connect user roles.

About this task

You can enable or disable access to analytics data for one or more of the following event types: API events, monitoring events, log events, and audit events. For more information about these event types and the effect of enabling or disabling access to analytics data in API Connect, see Supported event types for analytics offload.
Important: When you enable or disable access to API Connect Analytics, that change is applied across all provider and developer organizations in the cloud infrastructure.

Procedure

1. In the Cloud Manager, click Settings.
2. From the navigation pane, click Analytics to open the Analytics page.
3. For each event type, enable or disable access to analytics data from within API Connect:

   API Events

   To enable access to API event data for analysis within the API Connect user interfaces, ensure that the associated Use IBM API Connect® Analytics check box is selected. This is the default option.
   To disable access to API event data for analysis within API Connect, clear the Use IBM API Connect Analytics check box.

   Monitoring Events

   To enable access to monitoring event data for analysis within the API Connect user interfaces, ensure that the associated Use IBM API Connect Analytics check box is selected. This is the default option.
   To disable access to monitoring event data for analysis within API Connect, clear the Use IBM API Connect Analytics check box.

   Log Events

   To enable recording of log event data within API Connect, ensure that the associated Use IBM API Connect Analytics check box is selected. This is the default option.
   You must enable the webapi-mntr service to capture log events, which is disabled by default. Because the default state is disabled, you must enable it each time that you refresh your DataPower server. To enable the webapi-mntr service, complete the following steps:

   a. Log in to your DataPower Console.
   b. Select the API Connect domain.
   c. Select Logging Targets in the navigation.
   d. Select webapi-mntr.
   e. Select the checkbox for Enable administrative state.
   f. Select Apply > Save.

   You should only enable the webapi-mntr service temporarily to debug a problem. The service can slow down the rate at which the gateway pushes the API events to the management server.
   Remember: Even with the Log Events enabled and the webapi-mntr service enabled, you cannot view them in the API Connect user interfaces. To view them, you must configure your system to offload the log events.
   To disable recording of log event data within API Connect, clear the Use IBM API Connect Analytics check box.

Audit Events

To enable access to audit event data for analysis within the API Connect user interfaces, ensure that the associated Use IBM API Connect Analytics check box is selected. This is the default option.

To disable access to audit event data for analysis within API Connect, clear the Use IBM API Connect Analytics check box.

4. Click the Save icon 💾 to save the check box settings.
5. Optional: Configure the offload of API, monitoring, log, or audit events to one or more third-party systems. Events are not offloaded by default to third-party systems, as indicated by the Export events to a third-party system check box, which is initially clear for each event type.
   For more information, see Configuring the offload of analytics event data to third-party systems.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Configuring the offload of analytics event data to third-party systems

You can forward the analytics data that is captured for API Connect events to a number of third-party systems as a real-time data stream. This capability is useful if you want to consolidate data from multiple sources, if you require enhanced monitoring, or if you want to enrich your analytics data. Target systems to which you can offload data include HTTP, Elasticsearch, Kafka, and Syslog servers.

Before you begin

Collect the connection details and other configuration information that is required to configure a data offload to the target system. The configuration requirements are as follows:

- To configure data offload to an HTTP server, you must, at a minimum, provide the server URL. If you require encrypted communication with the server, you must also have a Transport Layer Security (TLS) profile defined. You can optionally add standardized or custom HTTP headers to define additional operating parameters.
- To configure data offload to an Elasticsearch cluster, you must, at a minimum, provide one or more server URLs, define how indexes 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. If you require encrypted communication with the cluster, you must also have a TLS profile defined. You can optionally specify server authentication credentials.
- To configure data offload to a Kafka cluster, you must, at a minimum, 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. If you require encrypted communication with the cluster, you must also have a TLS profile defined. For logging and monitoring purposes within Kafka, you can optionally specify a string identifier by which API Connect can be uniquely identified.
- To configure data offload to Syslog, you must, at a minimum, provide host and port connection details for the server. If you require encrypted communication with Syslog using the TCP protocol, you must also have a TLS profile defined.

It is advisable that you create one or more TLS profiles that you can use for encrypted communication instead of using the default TLS profile named Cloud Manager and API Manager TLS Profile. For information about creating TLS profiles, see TLS profiles.
To offload analytics data to third-party systems, you must be assigned one of the following roles: Cloud Owner, Cloud Administrator, Topology Administrator, or System User. These roles provide access to the Settings tab in the Cloud Manager user interface from where you can configure security, analytics, and other capabilities. For more information about roles and permissions, see Adding users and assigning roles and API Connect user roles.

About this task

You can offload data for one or more of the following event types: API events, monitoring events, log events, and audit events. You can, at the same time, continue to stream the analytics event data for display within the API Connect user interfaces, or you can switch off the display of analytics within API Connect if preferred. For information about these event types and the effect of enabling or disabling access to analytics data in API Connect, see Supported event types for analytics offload.
Important: When you configure the offload of analytics data, or enable or disable access to analytics data in API Connect, that change is applied across all provider and developer organizations in the cloud infrastructure.
If you want to use Transport Layer Security to secure the offloaded data while in transit to the target system, ensure that the target server or cluster of servers are TLS-enabled with a public/private key pair and signed certificate. When the API Connect Management server initiates a connection, the certificate that a target server presents (which contains the identity of the target server, the public key, and the digital signature of the certificate issuer), must be signed by a Certificate Authority (CA) that the Management server trusts. To verify the signature on a target server's certificate, the Management server requires the public key of the issuing CA. Because public keys are distributed in

certificates, this means that the Management server must have a certificate for the issuing CA. This certificate must be signed by the CA. If the Management server cannot determine whether the target server is a trusted source, the connection will be rejected.

Procedure

1. In the Cloud Manager, click Settings.
2. From the navigation pane, click Analytics to open the Analytics page.
    a. To offload data to HTTP, go to step 3.
    b. To offload data to Elasticsearch, go to step 4.
    c. To offload data to Kafka, go to step 5.
    d. To offload data to Syslog, go to step 6.
3. To offload the analytics data for a specific event type to an HTTP server, complete the following fields that are available under the API Events, Monitoring Events, Log Events, or Audit Events section:
    a. Select the Export events to a third-party system check box.
    b. From the Select Analytics Platform drop-down list that is displayed, select HTTP.
    c. Click Configure to specify connection details and other configuration options for the HTTP server.
    d. Complete the fields in the HTTP Output window as follows:
        i. In the URL field, specify the URL of the HTTP server.
        ii. To use TLS to set up a private connection to the HTTP server and secure the transmission of the data being offloaded, select the Use TLS check box. Then, select your preferred profile from the drop-down list. This list shows all TLS profiles that have been created in the Cloud Manager. For information about creating TLS profiles, see TLS profiles. The default TLS profile (Cloud Manager and API Manager TLS Profile), which is defined in the Cloud Manager, is selected by default.
        iii. If you selected Use TLS, and you want to verify the authenticity of the TLS certificate presented by the HTTP server before you establish a connection, select the Validate Certificate check box.
        iv. To transmit additional operating parameters in the header section of the offloaded event records, click the Add Header button and specify a header field name and associated value for a standardized or custom HTTP header. You can use this button to add one or multiple name/value pairs for HTTP headers.
        If you want to forward the event data to a system such as Splunk, which provides enhanced search, analysis, and visualization capabilities, you can add custom HTTP headers that the API Connect Management server can use to authenticate to the Splunk server, and which Splunk can use to determine where the analytics data should be sent. For an example of how this can be configured, see HTTP.

        Tip: You can remove any unwanted headers from the list of specified name/value pairs by clicking the associated Delete icon 🗑.
        v. To verify that your connection and configuration details are valid, click Send Test Event to generate and transmit a test event to the HTTP server.
        Restriction: For HTTP output, it cannot be determined whether the test event was successfully transmitted to the target server. Therefore, an error message is always displayed regardless of whether the transmission was successful, and you must check on the target server to see whether the test event was successfully received. If the test event was not received by the target system, verify that your specified configuration settings are correct.
        You can verify that the target system has received the event by looking for an event that includes `5apimanagement_testevent` as a value in the event record. To see examples of the test events that are generated for API, monitoring, log, and audit events, see Sample test events for analytics offload.

        vi. Click Update to store the settings configured for offloading data to the HTTP server.
    e. Click the Save icon 💾 to collectively save your defined settings for enabling or disabling access to analytics, and for offloading data.
4. To offload the analytics data for a specific event type to an Elasticsearch cluster of one or more servers, complete the following fields that are available under the API Events, Monitoring Events, Log Events, or Audit Events section:
    a. Select the Export events to a third-party system check box.
    b. From the Select Analytics Platform drop-down list that is displayed, select Elasticsearch.
    c. Click Configure to specify connection details and other configuration options for the Elasticsearch cluster.
    d. Complete the fields in the Elasticsearch Output window as follows:
        i. Add location details for one or more Elasticsearch servers which belong to the cluster that hosts the shards into which you want to offload data.
            1. In the URL 1 field, specify the URL of a server in the cluster.
            2. For each subsequent server in the cluster, click Add URL and then specify a URL for the server location.
        Tip: You can remove any unwanted servers by clicking the associated Delete icon 🗑.
        ii. To use TLS to set up a private connection to the Elasticsearch cluster and secure the transmission of the data being offloaded, select the Use TLS check box. Then, select your preferred profile from the drop-down list. This list shows all TLS profiles that have been created in the Cloud Manager. For information about creating TLS profiles, see TLS profiles. The default TLS profile (Cloud Manager and API Manager TLS Profile), which is defined in the Cloud Manager, is selected by default.
        iii. If you selected Use TLS, and you want to verify the authenticity of the TLS certificates presented by the cluster before you establish a connection, select the Validate Certificate check box.

iv. If authentication is required for access to the Elasticsearch cluster, specify valid credentials in the User and Password fields. Otherwise, ensure that these fields are blank.

v. In the Index Root field, specify a static base name (in lower case) for one or more indexes that will be used to store the offloaded data. The specified root value is combined with the suffix that you specify in the adjoining Index Suffix field to determine the destination Elasticsearch index for the data.
When the first event is streamed to Elasticsearch, an index is created with a name that is constructed from the static index root value and specified suffix. Because the suffix value is specified as a variable that describes a date pattern or dynamic field, subsequent indexes with the same index root name, but varying suffix values, can be created at appropriate points in the future. API Connect provides an Elasticsearch template that is automatically applied when an index, named using the specified static index root and variable suffix, is created.

vi. In the Index Suffix field, specify the syntax that represents a variable value that you want to append to the Index Root value, to construct an index name:

- You can specify a date pattern that enables you to partition your data. For example, `-%{+YYYY.MM.dd}` creates an Elasticsearch index daily. So for an index root value of `apiconnectevents` and a suffix of `-%{+YYYY.MM.dd}`, an index is created daily using the naming convention **apiconnectevents-*YYYY.MM.dd***.
- You can use percent literal syntax with any field name in an event record to specify a dynamic value. For example, `-%{org_id}` creates an organization-specific Elasticsearch index. So for an index root value of `apiconnectevents` and a suffix of `-%{org_id}`, an index is created when the first event from each provider organization is streamed to Elasticsearch. Each index uses the naming convention **apiconnectevents-*provider_organization_ID***. For information about the fields that can be specified for API events, see API event record fields.
  For more information about constructing index names, see the *index* section in https://www.elastic.co/guide/en/logstash/current/plugins-outputs-elasticsearch.html#plugins-outputs-elasticsearch-index and the *sprintf format* section in https://www.elastic.co/guide/en/logstash/current/event-dependent-configuration.html.

vii. In the Primary Shards field, specify the number of primary shards that are allocated to the specified Elasticsearch servers in the cluster. The offloaded data will be stored and indexed as Elasticsearch documents in these shards. The specified value determines how many primary shards are used to store the data in any new indexes that are created with the static index root value and variable suffix. Existing indexes are not affected. By default, 5 primary shards are defined.

viii. In the Replica Shards field, specify the number of replica shards that are defined for each primary shard, to hold copies of the offloaded data that is stored in each primary shard. The specified value is used to determine how many replica shards should be used to store copies of data in any new indexes that are created with the specified index suffix. Existing indexes are not affected. By default, 1 replica is defined for every primary shard.

ix. To verify that your connection and configuration details are valid, click Send Test Event to generate and transmit a test event to the Elasticsearch cluster. Messages are displayed beneath the primary banner in the Cloud Manager UI to indicate that an analytics event is being sent, and to indicate that the transmission was successful. If the transmission failed, an error message is displayed on top of the current window to inform you of the failure. Click OK to close the error message and then verify that your configuration settings are correct.
You can verify that the target system has received the event by looking for an event that includes **`5apimanagement_testevent`** as a value in the event record. To see examples of the test events that are generated for API, monitoring, log, and audit events, see Sample test events for analytics offload.

x. Click Update to store the settings configured for offloading data to the Elasticsearch cluster.

e. Click the Save icon 💾 to collectively save your defined settings for enabling or disabling access to analytics, and for offloading data.

5. To offload the analytics data for a specific event type to a Kafka system that runs as a cluster on one or more servers, complete the following fields that are available under the API Events, Monitoring Events, Log Events, or Audit Events section:

a. Select the Export events to a third-party system check box.
b. From the Select Analytics Platform drop-down list that is displayed, select Kafka.
c. Click Configure to specify connection details and other configuration options for the Kafka cluster.
d. Complete the fields in the Kafka Output window as follows:

i. Add location details for one or more Kafka servers that belong to the cluster into which you want to offload data.

1. In the Host 1 field, specify a fully qualified host name or IP address for a server in the cluster.
2. In the Port 1 field, specify the port number on which the server listens for incoming connections.
3. For each subsequent server in the cluster, click Add Server and then specify host and port details for that server. These host and port details will be used to establish the initial connection to the Kafka cluster, and to discover the full cluster membership. Because the cluster membership can change dynamically, you do not need to specify connection details for the full set of servers, but must specify at least two, in case a server is down.

Tip: You can remove any unwanted servers by clicking the associated Delete icon 🗑.

ii. To use TLS to set up a private connection to the Kafka cluster and secure the transmission of the data being offloaded, select the Use TLS check box. Then, select your preferred profile from the drop-down list. This list shows all TLS profiles that have been created in the Cloud Manager. For information about creating TLS profiles, see TLS profiles. The default TLS profile (Cloud Manager and API Manager TLS Profile), which is defined in the Cloud Manager, is selected by default.

        iii. In the Topic ID field, specify the name of the Kafka topic to which you want to publish the offloaded analytics data. If the Kafka dynamic topics function is enabled, then the topic that you specify is created automatically if it does not exist. If the Kafka dynamic topics function is disabled, then you must specify an existing topic.
        iv. In the Client ID field, specify a meaningful, distinguishing string value that API Connect can pass to Kafka when connecting to its cluster, and which can be used in server-side logging or monitoring.
        v. To verify that your connection and configuration details are valid, click Send Test Event to generate and transmit a test event to the Kafka cluster. Messages are displayed beneath the primary banner in the Cloud Manager UI to indicate that an analytics event is being sent, and to indicate that the transmission was successful. If the transmission failed, an error message is displayed on top of the current window to inform you of the failure. Click OK to close the error message and then verify that your configuration settings are correct.
            You can verify that the target system has received the event by looking for an event that includes `5apimanagement_testevent` as a value in the event record. To see examples of the test events that are generated for API, monitoring, log, and audit events, see Sample test events for analytics offload.
        vi. Click Update to store the settings configured for offloading data to the Kafka cluster.
    e. Click the Save icon 💾 to collectively save your defined settings for enabling or disabling access to analytics, and for offloading data.

6. To offload the analytics data for a specific event type to a Syslog server, complete the following fields that are available under the API Events, Monitoring Events, Log Events, or Audit Events section:
    a. Select the Export events to a third-party system check box.
    b. From the Select Analytics Platform drop-down list that is displayed, select Syslog.
    c. Click Configure to specify connection details and other configuration options for the Syslog server.
    d. Complete the fields in the Syslog Output window as follows:
        i. In the Host field, specify a fully qualified host name or IP address of the Syslog server.
        ii. In the Port field, accept the default port (which is set based on the protocol that you specify in the next field, for secure HTTP connections). Alternatively, specify another port number on which Syslog listens for incoming connections. The default port is 514 if connecting using the UDP protocol, and 601 for the TCP protocol.
        iii. From the Protocol drop-down list, specify your preferred protocol for secure HTTP connections from either of the following options:
            - UDP: The connectionless User Datagram Protocol (UDP), which is generally used for simpler messaging transmissions, offers no guarantee of delivery, and has no handshaking dialogues
            - TCP: The more complex Transmission Control Protocol (TCP), which is used for connection-oriented transmissions, with reliable, ordered data streaming between communicating network applications and error-correction facilities
        iv. To use TLS to set up a private connection to the Syslog server and secure the transmission of the data being offloaded, select the Use TLS check box. Then, select your preferred profile from the drop-down list. This list shows all TLS profiles that have been created in the Cloud Manager. For information about creating TLS profiles, see TLS profiles. The default TLS profile (Cloud Manager and API Manager TLS Profile), which is defined in the Cloud Manager, is selected by default.
            Restriction: Syslog does not support the use of chained client certificates for TLS profiles.
            Note: The Use TLS check box and corresponding TLS profile drop-down list are available only if you selected TCP in the Protocol field.
        v. If you specified TCP as the protocol and selected the Use TLS check box, select the Validate Certificate check box if you want to verify the authenticity of the TLS certificate presented by the Syslog server before you establish a connection.
        vi. To verify that your connection and configuration details are valid, click Send Test Event to generate and transmit a test event to the Syslog server. Messages are displayed beneath the primary banner in the Cloud Manager UI to indicate that an analytics event is being sent, and to indicate that the transmission was successful. If the transmission failed, an error message is displayed on top of the current window to inform you of the failure. Click OK to close the error message and then verify that your configuration settings are correct.
            You can verify that the target system has received the event by looking for an event that includes `5apimanagement_testevent` as a value in the event record. To see examples of the test events that are generated for API, monitoring, log, and audit events, see Sample test events for analytics offload.
        vii. Click Update to store the settings configured for offloading data to the Syslog server.
    e. Click the Save icon 💾 to collectively save your defined settings for enabling or disabling access to analytics, and for offloading data.

7. Optional: For each event type, enable or disable access to analytics data from within API Connect:
    a. To enable analytics data storage and access to analytics data in the API Connect UIs, ensure that the Use IBM API Connect® Analytics check box is selected under the API Events, Monitoring Events, or Audit Events section. To enable analytics data storage, ensure that the Use IBM API Connect Analytics check box is also selected in the Log Events section. Note that there are no UI visualizations for Log Events. The default setting for the Use IBM API Connect Analytics check box is selected in each of these cases.
    b. To disable analytics data storage and access to analytics data in API Connect, clear the Use IBM API Connect Analytics check box.
    c. Click the Save icon 💾 to collectively save your defined settings for enabling or disabling access to analytics, and for offloading data.

    For more information, see Enabling or disabling access to analytics event data in API Connect.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Sample test events for analytics offload

When you configure the offload of analytics data to third-party systems, you can use the Send Test Event button to verify that your configuration settings are valid before you save your settings. This button generates and sends a test event to the target system.

For information about configuring the offload of analytics data and generating test events, see Configuring the offload of analytics event data to third-party systems.

- Sample API test event
- Sample monitoring test event
- Sample log event
- Sample audit test event

Sample API test event

The following example shows a sample API test event that is generated and transmitted to a target system. The `org_id` field identifies this as a test event. For descriptions of the fields for test events, see API event record fields.

```
{
  "datetime": "2017-03-14T13:12:28.732Z",
  "app_type": "PRODUCTION",
  "debug": [
    {
      "transid": "57fe2ad1e4b0447b716d5112-2a455eeb-ebd0-466d-b651-0bdec227db4f"
    },
    {
      "name": "API Information",
      "input": {
        "parameters": {
          "properties": [
            {
              "request": {
                "request-uri": "https:\/\/localhost:443\/testorg\/sb\/jewellery\/watches\/stocklist",
                "org": "testorg",
                "env": "sb"
              },
              "plan": {
                "name": "",
                "id": "",
                "version": ""
              },
              "api": {
                "name": "watches",
                "type": "REST",
                "version": "1.0.0",
                "basepath": "jewellery\/watches",
                "operation": "stocklist"
              },
              "application": {
                "app-name": "",
                "client-id": ""
              }
            }
          ]
        }
      }
    },
    {
      "name": "Activity Log",
      "result": "OK",
      "input": {
        "parameters": {
          "logLevel": "header",
          "onError": "payload"
        }
      }
    },
    {
      "input": {
        "request": {
          "headers": {
            "Accept": "application\/json, text\/plain, *\/*",
            "Origin": "https:\/\/localhost",
            "APIm-Debug": "true",
            "User-Agent": "IBM-APIConnect\/tru",
            "Referer": "https:\/\/localhost\/apim\/",
            "Accept-Language": "en-US,en;q=0.8",
            "Via": "1.1 AwAAAK1Kyws-",
            "X-Client-IP": "9.27.111.170",
            "X-Global-Transaction-ID": "135409"
          }
        }
      },
      "output": {
        "message": {
          "body": "{\"country\":\"United 
Kingdom\",\"capital\":\"London\",\"totalPopulation\":\"64100000\",\"gpsCoordinates\":\"51.5072 N, 0.1275 
W\"}",
          "headers": {
            "X-Backside-Transport": "OK OK",
            "Content-Type": "application\/json",
            "Date": "Wed, 12 Oct 2016 17:08:28 GMT",
            "X-Global-Transaction-ID": "135409"
          },
          "status": {
            "code": 200,
            "reason": "OK"
          }
        }
      },
      "result": "OK",
      "endpoint": "http:\/\/localhost/endpoint",
      "properties": {
        "title": "Watcheservice",
        "target-url": "http:\/\/localhost/endpoint"
      },
      "name": "Watcheservice"
    }
  ],

  "@timestamp": "2016-10-12T17:08:28.829Z",
  "host": "127.0.0.1",
  "headers": {
    "request_method": "POST",
    "request_path": "\/_bulk",
    "request_uri": "\/_bulk",
    "http_version": "HTTP\/1.1",
    "http_host": "localhost:9700",
    "http_organization": "admin",
    "content_type": "text\/plain",
    "http_via": "1.1 AQAAANQgXos-",
    "http_x_client_ip": "127.0.0.1",
    "http_x_global_transaction_id": "41541",
    "http__ws_haprt_wlmversion": "-1",
    "http_x_forwarded_for": "localhost",
    "http_x_forwarded_host": "localhost:9443",
    "http_x_forwarded_server": "localhost",
    "http_connection": "Keep-Alive",
    "content_length": "3760"
  },
  "client_ip": "127.0.0.1",
  "latency_info": [
    {
      "task": "Start",
      "started": 0
    },
    {
      "task": "Plan Limit",
      "started": 7
    },
    {
      "task": "activity-log",
      "started": 9
    },
    {
      "task": "Watcheservice",
      "started": 11
    }
  ],
  "api_id": "57fe2d41e4b0447b716d5146",
  "org_id": "5apimanagement_testevent",
  "developer_org_name": "testdevorg",
  "plan_name": "testplan",
  "app_name": "testapp",
  "product_name": "testproduct",
  "rateLimit": {
    "rate-limit": {
      "limit": "10",
      "count": "5"
    },
    "rate-limit-1": {
      "limit": "15",
      "count": "4"
    },
    "rate-limit-2": {
      "limit": "12",
      "count": "11"
    },
    "per-minute": {
      "limit": "30",
      "count": "28"
    }
  },
  "api_version": "1.0.0",
  "catalog_id": "456",
  "plan_id": "",
  "developer_org_id": "",
  "resource_id": "watches:1.0.0:get:\/stocklist",
  "transaction_id": "135409",
  "uri_path": "\/testorg\/sb\/jewellery\/watches\/stocklist",
  "request_method": "GET",
  "log_policy": "header",
  "gateway_ip": "127.0.0.1",
  "http_user_agent": "Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_11_0) AppleWebKit\/537.36 (KHTML, like 
Gecko) Chrome\/53.0.2785.116 Safari\/537.36",
  "request_protocol": "https",
  "query_string": [

  ],
  "request_body": "",
  "response_body": "",
  "bytes_received": 0,
  "bytes_sent": 115,
  "time_to_serve_request": 534,
  "status_code": "200 OK",
  "request_http_headers": [
    {
      "Host": "localhost"
    },
    {
      "Accept": "application\/json, text\/plain, *\/*"
    },
    {
      "Origin": "https:\/\/localhost"
    },
    {
      "APIm-Debug": "true"
    },
    {
      "User-Agent": "Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_11_0) AppleWebKit\/537.36 (KHTML, like 
Gecko) Chrome\/53.0.2785.116 Safari\/537.36"
    },
    {
      "Referer": "https:\/\/localhost\/apim\/"
    },
    {
      "Accept-Encoding": "gzip, deflate, sdch, br"
    },
    {
      "Accept-Language": "en-US,en;q=0.8"
    },
    {
      "Via": "1.1 AwAAAK1Kyws-"
    },
    {
      "X-Client-IP": "127.0.0.1"
    },
    {
      "X-Global-Transaction-ID": "135409"
    }
  ],
  "response_http_headers": [
    {
      "Content-Type": "application\/json"
    },
    {
      "Date": "Wed, 12 Oct 2016 17:08:28 GMT"
    },
    {
      "X-Global-Transaction-ID": "135409"
    },
    {
      "User-Agent": "IBM-APIConnect\/tru"
    },
    {
      "X-Backside-Transport": "OK OK"
    },
    {
      "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:\/\/localhost"
    },
    {
      "Access-Control-Allow-Methods": "GET"
    },
    {
      "Access-Control-Allow-Credentials": "true"
    }
  ],
  "space_id": [
    "57fe2cd3e4b0447b716d5136"
  ],
  "org_name": "testorg",
  "api_name": "watches",
  "catalog_name": "sb",
  "resource_path": "get",
  "client_geoip": {
    "ip": "127.0.0.1",
    "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": "127.0.0.1",
    "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
    ]
  }
}
```

Sample monitoring test event

The following example shows a sample monitoring event that is generated and transmitted to a target system. The `nodeId` field identifies this as a monitoring test event. For a description of the monitor event fields, see Monitoring event fields.

```
{
   "datetime":"2017-01-09T15:23:50.585Z",
   "nodeId":"5apimanagement_testevent",
   "nodeType":"gw",
   "plLoad":1.0,
   "plGwRxTp":0.0,
   "plGwTxTp":0.0,
   "gwMem":0.0,
   "gwTranRate":0.0,
   "gwMeanTranTime":0.0,

   "host":"127.0.0.1",
   "headers":{"request_method":"POST",
   "request_path":"/apimgmt_XXXXXXXXXX/monitoringevent",
   "request_uri":"/apimgmt_XXXXXXXXXX/monitoringevent",
   "http_version":"HTTP/1.1",
   "http_host":"localhost:9700",
   "http__ws_haprt_wlmversion":"-1",
   "http_organization":"apimgmt_a8c0e202ca",
   "content_type":"application/json",
   "http_x_forwarded_for":"127.0.0.1",
   "http_x_forwarded_host":"127.0.0.13",
   "http_x_forwarded_server":"localhost",
   "http_connection":"Keep-Alive",
   "content_length":"409"},
   "memory":20.0,
   "cpu":0.0,
   "used_disk":9.0
}
```

Sample log event

The following example shows a sample log test event that is generated and transmitted to a target system. The `nodeId` field identifies this as a test event. For more information about the log event record fields, see Log event fields.

```
{
   "datetime":"2016-12-02T16:28:41.000Z",
   "nodeId":"5apimanagement_testevent",
   "nodeType":"gw",
   "messageId":"0x80e00161",
   "category":"mpgw",
   "level":"severe",
   "transaction":"3645905",
   "source":"apim-source-https-webapi-https",
   "client":"127.0.0.1",
   "message":"Request processing failed: Connection terminated before request headers read because of 
the connection error occurs from URL: 127.0.0.1:4507",

   "host":"127.0.0.1",
   "headers":{
      "request_method":"POST",
      "request_path":"/apim/logevent",
      "request_uri":"/apim/logevent",
      "http_version":"HTTP/1.1",
      "http_host":"localhost:9700",
      "http_soapaction":"\"\"",
      "content_type":"application/json",
      "http_via":"1.1 AwAAAE91u64-",
      "http_x_client_ip":"127.0.0.1",
      "http_x_global_transaction_id":"3645969",
      "http_organization":"apim",
      "http__ws_haprt_wlmversion":"-1",
      "http_x_forwarded_for":"127.0.0.1",
      "http_x_forwarded_host":"127.0.0.1",
      "http_x_forwarded_server":"localhost",
      "http_connection":"Keep-Alive",
      "content_length":"401"
   }
}
```

Sample audit test event

The following example shows a sample audit test event that is generated and transmitted to a target system. The `message` field identifies this as a test event. For more information about the audit event fields, see Audit event fields.

```
{
  "headers": {
    "http_accept": "application\/json",
    "content_type": "application\/json",
    "request_path": "\/cmc-2017.03\/auditevent",
    "http_version": "HTTP\/1.1",
    "http_connection": "keep-alive",
    "request_method": "POST",
    "http_host": "localhost:9700",
    "request_uri": "\/cmc-2017.03\/auditevent",
    "content_length": "289",
    "http_user_agent": "Wink Client v1.1.1"
  },
  "nlsMessage": {
    "resource": "messages",
    "replacements": [
      "testuser@apimanagement",
      "apimanager"
    ],
    "key": "notification.user.login"
  },
  "notificationType": "EVENT",
  "eventType": "AUDIT",
  "source": null,
  "message": "This is a test message for apimanagement_testorg, 5apimanagement_testevent.",
  "tags": [
    "_geoip_lookup_failure"
  ],
  "gateway_geoip": {

  },
  "datetime": "2017-03-07T23:36:25.070Z",
  "@timestamp": "2017-03-07T23:36:25.074Z",

  "host": "127.0.0.1",
  "id": null,
  "client_geoip": {

  }
}
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Troubleshooting your analytics offload

Review this guidance to help you resolve issues if your data offload fails. Common causes might be connection or security-related.

Before you save your configuration settings for offloading an event type to a third-party system, use the Send Test Event button to verify that a test event can be successfully transmitted. Also verify that you can see the test event in the target system. If the test event fails, complete the following checks:

- 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 remaining configuration settings are valid; for example, the user credentials for authenticating to Elasticsearch or the Kafka topic name.

To help with problem determination, review the Management server log files:

- You can use the API Connect Command Line Interface (CLI) to either run debugging commands to view the log file contents as they are added to the Logstash log files, or to generate a postmortem archive of all the latest Management server logs for review. The API Connect CLI is preinstalled on the Management server; you can log in to the CLI on the Management server by using a Secure Shell (SSH) session, terminal emulation, or telnet. For more information about logging in to the CLI, see The Command Line Interface. From the API Connect CLI, you can run the following **debug tail** commands on the Logstash log files and look for errors related to connection issues, SSL certificate issues, or other configuration issues. It is recommended that you monitor these logs for a few minutes to check whether the output changes. Start with the /var/log/logstashconfigure.log before moving on to review /var/log/logstash.log.

  ```
  debug tail file /var/log/logstashconfigure.log
  ```

  ```
  debug tail file /var/log/logstash.log
  ```

  If required, you can also run the following commands to generate a postmortem archive of all the latest Management server logs and to export the postmortem archive from your appliance to an FTP or SFTP server. When you run the command to export the archive, you are prompted to enter your password.

  ```
  debug postmortem generate newlogs
  ```

  ```
  debug postmortem export ftp ftpserver_hostname user my_user_name file file_path_name.zip
  ```

  For more information about using the debugging commands, see Debugging commands.

- You can alternatively collect and download postmortem information about your Management servers in a compressed (.zip) file by using the Cloud Manager user interface. You can then extract and review the /var/log/logstashconfigure.log and /var/log/logstash.log log files. For more information, see Gathering postmortem information about your servers.

After successfully configuring your API Connect data offload, consider implementing checks within your target systems to monitor the continuity of your data streams and to raise alerts if needed.

Related tasks

- TLS profiles

Related information

- Logstash

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Administering provider organizations

Manage the provider organizations that share in the use of your API Connect cloud.

About this task

Use the following tasks to administer your provider organizations:

- **Creating a provider organization account**
  For developer organizations to be able to share in your cloud and call published APIs, you first create a provider organization account and add an owner to the account.

- **Updating a provider organization account**
  You can update key information that is related to a provider organization account, including the name and the owner.
- **Deleting a provider organization account**
  You can only delete a provider organization account after the organization's Catalogs and resources are removed from the API Connect cloud. Empty Catalogs and organization memberships can be batch deleted by the Organization Manager. After being deleted, the organization is removed, however, the Organization Manager account remains in API Manager.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a provider organization account

For developer organizations to be able to share in your cloud and call published APIs, you first create a provider organization account and add an owner to the account.

Before you begin

This task can be completed by users who are assigned the following roles:

- Cloud Owner
- Organization Manager

You must also complete the following tasks before beginning:

1. Configuring the Management service
2. Configuring the initial Gateway service
3. Adding a Gateway server
4. Selecting the DNS scheme
5. Specifying the cloud settings
6. Connecting to an existing email server

About this task

You can create a provider organization account, then specify an owner for that organization.

Procedure

1. In the Cloud Manager, click Organizations.
2. In the Organizations pane, click Add Organization.
3. In the Display Name field, enter the name of the organization.
4. In the Name field, enter the string that is included in the organization segment of the URL for API calls.
   For more information, see Calling an API.
   Restriction: The path can consist of the following valid characters:
   - Lowercase alphabetic (a through z)
   - Numerals (0 through 9)
   - Hyphen (-)

   Note: a hyphen cannot be used as the first or last character. Also, the path must be 1-255 characters in length.

5. Specify the owner of the organization.
   - If you are using a local registry as the Identity provider: To add an existing user, click Existing User, enter a search string and select the user from the Search Results window. To add a new user, click New User and enter the user's email into the text field
   - If the provider is LDAP: Enter a search string and select the user from the Search Results scroll window.
   - For both local registry and LDAP: If the search returns more than 100 results, refine your filter string.
   - If the provider is an authentication URL, enter the email address of the intended invitee.

Important: When a user is invited to join a provider organization, secured by using LDAP, the username might (or might not) be case-sensitive, depending on how the user name is set in LDAP. For example, if LDAP is set as case-sensitive and the invitation includes a user name of USERA@mail.com, the user must sign into API Manager with USERA@mail.com (and not usera@mail.com). If the user name is entered incorrectly, the user cannot sign in and an error is displayed.

6. Click Save to set the owner and close the Add Owner dialog box.
7. When you are finished, click Add.
   The organization is added to the Organizations page, and an email invitation is delivered to the owner. `Pending` is displayed in the Status column of the Organizations pane, unless the new owner is an existing LDAP user, in which case a status of `Active` is displayed.

Results

The organization and owner are added to your cloud. If you selected a new user, the API Connect email invitation is delivered. The email invitation is shown as `Invitation Pending` until the recipient of the email clicks the link in the email to complete the creation of the account.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Updating a provider organization account

You can update key information that is related to a provider organization account, including the name and the owner.

Before you begin

This task can be completed by users who are assigned the following roles:

- Cloud Owner
- Organization Manager

Procedure

To update a provider organization account, complete the following tasks:

1. In the Cloud Manager, click Organizations.
2. Navigate to the organization that you want to update, click the Manage icon ▼, then click Update Organization.
   - If you are using a local registry as the Identity provider: To add an existing user, click Existing User, enter a search string and select the user from the Search Results window. To add a new user, click New User and enter the user's email into the text field
   - If the provider is LDAP: Enter a search string and select the user from the Search Results scroll window.
   - For both local registry and LDAP: If the search returns more than 100 results, refine your filter string.
   - If the provider is an authentication URL, enter the email address of the intended invitee.
3. When you are finished, click Update. Your changes are displayed.
   If you changed the owner name and the owner is new to API Connect, the email invitation is delivered and `Pending` is displayed in the Status column. If the owner is an existing user, the status is displayed as `Active`.

Results

The provider organization account information is updated.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deleting a provider organization account

You can only delete a provider organization account after the organization's Catalogs and resources are removed from the API Connect cloud. Empty Catalogs and organization memberships can be batch deleted by the Organization Manager. After being deleted, the organization is removed, however, the Organization Manager account remains in API Manager.

Before you begin

This task can be completed by users who are assigned the following roles:

- Cloud Owner
- Organization Manager

About this task

When you delete a provider organization account, it is removed permanently and users of the account can no longer access the API Connect cloud.

Procedure

To delete an organization account, complete the following tasks:

1. In the Cloud Manager, click Organizations.
2. Navigate to the organization that you want to delete, click the Manage icon ▾, then click Delete Organization.
3. Click OK to delete the organization.
   The organization is deleted and removed from the Organizations pane.

Results

The organization is deleted.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Administering user access

If you have either the Cloud Administrator or Cloud Owner role in the Cloud Manager, you can add additional administrators and users, and assign them roles, enabling them to perform administrative duties and manage provider organizations.

About this task

You can also delete users to prevent them from accessing the Cloud Manager. However, after deletion, the user's profile remains in API Manager.

To add users and assign to the Cloud Manager, perform the followings tasks:

- **Viewing your user information**
  To view current users of the Cloud Manager, you access the Users tab.
- **Adding users and assigning roles**
  As the Cloud Manager administrator, you can add and remove users and assign roles. Once removed, a user can no longer access the Cloud Manager. However, the user's profile remains in API Manager.
- **Deleting a user account**
  As the Cloud Manager administrator, you can delete users. Once deleted, the user and associated roles are removed, however the user's account still remains in API Manager.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing your user information

To view current users of the Cloud Manager, you access the Users tab.

Procedure

To view information about the users of the Cloud Manager, complete the following steps:

1. In the Cloud Manager, click Users.
2. In the Cloud Manager, click Members.
3. In the Users page, you can scroll through the list of users to see their user role authorizations, whether they are active and when they last logged in.
   Note: You can use the Users page to remove users, which prevents them from accessing the Cloud Manager. However, the user's profile remains in API Manager.

Results

All of your user accounts are listed.

What to do next

To modify your user accounts, see Adding users and assigning roles,
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding users and assigning roles

As the Cloud Manager administrator, you can add and remove users and assign roles. Once removed, a user can no longer access the Cloud Manager. However, the user's profile remains in API Manager.

About this task

Perform the following steps to add users to the Cloud Manager.

Procedure

1. In the Cloud Manager, click Users.
2. In the Cloud Manager, click Members.
3. In the Users page, click Add.
   The Add User window is displayed.

- If you are adding a role to an existing user that is not displayed on the Users page, complete the following steps:
  a. Click Existing User.
  b. Click the search icon to list all users, or enter a search term and click the search icon to list a subset of users.
  c. Select the required user from the search results.
- If you are adding a new user, click New User and enter the user's email into the text field.

4. To assign the user a role, complete the following steps:

   a. Click the Add organization role to this user icon ⊕ icon.
   b. Select the required user role. The assigned role is displayed in the Role column.
   The available roles are:

   Cloud Owner
      The Cloud Owner can access all Cloud Manager tabs and perform all activities associated with those tabs.
      Note: The Cloud Owner role cannot be assigned to a user.

Cloud Administrator

The Cloud Administrator can access all Cloud Manager tabs except the Organizations tab and perform all activities associated with those tabs.

Organization Manager

The Organization Manager can access the Organizations tab and perform all activities associated with that tab.

V5.0.5 + System User V5.0.4 and earlier System

This role can be assigned to any user of the Cloud Manager. Users assigned this role have full Cloud Manager access. In addition, users have REST access to all APIs but cannot access the API Manager or the Developer Portal user interfaces.

Topology Administrator

The Topology Administrator role can be assigned to any user of the Cloud Manager. Users assigned this role can log into the Cloud Manager and access the Home, Clusters, SSL Profiles and Settings tabs.

The following table shows which Cloud Manager tabs each user role can access.

Table 1. Permitted user role activities

Tab	Cloud Owner	Cloud Administrator	Organization Manager	V5.0.5 + System User	V5.0.4 and earlier System	Topology Administrator
Analytics	Yes	Yes	No	Yes	Yes	Yes
Services	Yes	Yes	No	Yes	Yes	Yes
Organizations	Yes	No	Yes	Yes	Yes	No
Users	Yes	Yes	No	Yes	Yes	No
TLS Profiles	Yes	Yes	No	Yes	Yes	Yes
User Registries	Yes	Yes	No	Yes	Yes	Yes
Settings	Yes	Yes	No	Yes	Yes	Yes

For full details of the permissions that are assigned to the various Cloud Manager roles, see API Connect user roles.

5. Click Add.

   The user name is added to the Name column, and the email invitation is delivered if the user is a new user. `Invitation Pending` is displayed in the Status column of the Users window.

Results

The API Connect user account is created and is activated when the user follows the activation link in the email invitation.

What to do next

The new user can access the Cloud Manager user interface. The user's authorization is defined by the roles assigned to them.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.

For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deleting a user account

As the Cloud Manager administrator, you can delete users. Once deleted, the user and associated roles are removed, however the user's account still remains in API Manager.

About this task

Complete the following steps to delete a user. Once deleted, the user can no longer access the Cloud Manager, however the user account still remains.

Procedure

1. In the Cloud Manager, click Users.
2. In the Cloud Manager, click Members.
3. Click the Delete icon that is adjacent to the user you want to delete.

Results

The user is deleted and removed from the Users page. However, the user account still remains in API Manager.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Authentication

As the Cloud Manager administrator, you can add new user registries to securely authenticate your Catalogs and APIs and create SSL profiles that secure transmission of data through web sites.

Use the following tasks to create user registries, edit profiles and configurations and create SSL profiles:

- **Working with user registries**
  To secure your Catalogs, you authenticate with user registries. Perform the steps in this topic if you are creating a new user registry. In the Cloud Manager and API Manager user interfaces, 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.
- **TLS profiles**
  In Cloud Manager, TLS profiles secure transmission of data through web sites. TLS certificates guarantee that information submitted to web sites will not be stolen or tampered with. In this topic, you learn how to create TLS profiles in Cloud Manager.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with user registries

To secure your Catalogs, you authenticate with user registries. Perform the steps in this topic if you are creating a new user registry. In the Cloud Manager and API Manager user interfaces, 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.

About this task

The following user registries are available for API Connect and Cloud Manager environments:

- Local user registry
- LDAP
- URL Authentication

Note:

- Local user registry is the default user registry, and cannot be configured.
- In the Cloud Manager and API Manager, you cannot change the user registry after a user is invited to be the owner of a provider organization, even if the invitation is not yet accepted. If you subsequently need to change the user registry, you must redeploy your API Connect cloud.
- You can only use a single user registry for authenticating Cloud Manager users and a single user registry for authenticating API Manager users, regardless of user registry type (LDAP, Local User Registry or Authentication URL).
- LDAP has an option to set case sensitivity. However, the Developer Portal does not support case-sensitive usernames so that LDAP option is not supported in a Developer Portal.

The Administration user is unique and always remains in the local user registry.

Procedure

1. In the Cloud Manager page, click User Registries, then click Add.
2. To authenticate with LDAP for the Cloud Manager or API Manager, complete the following steps. To avoid problems, be sure that your LDAP usernames match the case sensitivity of your actual registry. For additional information on using LDAP, see LDAP Authentication.
   a. Select LDAP Registry from the drop-down menu.

   b. Enter values in the Display Name, Name, and optionally, Description fields
      Note: The value that you specify in the Name field can consist of the following characters:
      - Lowercase alphabetic (a through z)
      - Numerals (0 through 9)
      - Hyphen (-). A hyphen cannot be used as the first or last character.
   c. In the Hostname and Port fields, enter the required information.
   d. From the Version drop-down menu, select the version number.
   e. To protect user credentials, move the Use TLS slider to the On position. If the Use TLS option is not selected, user credentials are not protected in transit.
      Note:
      If any user registry that is created in the Cloud Manager user interface, such as an LDAP registry, is used to authenticate access to APIs and is therefore configured to be public, then the Public option must be enabled in the associated TLS profile, otherwise the TLS connection will fail.

   f. Set the case-sensitive usernames slider to match the setting of your user registry.
      Note: By default, LDAP user names are case insensitive. Set the case-sensitive usernames option only if your LDAP is case-sensitive.
   g. Select Anonymous bind or Authenticated bind.
   h. Optional: If you selected Authenticated bind and want to do a test bind & get base DN, enter the distinguished name and password of the user in the Admin DN and Password text fields.
   i. In the Base DN text field, enter the Base DN information. If you are unsure of the correct value, click Test Bind & Get Base DN to display a drop-down menu of Base DNs available for the configured LDAP server.
   j. Enter Prefix and Suffix information
   k. When you are done, click Test Configuration. If the test is successful, a confirmation message is displayed. If the test is not successful, an error message is displayed. Recheck your settings and run the test again.
   l. Click Create to create the registry.
3. To authenticate with an authentication URL for the Cloud Manager or API Manager, complete the following steps. To avoid problems, be sure your authentication URL user names match the case sensitivity of your actual registry.
   a. Select Authentication URL from the drop-down menu.
   b. Enter values in the Display Name, Name, and optionally, Description fields
      Note: The value that you specify in the Name field can consist of the following characters:
      - Lowercase alphabetic (a through z)
      - Numerals (0 through 9)
      - Hyphen (-). A hyphen cannot be used as the first or last character.
   c. In the URL text field, enter the authentication URL.
   d. To protect user credentials, move the Use TLS slider to the On position. If the Use TLS option is not selected, user credentials are not protected in transit.
   e. Set the case-sensitive usernames slider to match the setting of your user registry.
      Note: By default, Authentication URL user names are case insensitive. Select the case-sensitive usernames option only if your Authentication URL is case-sensitive.
   f. Click Create to create the registry.

Results

The Catalog and user registry information are added to API Manager.

Related concepts

- Authentication

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

TLS profiles

In Cloud Manager, TLS profiles secure transmission of data through web sites. TLS certificates guarantee that information submitted to web sites will not be stolen or tampered with. In this topic, you learn how to create TLS profiles in Cloud Manager.

Before you begin

To complete the tasks described in this topic, you must have access to the TLS Profiles page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

About this task

There are instances in API Connect where data is transmitted across an untrusted network, for example, when accessing a website, a mail server, or an LDAP server. TLS (Transport Layer Security) profiles provide public key certificates, either from a Keystore or a Trust Store, to secure communication with external services.

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.

The Cloud Manager and API Connect both support and use TLS certificates but 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 Updating a TLS profile and Generating a PKCS#12 file for Certificate Authority.

Note: If you update a TLS Server profile that is associated with a Gateway service, the updates are not automatically propagated to Gateway servers. To resynchronize your servers with the latest configurations, see Gateway resynchronization.

Procedure

Perform the following steps to create a TLS Server profile:

1. In the Cloud Manager, click TLS Profiles.
2. Click Add, and enter values in the Display Name, Name, and optionally, Description fields.
3. Select Public if the TLS profile will be associated with a registry that is used to authenticate access to APIs, otherwise the TLS connection will fail.
4. In the Present Certificate section, click the Upload Certificate icon ⊕.
5. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
   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.
6. In the Password text field, enter the password for the certificate file.
   Note: The present certificate must be password protected.
7. Click Upload.
   The certificate is populated.
8. To enable validation of the certificate presented by the connecting client, move the Request and validate the certificate against the supplied CAs in the truststore slider to the On position.
   Important: When validation is enabled, the client must send a certificate that is in the trust store, or access is denied to resources including the Cloud Manager Console. The certificate can be an intermediate CA or a root CA.
9. In the Trust Store window section, click the Upload Certificate icon ⊕.
10. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
11. In the Password text field, enter the password for the certificate file.
12. Click Upload.
    The certificate is populated.
    Note:
    - If the trust store certificate is expired, you must upload the entire certificate bundle to replace all current certificates.
    - API Connect supports only the P12 (PKCS12) and PEM certificate formats for the trust store.
    - The Trust Store can contain CA and client certificates for TLS Profiles associated with the Gateway Cluster. API Connect supports exact matching of the certificate or matching on the immediate issuers. The Cloud Manager uses the IBM Global Security Kit (GSKit) for typical certificate management tasks such as self-signed certificate generation, creation of a Certificate Authority (CA), requesting a certificate from a third-party CA and installing certificates for use in SSL protocols.
13. Optional: Repeat steps 9 to 12 to add further trust store certificates.
14. Expand the Protocols section to display the TLS versions.
15. Use the check boxes to indicate the version of the TLS protocol. The TLS protocol version determines which ciphers are available for encryption/decryption.
    V5.0.8 + Note: For instructions on enabling or disabling ciphers, see Setting the ciphers for a TLS server profile
16. V5.0.6 + To enable or disable SNI, select or clear the Use SNI check box that is found under Trust Store > Features, or V5.0.8 + Trust Store > Client Feature depending on the version of IBM® API Connect that you are using.
    V5.0.6 + Server Name Indication (SNI) is an extension to the TLS protocol. SNI is enabled by default, allowing clients to access multiple virtual domains on a single HTTPS server's IP address and port number. The TLS client injects the SNI extension with the desired host name in its initial handshake with the server. The server replies with the appropriate certificate to continue the

interaction. Servers that do not support SNI often ignore this extension, but if you encounter compatibility issues, you can disable SNI.

17. V5.0.8 + To enable the TLS server to support TLS mutual authentication, complete the following steps:
    a. In the Present Certificate section, enable Request and validate the certificate against the supplied CAs in the trust store.
    b. Provide a list of certificates that TLS server will use to verify the client certificate. This field must contain one or more certificates.

    When you define a TLS profile in the Cloud Manager user interface, two profiles are created, client and server. When the profile is used as a TLS server profile, the Present Certificate must be provided, because this is used to provide the identity of the TLS server.
18. V5.0.8 + To allow or not allow a TLS client to initiate renegotiation with this TLS profile, expand Server Feature and select or clear Allow TLS Client Renegotiation.
    TLS client renegotiation enables a client to request the establishing of a new TLS session when a TLS session has already been established. TLS client renegotiation is allowed by default in a TLS profile.

    Restriction: TLS renegotiation cannot be changed through the API Connect v5 UI. This is because only one SSL SNI Server Profile is created in DataPower Gateway for the API Connect domain. This SNI profile has the default value of allowing (not limiting) TLS Client Renegotiation. Any values for TLS Client renegotiation in the corresponding SSL Server profile (from API Connect) are overwritten by the default value from the SNI Server profile.
19. Click Save.
    Note: Once uploaded, private keys cannot be downloaded from API Connect.

- **Setting the ciphers for a TLS server profile**
  Ciphers are encryption/decryption algorithms used to secure HTTPs communication with the API Connect Management Server. The available ciphers are determined by the TLS Protocol version.
- **Generating a self-signed certificate using OpenSSL**
  OpenSSL is an open source implementation of the SSL and TLS protocols. It provides an encryption transport layer on top of 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 profile to a new gateway service**
  After you create an TLS profile, you can bind the SSL certificate to a new gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.
- **Binding a TLS profile to an existing gateway service**
  After you create a TLS profile, you can bind the SSL certificate to a previously created gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.
- **Deleting a TLS profile bound to a gateway service**
  You can delete a TLS profile that is bound to gateway service.
- **Updating a TLS 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 concepts

- Authentication

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Setting the ciphers for a TLS server profile

Ciphers are encryption/decryption algorithms used to secure HTTPs communication with the API Connect Management Server. The available ciphers are determined by the TLS Protocol version.

Before you begin

Create your TLS profiles and assign one or more protocol versions to the profiles. For instructions on creating TLS profiles, see TLS profiles. Assign a TLS profile to connections with the Cloud Manager, API Manager, and the Developer Portal REST APIs. See Specifying the cloud settings and Using the Developer Portal REST APIs.

About this task

Starting with version 5.0.8.2, you can choose the ciphers to be enabled for each TLS protocol version from a list of available ciphers. Under normal circumstances, you can accept the default enabled ciphers. These ciphers will only be used for HTTPs communication with the management server, and should not be confused with the ciphers used by the Gateway servers for API invocation or authentication.

Note: Any change in TLS Profiles and Ciphers will cause an automatic reboot of the HTTP services on all Management Servers.

Procedure

1. In Cloud Manager, chooseSettings >Ciphers.
2. View the available ciphers for each TLS protocol version. Enabled ciphers are marked with a check mark.
3. If a change is needed, add or remove check marks next to the ciphers you want to support for each version of the TLS protocol. Some available ciphers are known to be weak or insecure. Be careful when enabling new ciphers.
4. Toggle protocol versions on or off. The enabled TLS protocol versions are determined by the TLS Profile used by the cloud settings. Toggling off a TLS protocol version on the Ciphers screen will not disable that TLS protocol version. Toggling off only means that default ciphers will be used for that TLS protocol version.

Results

HTTPs communication with the Management Server will be protected by the enabled ciphers.

Related concepts

- Authentication

Related tasks

- Specifying the cloud settings
- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Generating a self-signed certificate using OpenSSL

OpenSSL is an open source implementation of the SSL and TLS protocols. It provides an encryption transport layer on top of the normal communications layer, allowing it to be intertwined with many network applications and services.

Before you begin

To complete the tasks described in this topic, you must have access to the TLS Profiles page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

About this task

The default TLS Profile in the Cloud Manager has a generic Common Name. When associating an SSL profile to a Gateway Cluster, if using the default TLS Profile, your application making API calls might fail to verify the host name it is connecting to against the certificate presented.

In this case, you can generate a new self-signed certificate that represents a Common Name your application can validate. This topic tells you how to generate self-signed SSL certificate requests 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. If you are using Dynamic DNS, your CN should have a wild-card, for example: `*.api.com.` Otherwise, 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
   ```

6. In the Cloud Manager, click TLS Profiles.
7. Click Add, and enter values in the Display Name, Name, and optionally, Description fields.
8. In the Present Certificate section, click the Upload Certificate icon ⊕.
9. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
   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. In the Password text field, enter the password for the certificate file.
    Note: The present certificate **must** be password protected.
11. Click Upload.
    The certificate is populated.
12. To validate the certificate, move the Request and validate the certificate against the supplied CAs in the truststore slider to the On position.
13. In the Trust Store section, click the Upload Certificate icon ⊕.
14. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
15. In the Password text field, enter the password for the certificate file.
16. Click Upload.
    The certificate is populated.
17. Expand the Protocols section to display the SSL and TLS versions.
18. Use the check boxes to indicate the SSL or TLS version.
19. Click Save.
20. In the Cloud Manager, click Services.
21. In the Gateway Services pane, click the Service Settings icon ⚙.
22. In the TLS Profile field, select the required profile, then click Save Service.

Related tasks

- TLS profiles
- Generating a PKCS#12 file for Certificate Authority

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

To complete the tasks described in this topic, you must have access to the TLS Profiles page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

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+BQMG1Dwl6jANBgkqhkiG9w0BAQUFADCB
   ...
   lQX7CkTJn6lAJUsyEa8H/gjVQnHp4VOLFR/dKgeVcCRvZF7Tt5AuiyHY
   -----END CERTIFICATE-----
   -----BEGIN CERTIFICATE-----
   MIIEPDCCAySgAwIBAgIQSEus8arH1xND0aJ0NUmXJTANBgkqhkiG9w0BAQUFADBv
   ...
   -----END CERTIFICATE-----
   -----BEGIN CERTIFICATE-----
   MIIENjCCAx6gAwIBAgIBATANBgkqhkiG9w0BAQUFADBvMQswCQYDVQQGEwJTRTEU
   ...
   -----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
   ```

3. In the Cloud Manager, click TLS Profiles.
4. In the Present Certificate section, click the Upload Certificate icon ⊕.
5. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
   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.
6. In the Password text field, enter the password for the certificate file.
   Note: The present certificate **must** be password protected.
7. Click Upload.
   The certificate is populated.
8. To validate the certificate, move the Request and validate the certificate against the supplied CAs in the truststore slider to the On position.
9. In the Trust Store section, click the Upload Certificate icon ⊕.
10. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
11. In the Password text field, enter the password for the certificate file.
12. Click Upload.
    The certificate is populated.
13. Expand the Protocols section to display the SSL and TLS versions.
14. Use the check boxes to indicate the SSL or TLS version.
15. Click Save.

Related tasks

- TLS profiles
- Generating a self-signed certificate using OpenSSL

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Binding a TLS profile to a new gateway service

After you create an TLS profile, you can bind the SSL certificate to a new gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.

Before you begin

To complete the tasks described in this topic, you must have access to the Services page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

About this task

Perform the following steps to bind an TLS profile to a new gateway service.

Procedure

1. In the Cloud Manager, click Services.
2. In the Gateway Services pane, click Add DataPower Gateway Service.
3. In the Service Name field, enter a name for the Gateway service.
4. In the Address field, enter the virtual IP address or host name that is to be used for inbound API calls, or of an external load balancer if one is being used.
   V5.0.6+ If you have configured your API Connect cloud to use Dynamic DNS, you can specify the same host name in the Address field for two or more Gateway services, to give the appearance that the APIs that are deployed to the separate Gateway services are on the same Gateway. For more information, see Configuring multiple Gateway services to have the same host name.
5. In the Port field, enter the API data port for inbound API calls.
6. In the Port Base field, enter the reserved port base for internal traffic. Note that the Gateway service's Port Base value should represent a set of 10 ports and should not overlap with another cluster's set of 10 ports or with any ports in use by other non-API Management applications on the DataPower appliance.
7. Optional: You can add extra enforcement capabilities to your Gateway server by uploading an IBM® DataPower® exported configuration .zip file. For more information, see Configuring your Gateway server extensions.
8. Optional: If you configure two or more Gateway servers, you can choose which load balancing option to use. In the Load balancing section, select one of the following load balancing choices.
   - Select External or no load balancer if you configure more than one Gateway server but you do not want to use the API Connect load balancing function. This option is selected by default.
     Important: You must configure your external load balancer separately. If you do not configure an external load balancer, no load balancing is done for inbound API calls.
   - Select DataPower Gateway load balancer to use the API Connect load balancing function. Enter the unique (for the local network) Load Balancing group number for the Gateway service.
     Note: The load balancing settings for internal traffic are reused to load balance inbound API calls.

   By setting the load balancing options for the Gateway service, the workload is distributed across multiple servers and ensures that the defined configuration runs as efficiently as possible. For more information, see Load balancing in IBM API Connect.
   Restriction: VLAN is not supported if you are using the Gateway as a load balancer option.
9. In the TLS Profile section, select a TLS Profile to use for the cluster. By default, the Default SSL Profile is used.
   Note: If you are binding a custom SSL certificate, you must first create a TLS Profile. For more information, see TLS profiles.
10. Click Save Service to save the service.
11. In the Gateway Services pane, click Add Server alongside the name of the service to which you want to add the Gateway server.
12. In the text fields, enter the name, address, port, user name, password, and network interface.
13. Click Create Server.

Results

The TLS Profile binds to the new gateway service.

Related tasks

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Binding a TLS profile to an existing gateway service

After you create a TLS profile, you can bind the SSL certificate to a previously created gateway service to establish an HTTPS binding. This enables you to access the service by using the HTTPS communication protocol.

Before you begin

To complete the tasks described in this topic, you must have access to the Service page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

About this task

Perform the following steps to bind a TLS profile to an existing gateway service.

Procedure

1. In the Cloud Manager, click Services.
2. Remove the servers from the service by completing the following steps for each server:
   a. Navigate to the server that you want to remove, and click the Server Actions icon ⋯, then click Delete Server.
      A window opens asking you to confirm removal of the server.
   a. To remove the server, click OK.
3. In the Gateway Services pane, click the Service Settings icon ⚙.
4. In the TLS Profile field, select a TLS profile to bind to the existing gateway service.
5. Click Save Service.
6. Re-add the Gateway servers to the Gateway service.
   For more information, see Adding a Gateway server.

Results

The TLS profile binds to the gateway service.

Related tasks

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deleting a TLS profile bound to a gateway service

You can delete a TLS profile that is bound to gateway service.

Before you begin

To complete the tasks described in this topic, you must have access to the TLS Profiles page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

About this task

Perform the following steps to delete a TLS profile bound to a gateway service.

Procedure

1. In the Cloud Manager, click TLS Profiles.
2. In the left pane, click the Delete icon alongside the TLS profile that you want to delete, and click OK.

Related tasks

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Updating a TLS 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

To complete the tasks described in this topic, you must have access to the TLS Profiles page of the Cloud Manager. For more information on which user roles have access, see Adding users and assigning roles.

CA certificate and P12 file expiration dates are displayed in the SSL Profiles page of the Cloud Manager. 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.

About this task

Perform the following steps to update a TLS profile with 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.

Note: If you update a TLS profile that is associated with a Gateway service, the updates are not automatically propagated to Gateway servers. To resynchronize your servers with the latest configurations, see Gateway resynchronization.
Important: There is no support for presenting a client certificate when a call to /v1/portal/ is made.

Procedure

1. In the Cloud Manager, click TLS Profiles.
2. In the left pane, select the TLS profile with an invalidated or expired certificate or P12 file.
   The page updates to display the TLS profile details.
3. In the Present Certificate section, click the Upload Certificate icon ⊕.
4. Click Select File, browse for the certificate file that you want to present for authentication, and click Open.
   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.

Results

The updated certificate or P12 file is added to the Cloud Manager.

Related tasks

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your Cloud Manager password

You can change your Cloud Manager password.

Procedure

To change your password, complete the following steps:

1. Log in to the Cloud Manager user interface.
2. Click the User icon then click My Account.
3. In the Change password section, enter your current password and your new password, and confirm the new password.
   The password must consist of at least six characters and must contain at least three of the following types of characters:
   - Lowercase letters
   - Uppercase letters
   - Numbers
   - Special characters; the following characters are allowed:

     ```
     !
     \"
     #
     $
     %
     &
     \
     (
     )
     *
     +
     ,(comma)
     - (dash/hyphen)
     .(period)
     /
     :(colon)
     ;(semi-colon)
     <
     =
     >
     ?
     @
     [
     \\
     ]
     ^
     _ (underscore)
     ` (back quote/grave accent)
     {
     | (pipe)
     ```

```
}
~ (tilde)
```

The password cannot contain more than two consecutive repeating characters.

4. Click Save.

Results

Your password is changed.

Related concepts

- [Configuring and managing your server environment]

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See [support policy] for details.
For a more recent version, see the [IBM API Connect 10.0.5.x product documentation].

Changing the expiration settings for Cloud Manager user accounts

You can change the expiration settings for your Cloud Manager user accounts by using REST API calls.

About this task

You can change the following settings:

- Account activation expiry time: the time, in minutes, after which a new account expires if the user does not activate it. The default value is 1440 minutes (24 hours).
- Password reset token expiry time: the time, in minutes, after which a password reset token expires if the requesting user does not reset the password. The default value is 1440 minutes (24 hours).

V5.0.3 and earlier You can change a **single** setting that controls **both** the account activation expiry time **and** the password reset token expiry time.

V5.0.4 + You can separately change the account activation expiry time and the password reset token expiry time.

In these instructions, the **curl** command is used to demonstrate the REST API calls. The **curl** commands use the following variables:

- *userID* and *password*: the login credentials for a Cloud Manager user account that has been assigned one of the following roles:
  - Cloud Owner
  - System
  - Cloud Administrator
  - Topology Administrator
- *hostname*: the host name or IP address of your IBM® API Connect Management cluster.

Procedure

To change the expiration settings for your Cloud Manager user accounts, complete the following steps depending on the version of IBM API Connect you are using:

- V5.0.3 and earlier Complete the following steps:
  1. Obtain the current cloud settings by using the following command:

     ```
     curl -k -u "cmc/userID:password" -H "accept: application/json" "https://hostname:443/v1/cloud"
     ```

     Note: You must use the settings returned by this REST API call in the following step.
  2. Modify the password expiration settings by using the following command:

     ```
     curl -k -u "cmc/userID:password" -H "content-type: application/json" -X PUT 
     "https://hostname:443/v1/cloud" -d
     '{

       ... the other settings obtained from GET /cloud response in the previous step  ...
     ```

```
  "userAccountSettings": {
    "userActivationExpiration": value_in_minutes
  }
}'
```

where **userActivationExpiration** specifies both the account activation expiry time and the password reset token expiry time.

- V5.0.4 + Complete the following steps:
  1. Obtain the current cloud settings by using the following command:

     ```
     curl -k -u "cmc/userID:password" -H "accept: application/json" "https://hostname:443/v1/cloud"
     ```

     Note: You must use the settings returned by this REST API call in the following step.
  2. Modify the password expiration settings by using the following command:

     ```
     curl -k -u "cmc/userID:password" -H "content-type: application/json" -X PUT
     "https://hostname:443/v1/cloud" -d
     '{

       .... the other settings obtained from GET /cloud response in the previous step ...

       "userAccountSettings": {
         "userActivationExpiration": value_in_minutes,
         "userPasswordResetExpiration": value_in_minutes
       }
     }'
     ```

     where **userActivationExpiration** specifies the account activation expiry time and **userPasswordResetExpiration** specifies the password reset token expiry time.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Troubleshooting

Troubleshooting topics for the Cloud Manager.

- **Graphs displaying no values**
  In the Services page, the Gateway server statistics are not displayed in the graphs.
- **An error is seen when you add a Gateway server**
  After a Gateway server is removed from an API Connect on-premises cloud, when the administrator tries to add the same Gateway server again an error is displayed.
- **AppOpt license activation fails**
  For API Connect versions lower then 5.0.8.1, the AppOpt license activation fails and requires a script from IBM Support.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Graphs displaying no values

In the Services page, the Gateway server statistics are not displayed in the graphs.

Symptom
    When you look at the monitoring statistics for the Gateway servers in the Cloud Manager, no statistics are displayed.

Solution

Ensure that the statistics service is enabled for the IBM® DataPower® appliances in the DataPower default domain. The statistics service setting is disabled by default.
To enable the statistics service, you can enter the following commands by using the Cloud Manager command-line interface (CLI):

- **config**
- **statistics**
- **write mem**

If you use the IBM DataPower WebGUI, complete the following steps:

1. Log in to the default domain.
2. Select Administration > Device > Statistics Settings and change the Administrative State to enabled.
3. Click Apply and then click Save Config.

For more information, see the IBM WebSphere® DataPower Version 6.0.1 documentation , and search for *Statistics* or *Enabling statistics*.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

An error is seen when you add a Gateway server

After a Gateway server is removed from an API Connect on-premises cloud, when the administrator tries to add the same Gateway server again an error is displayed.

Description

After a Gateway server is removed from an API Connect cloud, some scenarios exist where the configuration of the Gateway server might not be successfully cleaned. If the administrator tries to add the same Gateway server back in to the cloud, the following error is displayed:

```
...Error adding loadbalancer for API calls
```

In particular, this error can occur if the Management server for the cloud is deleted and re-created without first removing the Gateway server, that is, the Gateway server was not removed from the cloud.

Solution

This issue can be resolved by completing the following steps:

1. Manually remove the load balancer configuration from the Gateway server (the DataPower® appliance), by using one of the following methods:
   - In the command-line interface, run the following command:

     ```
     ssh datapower_hostname
     login
     co
     int eth0
     no standby
     exit
     write mem
     ```

     Where *login* is your login credentials, *datapower_hostname* is your DataPower host name, and your DataPower appliance is configured to use the network interface *eth0*.
   - In the DataPower WebGUI:
     - In the navigation tree, open the Network and Interface folders.
     - Click Ethernet Interface.
     - In the table, click the relevant interface row.
     - Click the Standby Control tab.
     - To delete the load balancing group, click X in the relevant table row.
     - Click Apply.

- In the page header, click Save Config.

2. To prevent any port conflicts, remove all of the IBM API Connect domains from the Gateway server (the DataPower appliance). Otherwise, adding the Gateway server appears to be successful, but when APIs that are created by using the readded Management server are accessed, you receive 404 errors. The IBM API Connect related domains are of the format: APIMgmt_*10_characters*, where *10_characters* is 10 alphanumeric characters.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

AppOpt license activation fails

For API Connect versions lower then 5.0.8.1, the AppOpt license activation fails and requires a script from IBM Support.

Symptom
Error occurs in AppOpt license activation.

Solution
For API Connect versions lower then 5.0.8.1: When adding a DataPower to a Gateway server, the AppOpt (AO) license will not be activated automatically on DataPower. To activate the AO license on DataPower, contact IBM Support for a script that will complete the operation. Another option is to upgrade to API Connect version 5.0.8.1, which does activate the AO license on DataPower.

Related concepts

- Configuring and managing your server environment

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Developing your V5 APIs and applications

You develop APIs and LoopBack® applications by using the IBM® API Connect V5 developer toolkit.

The IBM API Connect developer toolkit provides both the API Designer user interface and a command line interface that you can use to develop APIs and LoopBack applications and publish them to IBM 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 IBM API Connect by using either the API Designer 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.
- **Creating API definitions by using the API Designer**
  The API Designer is a graphical user interface within the developer toolkit and provides functions for the creation and configuration of API definitions, running offline.
- **Configuring API security by using the API Designer**
  You configure security for an API by creating one or more security definitions 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 the API Designer**
  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. You use the API Designer to create, edit, and stage your Product, and the API Manager to manage the lifecycle of your Product.

- **Creating and validating API and Product definitions by using the command line interface**
  The developer toolkit 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.
- **Developer toolkit tutorials**
  Tutorials for using the developer toolkit. Ensure you select the correct tutorials for the version of IBM API Connect that you are using.
- **Reference**
  Reference information for the API Designer component in API Connect.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 either from npm or from a Management server in your IBM API Connect cloud.
- **Getting started with the developer toolkit command-line tool**
  The IBM API Connect developer toolkit provides a command-line tool, `apic`, for creating and testing APIs that you can then run, manage, and secure with IBM API Connect. You can also use the command-line tool to script tasks such as continuous integration and delivery.
- **Toolkit command summary**
  A summary of the core commands in the IBM API Connect developer toolkit.
- V5.0.3 + **Creating new projects with API Designer**
  You can create new LoopBack and OpenAPI (Swagger 2.0) projects with the API Designer.
- V5.0.2 + **Creating a database schema from models**
  Database schemas based on your models can be created and updated through API Designer, for data sources that support the models. This enables you to develop your models first, and create (and update) your database schema to match them. This process is sometimes referred to as "auto-migration" and is supported by data source connectors for MongoDB, MySQL, Oracle, PostgreSQL, and SQL Server.
- V5.0.2 + **Discovering models from relational databases**
  You can use API Designer to create models from existing database tables. This process is called *discovery* and is supported by data source connectors for: MySQL, Oracle, PostgreSQL, and SQL Server.
- V5.0.2 + **Adding an existing project to API Designer**
  You can add existing LoopBack and OpenAPI (Swagger 2.0) projects from the local file system and then edit them withAPI Designer.
- **Publishing a LoopBack application through the API Designer**
  Publish a LoopBack application to an API Connect collective to make the application available for use by your API definitions. You can also publish a Product to a Catalog at the same time. V5.0.5 + The syndication feature in IBM API Connect means that you can also publish a Product to a Space in a Catalog.
- V5.0.7 and earlier **Viewing application performance metrics**
  IBM API Connect provides two ways to view application performance metrics for Node.js applications running locally: V5.0.7 + the built-in metrics dashboard and sending the data to third-party consoles or log files.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing the toolkit

You can install the toolkit either from npm or from a Management server in your IBM® API Connect cloud.

Before you begin

Features of the toolkit include the default ability for local testing of APIs with a DataPower® Gateway Docker container. When Docker and Docker Compose are present and functioning normally on your system, the DataPower image is downloaded from Docker Hub. For more information, see: Testing APIs with the IBM DataPower Gateway.

The following steps create a software environment that is ready to install the IBM API Connect toolkit.

Note: On Windows, use the Windows command shell run as an administrator to enter Node or npm commands, instead of Powershell or Cygwin (the Windows bash shell emulator).

1. Install Community Node.js distribution version 10, or a higher 10.x.x version. For details of supported Node.js versions, see Detailed System Requirements, then click the Prerequisites tab.
2. Ensure that `node` is in your PATH.
3. Use the `npm -v` command to check the version of npm and ensure that it shows 10.x.x, or higher.

About this task

You install the toolkit by using the `npm` command that is installed as part of Node.js. Installing the toolkit installs:

- API Connect command-line tool, `apic`.
- API Connect API Designer visual tool.
- API Connect Micro Gateway.

You can also uninstall the toolkit and display the version number of the currently installed toolkit.
Use the version of the toolkit that corresponds to the version of your API Connect Management Server. To find the appropriate toolkit version, use the `npm dist-tag ls apiconnect` command. For example:

```
$ npm dist-tag ls apiconnect
1.0.3.0: 1.0.3
apic-v5.0.1.0: 2.0.18
apic-v5.0.2.1: 2.1.19
apic-v5.0.3.0-iFix1: 2.2.12
apic-v5.0.3.0-iFix2: 2.2.14
apic-v5.0.3.0-iFix4: 2.2.17
apic-v5.0.3.0: 2.2.9
apic-v5.0.4.0-iFix2: 2.3.10
apic-v5.0.4.0: 2.3.6
apic-v5.0.5.0: 2.4.11
apic-v5.0.6.0: 2.5.8
apic-v5.0.6.1: 2.5.17
apic-v5.0.6.2: 2.5.21
apic-v5.0.6.3: 2.5.33
apic-v5.0.6.4: 2.5.40
apic-v5.0.6.5: 2.5.52
apic-v5.0.6.6: 2.5.80
apic-v5.0.7.0: 2.6.2
apic-v5.0.7.1: 2.6.55
apic-v5.0.7.2: 2.6.71
apic-v5.0.8.0: 2.7.30
apic-v5.0.8.1: 2.7.62
apic-v5.0.8.2: 2.7.111
apic-v5.0.8.3: 2.7.209
apic-v5.0.8.4-iFix: 2.8.39
apic-v5.0.8.4: 2.8.14
apic-v5.0.8.5: 3.0.17
apic-v5.0.8.6: 4.0.x
latest: 4.0.x
v5.0.0.1: 1.0.3
```

Then, to install a specific version of toolkit use the command `npm install apiconnect@<version>`. For example, if you have API Connect v5.0.6.5, use the command:

```
npm install apiconnect@3.0.17
```

Procedure

Note: Do not use the npm configuration setting `engine-strict` (or use the `--engine-strict` option) since it will prevent installation from completing.
During installation, you may see errors from the `node-gyp` module, but because these errors are from an optional dependency, the installation should finish successfully.

You may also see npm warnings of `Possible EventEmitter memory leak detected`, but these are spurious and do not indicate a memory leak or any other issue.

To install the toolkit, complete the following steps:

- If you are not using trusted certificates, enter the following command:

  ```
  npm config -g set strict-ssl false
  ```

- If you are using a proxy server, enter the following commands:

  ```
  npm config set proxy http://proxy_address:port
  npm config set https-proxy http://proxy_address:port
  ```

  where *proxy_address* is the host name or IP address of the proxy server, and *port* is the port number.
- You are going to install the toolkit globally (**-g**), because it includes a command line interface. You can use **npm config get prefix -g** to learn the location of the global installation directory. Ensure that you are installing with sufficient permissions to write system files. Install the toolkit in either of the following ways:
  - To install the toolkit from npm, enter the following command:

    ```
    npm install -g apiconnect
    ```

  - To install the toolkit from a Management server in your IBM API Connect cloud, enter the following command:

    ```
    npm install -g --unsafe-perm https://appliance/packages/apiconnect
    ```

    Where *appliance* is the host name or IP address of the Management server appliance.
  - To install the toolkit locally from the toolkit tarball, download the tarball .tgz file from IBM Fix Central, then enter the following command:

    ```
    npm  i -g apiconnect-version.tgz
    ```

    or if you installing from root, enter the following command:

    ```
    npm  i -g apiconnect-version.tgz --unsafe-perm
    ```

- To display the version number of the currently installed toolkit, enter the following command:

  ```
  apic -v
  ```

What to do next

Obtain an IBM Cloud API Key for authenticating with the toolkit.

1. Obtain the API Key by completing the following steps:
   a. Browse to the IBM Cloud Identity and Access Management page.
   b. In the navigation list, click IBM Cloud API Keys.
   c. On the IBM Cloud API Keys page, click Create an IBM Cloud API key.
   d. In the Create API key dialog box, provide a Name and a Description for your new key, and then click Create.
   e. In the Create API key message box, either Copy or Download the key to make sure you save a copy. If you save the file, it is named apiKey.json.
   Attention: Do not navigate away from the page until you have saved the key. Once you lose this message, you cannot obtain a copy of the key (in that case, delete the key and create a new one).
2. Log in to the toolkit with the new API Key and the following command:

   ```
   apic login --server=Cloud.apiconnect.ibmcloud.com --apikey=Your_new_api_key
   ```

   In the command, ***`Cloud`*** is the hosted instance of API Connect you are connecting to; for example, **`us`** for US South, or **`eu-de`** for Frankfurt. In Reserved Instance, this is the host name that you use to access API Manager (begins with **`mgr-`**).

Uninstalling the toolkit

Before you begin

Before uninstalling the toolkit, stop any apps that are running locally by entering the command:

```
apic stop --all
```

Note: Configuration settings for the toolkit are stored in the `home_dir\.apiconnect` directory, where *home_dir* is the home directory of the user account under which the toolkit was installed. When uninstalling, you have the option to either keep or delete the configuration settings. The default behavior is to delete the `home_dir\.apiconnect` directory. Use the `--no-config-clear` command to preserve the configuration settings.

Procedure

1. Uninstall using npm:

   ```
   npm uninstall -g apiconnect
   ```

   By default, this command removes the toolkit configuration settings, as described above. To keep toolkit configuration settings, instead enter the command:

   ```
   npm uninstall -g apiconnect --no-config-clear
   ```

   To uninstall the toolkit, clear your npm cache, and remove the toolkit configuration, instead enter the command:

   ```
   npm uninstall -g apiconnect
   ```

   To uninstall the toolkit, clear your npm cache, and keep the toolkit configuration, instead enter the command:

   ```
   npm uninstall -g apiconnect --no-config-clear
   ```

   To verify that the cache is consistent after uninstalling the toolkit, enter the following command:

   ```
   npm cache verify
   ```

2. On Windows, delete all files whose names begin with `npm-` in `C:\Users\`*`username`*`\AppData\Local\Temp`

Updating your toolkit installation

Before you begin

Before updating your toolkit installation, be sure to stop any apps that are running locally by entering the command:

```
apic stop --all
```

Procedure

1. Uninstall the toolkit, as described in Uninstalling the toolkit.
2. Reinstall the toolkit by entering this command:

   ```
   npm install -g apiconnect
   ```

What to do next

If you previously used a username and password to authenticate in the API Connect Developer Toolkit, then you must start using API Key authentication instead.

1. Obtain the API Key by completing the following steps:
   a. Browse to the IBM Cloud Identity and Access Management page.
   b. In the navigation list, click IBM Cloud API Keys.
   c. On the IBM Cloud API Keys page, click Create an IBM Cloud API key.
   d. In the Create API key dialog box, provide a Name and a Description for your new key, and then click Create.
   e. In the Create API key message box, either Copy or Download the key to make sure you save a copy. If you save the file, it is named apiKey.json.
   Attention: Do not navigate away from the page until you have saved the key. Once you lose this message, you cannot obtain a copy of the key (in that case, delete the key and create a new one).
2. Log in to the toolkit with the new API Key and the following command:

   ```
   apic login --server=Cloud.apiconnect.ibmcloud.com --apikey=Your_new_api_key
   ```

   In the command, ***`Cloud`*** is the hosted instance of API Connect you are connecting to; for example, `us` for US South, or `eu-de` for Frankfurt. In Reserved Instance, this is the host name that you use to access API Manager (begins with `mgr-`).

Related concepts

- Developer toolkit tutorials

Related information

- Troubleshooting the Micro Gateway

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Getting started with the developer toolkit command-line tool

The IBM® API Connect developer toolkit provides a command-line tool, `apic`, for creating and testing APIs that you can then run, manage, and secure with IBM API Connect. 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:

- **Overview of the command-line tool**
  The developer toolkit provides commands for working with APIs, Products, and applications, and for launching the API Designer graphical tool.
- **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.
- **Running APIs locally**
  The developer toolkit includes a Micro Gateway and a Node.js process manager that you use to test APIs and applications. You can optionally set a IBM DataPower® Gateway Docker container to test more complex features.
- **Publishing APIs and applications**
  To publish APIs and applications by using the developer toolkit, 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. When you publish LoopBack® projects, you must publish both the APIs and the associated applications so the projects can be run.
- **Managing API Products**
  Use the `apic products` and `apic apis` commands to manage Products and APIs that have been published to API Connect Catalogs. V5.0.5+ 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Overview of the command-line tool

The developer toolkit provides commands for working with APIs, Products, and applications, and for launching the API Designer graphical tool.

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 mgmnthost.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`.

Note: The `create` command has a slightly different syntax:

```
apic create --type sub-command [options]
```

Use the `-h` or `--help` option to view command help. Some of the commands in the following tables are annotated with **Stability: prototype**, which indicates that IBM® is in the process of collecting customer feedback on the commands and you should not use them in production scripts.

Viewing version information

Display the version of the command-line tool by entering the command: `apic --version` or `apic -v`. Display extended version information, including the versions of all the command-line tool plug-in modules, by entering the command: `apic --ext-version`.

Launching the API Designer

The API Designer graphical tool provides most of the capabilities of the command-line tool but with a visual interface.

To launch the API Designer, enter the command `apic edit`.

Creating and validating artifacts

Table 1. Summary of general-purpose apic commands

Command	Description	Sub-commands
`apic config`	List and manage configuration variables. For more information, see Using configuration variables. With no sub-command, lists values of defined configuration variables.	• `get` - Get a configuration variable. • `list` - List configuration variables (default). • `set` - Set configuration variables. • `delete` - Delete a configuration variable. • `clear` - Delete all configuration variables
`apic create`	Create project artifacts.	• `--type api` - Create an API. • `--type api --wsdl *filename*` - Create a SOAP API 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. • `--type product` - Create a Product. • `--type model` - Create a LoopBack® model. • `--type datasource` - Create a LoopBack data source. V5.0.2 + Note: You can create an API or Product from an OpenAPI (Swagger 2.0) template file by using the `--template *template-name*` option.
`apic edit`	Run the API Designer and open in default web browser.	None
V5.0.7 + `apic extensions`	Manage OpenAPI (Swagger 2.0) extensions in a catalog. With no sub-command, lists the extensions in the production catalog. **Stability: prototype**	• `clone` - Pull all extensions from a catalog. • `delete` - Delete an extension in a catalog. • `get` - Get information on an extension in a catalog. • `list` - List all extensions published to a catalog (default). • `publish` - Publish an extension to a catalog. • `pull` - Pull an extension from a catalog.

Command	Description	Sub-commands
`apic loopback`	Create LoopBack project and project artifacts. With no sub-command, creates a new LoopBack project. All of these commands are **Stability: prototype**, except for `loopback:app`.	• `acl` - Add access control list specification. • `app` - Create a new LoopBack project (default). • `boot-script` - Add boot script. • `export-api-def` - Generate OpenAPI (Swagger 2.0) definitions from models. • `middleware` - Add middleware function. • `property` - Add property to existing model. • `refresh` - Update Product and API definition from a model. • `relation` - Add relation between models. • `remote-method` - Add remote method • `swagger` - Generate LoopBack project from an OpenAPI (Swagger 2.0) definition.
V5.0.7 and earlier `apic microgateway`	Create Micro Gateway applications. Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.	None
`apic validate`	Validate API or Product definition YAML file.	None

Managing and testing applications

Table 2. Summary of apic commands to manage and test applications locally

Command	Description	Sub-commands
`apic logs`	Display server logs continuously to console.	None
V5.0.7 and earlier `apic props`	List and manage service properties for a LoopBack application running locally. With no sub-command, lists values of defined service properties.	• `get` - Get a property value. • `list` - List all property values (default). • `set` - Set or update property values. • `delete` - Delete a service property. • `clear` - Delete all service properties.
V5.0.7 and earlier `apic services`	List and manage services. With no sub-command, lists all services that are currently executing.	• `list` - List all services currently executing (default) • `start` - Start a service. • `stop` - Stop a service. • `logs` - Display logs to console. • `get` - Get information about a service.
V5.0.7 and earlier `apic start`	If run in LoopBack project directory, start the LoopBack application; otherwise, start the Micro Gateway. Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.	None
V5.0.7 and earlier `apic stop`	Stop the specified service or all services.	None

Command	Description	Sub-commands
V5.0.6 + `apic explore`	Opens the API Explore tool. Shows the operations, definitions, and documentation for all of the APIs that are contained in the project directory. To specify a particular API, include the name of the API in the command, for example: `apic explore` *`apiname`*`.yaml`	Option: • `-e, --external` Opens the Explore tool on 0.0.0.0 instead of the default 127.0.0.1. This option binds the server to all IP addresses on the machine, and makes the tool accessible on the wider network.

Managing artifacts and publishing to the cloud

Table 3. Summary of apic commands to manage APIs, apps, Products, and Catalogs, and publish them to the cloud

Command	Description	Sub-commands
`apic apps`	List, manage, or publish applications. Default sub-command is `list`.	• `get` - Get information about an application. • `build` - Build an application. • `list` - List provider apps contained in organizations of which the currently-authenticated user is a member (default sub-command). • `publish` - Publish an application to a provider app. • `set` - Update an application.
`apic apis`	List and manage APIs in a Catalog or Space.**Stability: prototype** Default sub-command is `list`.	• `clone` - Pull all APIs from a Catalog or Space - Build an application. • `get` - Get information on an API in a Catalog or Space. V5.0.7 + The output separately lists the production and development endpoints at which an application can call the API. For more information, see Managing the application lifecycle. • `list` - List APIs in a Catalog or Space (default sub-command). • `pull` - Pull an API from a Catalog or Space. • `set` - Update an API in a Catalog or Space.
`apic catalogs`	List and manage Catalogs.	• `catalogs:create` Create a Catalog in an organization • `catalogs:delete` Delete a Catalog in an organization • `catalogs:get` Get information on a Catalog in an organization • `catalogs:list` - List Catalogs contained in organizations of which the currently-authenticated user is a member (default sub-command). • V5.0.8 + `catalogs:transfer` - Beginning with Version 5.0.8.4: Transfers the ownership of a Catalog to another user.
V5.0.4 + `apic devapps`	List and get information about consumer applications. Default sub-command is `list`.	• `get` - Get information about a consumer application. • `list` - List consumer apps in an organization and a Catalog.

Command	Description	Sub-commands
`apic drafts`	List and manage APIs and Products in drafts. Default sub-command is `list`.	• `clear` - Delete all API and Product definitions in drafts. • `clone` - Pull all API and Product definitions from drafts. • `delete` - Delete an API or Product definition from drafts. • `list` - List APIs and Products in drafts (default sub-command). • `get` - Get information on an API or Product definition in drafts. • `publish` - Stage and publish a Product and its referenced APIs in drafts to a Catalog. • `pull` - Pull API or Product definitions from drafts. • `push` - Push local API or product definitions to drafts.
`apic login`	Log in to API Manager.	None. Specify server and credentials with the required flags: • `-u, --username` *user_name* • `-p, --password` *password* • `-s, --server` *mgmt_server*. V5.0.7+ You can append the port number to the server name if it is not the default value of 443.
`apic logout`	Log out from API Manager.	None. Specify server with the required flag: • `-s, --server` *mgmt_server*. V5.0.7+ You can append the port number to the server name if it is not the default value of 443.
V5.0.5+ `apic members`	List members in an organization. **Stability: prototype**	None
V5.0.4+ `apic orgs:get`	Display information on a consumer or provider organization. Use `--type provider|consumer` to specify either provider or consumer organization.	• `get` - Display information on a consumer or provider organization.
`apic organizations` V5.0.4+ `apic orgs`	List and get information about organizations. Default sub-command is `list`. Note: The `organizations` command available in early releases is deprecated in favor of `orgs`.	• `get` - Get information on a provider organization. • `list` - List organizations of which the currently-authenticated user is a member. On IBM Cloud, lists only organizations where the API Management tile is provisioned and added.
`apic policies`	List and manage policies in a Catalog. Default sub-command is `list`.	• `clone` - Pull all policies from a Catalog. • `delete` - Delete a policy from a Catalog. • `get` - Get information on a policy in a Catalog. • `list` - List policies in a Catalog (default sub-command). • `publish` - Publish a policy to a Catalog. • `pull` - Pull a policy from a Catalog.
`apic publish`	Publish a Product and its referenced APIs to a Catalog. V5.0.5+ If Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog. To publish to a Space, the `--scope space` option must be included in the publish command.	None. This is an alias for `apic products:publish`.

Command	Description	Sub-commands
`apic products`	List and manage Products in a Catalog. Default sub-command is `list`.	- `clear` - Clear all Products in a Catalog. **Stability: prototype** - `clone` - Pull all Product definitions from a Catalog. **Stability: prototype** - `delete` - Delete a Product in a Catalog. - `get` - Get information on a Product in a Catalog. - `list` - List Products in a Catalog (default sub-command). - `publish` - Stage and publish Product and referenced APIs to a Catalog. - `pull` - Pull a Product from a Catalog. - V5.0.4+ `replace` - Replace a Product in a Catalog with another Product . **Stability: prototype** - `set` - Update a Product in a Catalog. - V5.0.4+ `set-migration-target` - Set migration target of a Product in a Catalog with another Product. **Stability: prototype** - V5.0.4+ `supersede` - Supersede a Product in a Catalog with another Product. **Stability: prototype**
V5.0.5+ `apic spaces`	List and manage Spaces contained in a Catalog. Default sub-command is `list`.	- `spaces:create` Create a Space in a Catalog - `spaces:delete` Delete a Space in a Catalog. - `spaces:get` Get information on a Space in a Catalog. - `spaces:list` List Spaces contained in a Catalog (default sub-command). - `spaces:set` Set information on a Space in a Catalog.
V5.0.5+ `apic subscriptions`	List and manage subscriptions in a Product or a Catalog. Default sub-command is `list`.	- `subscriptions:get` - Get information on a subscription in an app. - `subscriptions:list` List Spaces contained in a Catalog (default sub-command). **Stability: prototype**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

The developer toolkit provides an integrated development environment for developing APIs and applications that use the LoopBack® framework. To create a new LoopBack project, use the command **`apic loopback`**; then use the **`apic edit`** command to edit the project in the API Designer.

LoopBack is a high-performance Node.js interaction-tier framework for APIs and micro-services. However, you can also use the developer toolkit to create language-independent APIs by using OpenAPI (Swagger 2.0) 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.

When using LoopBack projects, you can publish the API to an API Connect Catalog that provides socialization via the developer portal and policy enforcement via the gateway, and the application to an API Connect App that provides Node.js runtime capability. To proxy to existing backend services or develop applications using other languages or frameworks, use OpenAPI projects that support publishing the OpenAPI (Swagger 2.0) definitions to API Connect Catalogs.

Creating development artifact definitions

Use the `apic create` command to create development artifacts, by using the following commands:

Command	Description
`apic create --type api`	Create an OpenAPI (Swagger 2.0) definition.
`apic create --type api --wsdl` ***`filename`***	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 Using an options file when importing a WSDL service.
`apic create --type product`	Create an API Product definition.
`apic create --type model`	Create a LoopBack model to a LoopBack project: (Use in a LoopBack application project).
`apic create --type datasource`	Create a LoopBack data source to a LoopBack project: (Use in a LoopBack application project).

Note: You can create an API or Product from an OpenAPI (Swagger 2.0) 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 --type api --title Routes
apic create --type product --title "Climb On"
```

You can also create the API and Product definitions at the same time:

```
apic create --type api --title Routes --product "Climb On"
apic create --type 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 --type api --title Routes
apic create --type api --title Ascents
apic create --type product --title "Climb On" --apis "routes.yaml ascents.yaml"
```

Validating development artifact definitions

After you edit development artifacts or right 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 APIs created above
apic validate climb-on.yaml --product-only              # Validate the Product only (do not validate the
referenced APIs)
```

V5.0.5 + Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the API definition is validated by the **apic validate** command. For more information, see V5.0.5 + Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.

Developing LoopBack applications

After you've created a LoopBack application with the `apic loopback` command, you can add additional functionality to the application with the commands listed in the following table . Run these commands from the application's project root directory.

Command	Description
`apic loopback:boot-script`	Create a new boot script.
`apic loopback:acl`	Define an access control list for accessing LoopBack models.
`apic loopback:middleware`	Define and register Express middleware to define the phase of execution.
`apic loopback:property`	Add additional properties to a Loopback model.
`apic loopback:remote-method`	Add remote methods to a LoopBack model.
`apic loopback:relation`	Add relationships between LoopBack models.
`apic loopback:export-api-def`	Export an OpenAPI (Swagger 2.0) and a Product definition from LoopBack models.
`apic loopback:swagger`	Generate a LoopBack application project from an OpenAPI (Swagger 2.0) 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Running APIs locally

The developer toolkit includes a Micro Gateway and a Node.js process manager that you use to test APIs and applications. You can optionally set a IBM® DataPower® Gateway Docker container to test more complex features.

Important: From IBM API Connect Version 5.0.8.7, some commands are no longer supported by the API Connect toolkit. Such commands are indicated on this page by the V5.0.7 and earlier icon.
Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
V5.0.7 and earlier

Creating and running a service

Use the **`services`** command to manage running processes for testing APIs and applications. By default, the actions for the **`services`** command work on the project or directory where you executed the command, enabling you to manage services for multiple projects concurrently and independently.

Some of the most common actions with the **`services`** command are:

- **`apic services`** - List the local running services (alias for **`services:list`**).
- **`apic start`** - Start the local services (alias for **`services:start`**).
- **`apic stop`** - Stop the local services (alias for **`services:stop`**).
- **`apic stop --all`** - Stop all services across all projects.

Note: When you run a project locally, it is also known as a "service."
V5.0.7+ If you want to use the IBM DataPower Gateway Docker image for testing your API locally, see Testing APIs with the IBM DataPower Gateway for details.

The following procedure is a typical workflow to create a LoopBack® project, start the Micro Gateway and the Node.js server running the LoopBack application, and test the API using the endpoint exposed by the Micro Gateway.

1. Create a LoopBack application project called `climbon` in a climbon directory:

   ```
   apic loopback --name climbon
   ```

2. Update the API and application development artifacts.
3. Test the project (service) locally. V5.0.7 and earlier

   ```
   apic start
   ```

   V5.0.7+ Note: If the security policy provider is set to DataPower Gateway, this command will attempt to download a Docker container. For more details, see Testing APIs with the IBM DataPower Gateway.
4. Update the API and application development artifacts as required.
5. Restart the services so they run with the latest artifact definitions, and re-test. V5.0.7 and earlier

   ```
   apic stop
   apic start
   ```

V5.0.6+ Note: You can also use the API Explore tool to test and explore your APIs. Ensure that your local test servers are running, then run the command **apic explore**. The API Explore tool opens, and shows the operations, definitions, and documentation for all of the APIs that are contained in your project directory. You can specify a single API to explore by specifying the name of the API in the command. The left pane of the Explore window can be used to select an operation to test. The center pane displays summary information about the endpoint, including its parameters, model instance data, and response codes, and the right pane provides template code to call the endpoint. Including the option **`-e`** or **`--external`** in the command, opens the Explore tool on 0.0.0.0 instead of the default 127.0.0.1. This option binds the server to all IP addresses on the machine, and makes the tool accessible on the wider network.

Viewing logs

When running services and testing APIs and applications, it's often useful to view the Micro Gateway and Node.js LoopBack logs by using the **`apic logs`** command. For example:

Command	Description
`apic logs`	View the logs for the default service (alias for services:logs)
`apic logs --service` ***`service_name`***	Use apic services to list the service names.

Set the log-level service property to determine the level of logging detail for the local application and Micro Gateway. The value must be one of:

- `debug` - Include all messages in the log. This is the most verbose log level, and is useful for debugging.
- `info` - Include messages of "info" level and more severe in the log.
- `warning` - Include only messages of "warning" level or more severe in the log.
- `error` - Include only messages of "error" level and more severe in the log.
- `fatal` - Include only messages of "fatal" (the most severe) level in the log.

- V5.0.7+ **Testing APIs with the IBM DataPower Gateway**
  Using the API Connect Toolkit with IBM DataPower Gateway unlocks the full set of security and policy capabilities for local testing.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Testing APIs with the IBM DataPower Gateway

Using the API Connect Toolkit with IBM® DataPower® Gateway unlocks the full set of security and policy capabilities for local testing.

Some key benefits of this feature include: Develop and test APIs that use the full complement of IBM API Connect Assembly policies; saved changes are instantly synchronized with the Gateway for rapid testing and feedback; test product and plan level concepts such as rate-limiting; the reserved Catalog name `apic-dev` is available to provide substitute assembly properties at runtime; DataPower error logging is integrated into the API Designer logging console; Request/Response logging is also available from the logging console with latency data.

Before you begin

You must have the following installed and functioning properly to use this feature:

- Docker v1.13 or higher.
- Docker Compose v1.11 or higher.
- On the Windows platform, Windows 10 Professional and Docker for Windows are required.
- On the Mac platform, Docker for Mac is required.
- On the Linux® platform, install Docker by following the Docker installation instructions for your particular distribution.

Note:

- This feature is not available with IBM API Connect Version 5.0.8.7.
- If you are using a version of Linux that includes Security-Enhanced Linux (SELinux) and you find that, after completing the steps described in this topic, you are unable to call your API, enter the command **setenforce 0** to run SELinux in permissive mode, then retry. To verify that SE Linux is running in permissive mode, enter the command **sestatus** and check the **Current mode** property.
- When you use Docker for Windows with Windows 10, you must adjust the Docker software settings to share the C: drive. If you choose to install Docker on Windows 7, you must use the Docker Toolbox, which includes Oracle VirtualBox. Under this configuration, Docker containers run in a VM hosted by VirtualBox. Ensure that you increase the VirtualBox resource settings to provide at least 4 GB RAM and 2 CPUs to the Docker VM (usually named `default`).

About this task

The Toolkit determines which gateway type to run based on the metadata that is stored in the current project's API definition YAML file based on OpenAPI (Swagger 2.0). You can manually edit this file and insert `gateway: datapower-gateway` as a child of the `x-ibm-configuration:` block. Alternatively you can select the gateway type with the API Designer Assembly view, selecting between the Micro Gateway or DataPower provider and press save. The following entry in your file indicates a selection of the IBM DataPower Gateway for API testing:

```
x-ibm-configuration:
  # ... other configuration options
  gateway: datapower-gateway
```

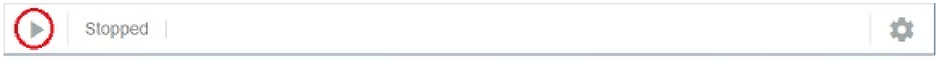
When the current project's API definition file includes `gateway: datapower-gateway`, the API is deployed locally on a IBM DataPower Gateway, which runs within a Docker container on your workstation. When you start the DataPower Gateway, your workstation must download the Docker image from Docker Hub and perform a build step. It can take some minutes to complete this step and another minute or two to start the gateway.

You can update the OpenAPI (Swagger 2.0) YAML file manually, or with the API Designer interface as shown in the instructions that follow.

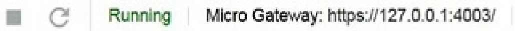
Important: From IBM API Connect Version 5.0.8.7, some commands are no longer supported by the API Connect toolkit. Such commands are indicated on this page by the V5.0.7 and earlier icon.

Procedure

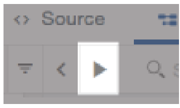
1. Start the API Designer.
   a. Enter **apic edit** to start the API Designer in a new browser window.
2. Select your test gateway.
   a. If not already open, use the Show / hide policy palette icon › to view the policy palette.
   b. If necessary, click the select provider icon to show the test gateways.
   c. Choose DataPower Gateway policies.
   When you select a gateway, the features that are available to it become visible in the policy palette.
   d. Click the save icon to save your preference in the YAML file.
3. Generate a gateway instance with just one of the methods that are described in this step. Allow several minutes for the gateway to come online, especially the first time it is run.
   a. V5.0.7 and earlier In the CLI, enter **apic start**.
   b. In the test console at the bottom of the screen, click the Start the servers icon:



A completed service start displays `Running` next to the gateway type, URL, and the port where it is available.



Your project configuration and other running processes can produce a different gateway type, URL, or port number than is displayed in the image.

4. Show the test tool pane in the API Designer interface, by clicking the Test icon that is highlighted in the image  .

   - You are ready to continue API testing on an instance of the IBM DataPower Gateway Docker container. For further instructions, see Testing an API with the API Designer test tool.
   - The new gateway entry can be viewed in the OpenAPI (Swagger 2.0) YAML file directly, or from within the Source tab in the API Designer interface.
   - You can treat your testing environment as a Catalog capable of property substitutions by configuring the special `apic-dev` Catalog. See the section that is entitled Configuring API definitions for container run times, at ../com.ibm.apic.install.doc/tapim_migrating_to_containers.html.

Related tasks

- Testing an API with the API Designer test tool

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Publishing APIs and applications

To publish APIs and applications by using the developer toolkit, 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. When you publish LoopBack® projects, you must publish both the APIs and the associated applications so the projects can be run.

See the following sections for more information:

- Setting configuration variables
- Logging in to cloud platforms
- Publishing APIs
- Publishing Node.js applications
- Publishing LoopBack APIs and applications

Setting configuration variables

The **apic config** command provides global and project-based configuration variables that specify the target Catalog and App 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 Using configuration variables.

Note: You can set a configuration variable locally (the default) to affect only the current LoopBack project, or globally (with command-line option **--global** or **-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.
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:

```
apic-catalog://management_server/orgs/org_name/catalogs/catalog_name
```

where *management_server* is the API Manager server, *org_name* is the organization name, and *catalog_name* is the Catalog name. You can override the values set by the Catalog URI configuration variable by using the **--catalog** command-line option.

Set the **app** configuration variable to the URI of an API Connect App to define the default App target for all commands managing applications. The **app** URI has the form:

```
apic-app://management_server/orgs/org_name/apps/app_name
```

Set the **org** configuration variable to the URI of an API Connect organization to define the default organization and server . The **org** URI has the form:

```
apic-org://management-server/orgs/org-name
```

where *management_server* is the API Manager server, and *org_name* is the organization name. The *management_server* portion sets the default value of the **--server** option and the *org-name* portion sets the default value of the **--organization** option.
V5.0.7 + You can append the port number to the server name if it is not the default value of 443.

The easiest way to determine the values for these configuration variables is to sign in to the API Manager of your provisioned API Connect service or your on-premises cloud, and click on the link icon of the Catalog or App to which you want to publish your API or application. The dialog that appears will provide you with the identifier of the Catalog or App along with the appropriate **config:set** command.

V5.0.5 + 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:

```
apic-space://management_server/orgs/org_name/catalogs/catalog_name/spaces/space_name
```

where *management_server* is the API Manager server, *org_name* is the organization name, *catalog_name* is the Catalog name, and *space_name* is the name of the Space. You can override the default values set by the Space URI configuration variable by using the **--server**, **--organization**, **--catalog**, and **--space** command-line options.
Although setting these configuration variables is not required, doing so simplifies commands that interact with API Connect clouds by providing default values for the following options:

- **--server**
- **--organization**
- **--catalog**
- **--app**
- V5.0.5 + **--space**

Here is an example of publishing with and without the **catalog** configuration variable set.

Without the configuration variable set:

```
apic publish climb-on.yaml --server mgmnthost.com --organization climbon --catalog sb
```

With the configuration variable set:

```
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb
apic config:set org=apic-org://mgmnthost.com/orgs/climbon
apic 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 publish** command to specify the **qa** Catalog:

```
 apic 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=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb
```

Logging in to cloud platforms

Use the `apic login` and `apic logout` commands to manage your authentication credentials for IBM® API Connect clouds. Unlike many systems, API Connect enables you to be logged in simultaneously to multiple clouds, so that you can easily publish and manage APIs and applications to disparate on-premises and IBM Cloud targets.

Log in supports both interactive and non-interactive modes. To log in interactively:

```
$ apic login
Enter your API Connect credentials
? Server: us.apiconnect.ibmcloud.com
? Username: username@example.com
? Password (typing will be hidden) ****************
Logged into us.apiconnect.ibmcloud.com successfully
```

You can perform scripted non-interactive log in by using the `--server`, `--username`, and `--password` options.

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 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 --type api --title Routes
apic create --type api --title Ascents
apic create --type product --title "Climb On" --apis "routes.yaml ascents.yaml"
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb
apic login --username some-user --password some-password --server mgmnthost.com
apic publish climb-on.yaml
```

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, retired, or archived. For example:

```
apic publish --stage climb-on.yaml
```

V5.0.5+ 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
```

V5.0.5+ If Spaces are enabled for a Catalog, Products can be staged and published only to a Space within that Catalog. To publish to a Space, the `--scope space` option must be included in the publish command, for example:

```
apic publish --scope space product.yaml --space space --catalog catalog --organization organization --
server server
```

where

- *product* is the name of the product that you want to publish.
- *space* is the name of the Space to publish to.
- *catalog* is name of the Catalog that contains the Space.
- *organization* is the name of the organization.
- *server* is the management server; *port*number is optional (default is 443).

If default configuration values have been set for the Space, Catalog, organization and management server, the following publish command could be used:

```
apic publish --scope space product.yaml
```

where *product* is the name of the product that you want to publish.
V5.0.5+ Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$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 V5.0.5+ Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.

Publishing Node.js applications

You can publish any Node.js application that has a properly-configured **package.json** file. Before publishing, you must install dependencies by running **npm install**. You must run the **apic apps:publish** command from the directory that contains **package.json**.

Publishing LoopBack APIs and applications

LoopBack projects contain both APIs and applications. Use the **apic publish** command described previously to publish the LoopBack APIs, and use the **apic apps:publish** command to publish the LoopBack application.

By default, a LoopBack project has default API and Product definitions in the ***project_dir*/definitions** directory. Publishing the API and Product artifacts is the same as any other set of API and Product artifacts except you can generate the artifacts directly from LoopBack models. For example:

```
apic loopback     # When prompted, enter "climbon" for project name and directory
cd climbon
apic create --type model           # Use as many times as required
apic loopback:property             # Use as many times as required
apic loopback:remote               # Use as many times as required
apic loopback:relation             # Use as many times as required
apic loopback:refresh              # (Re)generate the Product and API artifacts
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb
apic login --username some-user --password some-password --server mgmnthost.com
apic publish definitions/climbon-product.yaml
```

In addition to publishing the LoopBack APIs, you must also publish the associated LoopBack applications to an API Connect App that represents a Node.js run time. For example, add the following two commands to the above set to publish the LoopBack application:

```
apic config:set app=apic-app://mgmnthost.com/orgs/climbon/apps/sb-app
apic apps:publish
```

Note: You must run the **apps:publish** command from within the LoopBack project root directory.
If you publish the LoopBack project to IBM Cloud, the App can optionally be created in the organization as a side effect of **apps:publish**. In that case, the **app** configuration variable does not have to be set and the **--app** option on **apps:publish** is not required.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing API Products

Use the **apic products** and **apic apis** commands to manage Products and APIs that have been published to API Connect Catalogs.
V5.0.5 + Use the **--scope space** option to manage Products and APIs that have been published to Spaces within Catalogs.

Here's an example of using the **products** and **apis** commands through a full lifecycle:

Example command	Description
`apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb`	Set the default Catalog.
`apic login --username some-user --password some-password --server mgmnthost.com`	Login into the mgmnthost.com cloud.
`apic create --type api --title Routes --product "ClimbOn"`	Create the Product and API.
`apic publish --stage climbon.yaml`	Publish the Product to staged status.
`apic products`	List the Products in the Catalog.
`apic products:get climbon`	Display the Product's properties.
`apic apis`	List the APIs in the Catalog.
`apic apis:get routes`	Get the API's properties.
`apic products:set climbon --status published`	Publish the Product, making the API online.
`apic apis:set routes --status offline`	Take the API offline.
`apic apis:set routes --status online`	Bring the API online.
`apic products:set climbon --status deprecated`	Deprecate the Product.
`apic products:set climbon --status retired`	Retire the Product.
`apic products:set climbon --status archived`	Archive the Product.

Example command	Description
`apic products:delete climbon`	Delete the Product from the Catalog.

V5.0.5 + Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$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 V5.0.5 + Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.

Here's an example of a more complex lifecycle where a new version of a Product and API replaces the original version at runtime.

Set the default Catalog and login to the mgmnthost.com API Connect cloud:

```
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb
apic login --username some-user --password some-password --server mgmnthost.com
```

Create and publish an initial version:

```
apic create --type api --title Routes --version 1.0.0 --filename routes100.yaml
apic create --type product --title "Climb On" --version 1.0.0 --apis routes100.yaml --filename
climbon100.yaml
apic publish climbon100.yaml
```

Create a new version to fix a bug in the API, stage it to the Catalog:

```
apic create --type api --title Routes --version 1.0.1 --filename routes101.yaml
apic create --type product --title "Climb On" --version 1.0.1 --apis routes101.yaml --filename
climbon101.yaml
apic publish --stage climbon101.yaml
```

Inspect the Catalog:

```
apic products
apic products:get climbon:1.0.0
apic products:get climbon:1.0.1
```

"Hot-replace" version 1.0.0 with 1.0.1:

```
apic products:replace climbon:1.0.0 climbon:1.0.1 --plans default:default
```

In addition to the lifecycle management capabilities, you can download Products and APIs in Catalogs using the `pull` and `clone` sub-commands:

Command	Description
apic products:clone	Download all Products and their APIs from the Catalog.
apic products:pull climbon:1.0.0	Download the `climbon:1.0.0` Product and its APIs from the Catalog.
apic apis:clone	Download all APIs from the Catalog.
apic apis:pull routes:1.0.0	Download the `routes:1.0.0` API from the Catalog.

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.

V5.0.5 + 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
```

V5.0.5 + If Spaces are enabled for a Catalog, Products can be staged and published only to a Space within that Catalog. To publish to a Space, the `--scope space` option must be included in the publish command, for example:

```
apic publish --scope space product.yaml --space space --catalog catalog --organization organization --
server server
```

where

- *product* is the name of the product that you want to publish.
- *space* is the name of the Space to publish to.
- *catalog* is name of the Catalog that contains the Space.
- *organization* is the name of the organization.
- *server* is the management server; *port*number is optional (default is 443).

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 --organization climbonorg --server
mgmnthost.com
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

IBM 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.

Similar to the `products` and `apis` commands, you can use `drafts` to push, pull, and clone artifacts. For example:

```
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb    # set the default
Catalog
apic login --username some-user --password some-password --server mgmnthost.com  # login into the
mgmnthost.com cloud

apic create --type api --title routes --product ClimbOn
apic drafts:push climbon.yaml                             # push climbon:1.0.0 and routes:1.0.0 to
drafts
apic drafts                                               # list what is in drafts
apic drafts:get climbon                                   # get climbon:1.0.0
apic drafts:get routes                                    # get routes:1.0.0
apic drafts:pull climbon                                  # pull climbon:1.0.0 and routes:1.0.0 from
drafts
apic drafts:clone                                         # pull every Product/api from drafts
apic drafts:clear --confirm drafts                        # clear drafts collection
```

V5.0.5 + Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the draft API is created with the **apic drafts:push** command. For more information, see V5.0.5 + Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.
In addition to synchronizing data between local developer source code control systems and drafts, you can publish Products that are in drafts. For example:

```
apic config:set catalog=apic-catalog://mgmnthost.com/orgs/climbon/catalogs/sb    # set the default
Catalog
apic login --username some-user --password some-password --server mgmnthost.com  # login into the
mgmnthost.com cloud

apic create --type api --title routes --product ClimbOn
apic drafts:push climbon.yaml                             # push climbon:1.0.0 and routes:1.0.0 to
drafts
apic drafts:publish climbon
```

Although you can develop Products and APIs as drafts on your API Connect Management server by using the API Manager user interface, the preferred practice is to develop them locally. Therefore, if you already have draft Products and APIs in API Manager, either because you were previously a user of IBM API Management Version 4.0 or because you began by creating them there in IBM API Connect Version 5.0, use the `apic drafts:clone` command to create local versions of them, check them into your local source code control system, and continue developing them there. You can then publish to your Catalogs directly from your source code control system.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Toolkit command summary

A summary of the core commands in the IBM® API Connect developer toolkit.

Important: From IBM API Connect Version 5.0.8.7, some commands are no longer supported by the API Connect toolkit. Such commands are indicated on this page by the V5.0.7 and earlier icon.

- Viewing command line tool help
- Viewing version information
- Command stability
- Authenticating
- Using configuration variables
- Configuring the command-line tool to use TLS certificates
- Creating and managing local files
- Listing API Manager items
- Working with drafts
- Working with Catalogs and Spaces
- Command summary

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`** or **`apic -v`**. Display extended version information, including the versions of all the command-line tool plug-in modules, by entering the command: **`apic --ext-version`**.

Command stability

Some commands are identified as **`Stability: prototype`** in the command help text (and in the tables later in this topic). These commands are not production quality, but are provided for testing and customer feedback. The syntax and functionality of these commands will likely change before they are released as production quality.

Authenticating

Use the **`apic login`** command to authenticate to an API Manager endpoint, 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.

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.

The following table describes the configuration variables:

Table 1. Configuration variables

Variable name	Description	Use instead of (or override with) option...
V5.0.4+ accessibility-mode	Enable accessibility mode. To enable accessibility features, set to `enabled`. Accessibility features make the tools easier to use for those with limited eyesight.	N/A
app	Default app URI for all commands that manage aspects of an app. Form: `apic-app://mgmt-server/orgs/org-name/apps/app-name`, where *mgmt-server* is the management server, *org-name* is the organization name, and *app-name* is the Catalog name.	--app
catalog	Default Catalog URI for all commands that manage aspects of a Catalog. Form: `apic-catalog://mgmt-server/orgs/org-name/catalogs/catalog-name`, where *mgmt-server* is the management server, *org-name* is the organization name, and *catalog-name* is the Catalog name. Note: The Catalog name `apic-dev` is reserved for local testing.	--catalog
V5.0.1+ log-level	Level of logging detail for the local application and Micro Gateway. Must be one of: • `debug` - Verbose logging full debugging information. • `info` - Log messages of "info" level and more severe. • `warning` - Log only messages of "warning" level or more severe. • `error` - Log only messages of "error" level and more severe. • `fatal` - Log only messages of "fatal" (the most severe) level. Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.	N/A
org	Default org URI for all commands that manage organizations. Form: `apic-org://mgmt-server/orgs/org-name`, where *mgmt-server* is the management server, *org-name* is the organization name. The *mgmt-server* portion sets the default value of the `--server` option. V5.0.7+ You can append the port number to the server name if it is not the default value of 443.	--organization, --server
V5.0.5+ space	Default Space URI for all commands that manage aspects of a Space. Form: `apic-space://mgmt-server/orgs/org-name/catalogs/catalog-name/spaces/space-name`, where *mgmt-server* is the management server, *org-name* is the organization name, *catalog-name* is the Catalog name, and *space-name* is the Space name. V5.0.7+ You can append the port number to the server name if it is not the default value of 443.	--server, --organization, --catalog, --space
V5.0.2+ template-default-api	V5.0.2+ Default API template to use. Specify base file name of Handlebars (.hbs) file.	--template
V5.0.2+ template-default-product	V5.0.2+ Default Product template to use. Specify base file name of Handlebars (.hbs) file.	--template
V5.0.2+ template-path	V5.0.2+ Space-delimited list of absolute local directory paths containing Handlebar templates.	--template

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=apic-catalog://mgmtnhost.com/orgs/climbon/catalogs/sb
```

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:

1. Set the `trust-store` configuration variable to the name of the certificate file used in the server's trust store of the TLS profile by entering this command:

   ```
   apic config:set trust-store=<cert-file>
   ```

Where `<cert-file>` is the absolute file path to the TLS certificate file.

2. Set the value of the NODE_EXTRA_CA_CERTS environment variable to extend Node's built-in CA certificate store by entering this command:

   ```
   export NODE_EXTRA_CA_CERTS=<cert-file>
   ```

   Where `<cert-file>` is the absolute file path to the TLS certificate file.

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 --type api` command. To create a local Product definition file, use the `apic create --type 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 or Product in API Manager from a local API or Product definition file, use the `apic drafts:push` command.

To stage and publish a locally defined application, Product, and its referenced APIs to a Catalog in API Manager, use the `apic publish` and `apic apps:publish` commands.

V5.0.5 + If Spaces are enabled for a Catalog, Products can be staged and published only to a Space within that Catalog. To publish to a Space, the `--scope space` option must be included in the publish command, for example:

```
apic publish --scope space product.yaml --space space --catalog catalog --organization organization --
server server
```

where

- *product* is the name of the product that you want to publish.
- *space* is the name of the Space to publish to.
- *catalog* is name of the Catalog that contains the Space.
- *organization* is the name of the organization.
- *server* is the management server; *port*number is optional (default is 443).

V5.0.5 + Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$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 V5.0.5 + Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.
V5.0.6 +

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 --disable-analytics
```

Listing API Manager items

Use these commands to list items of the specified type:

```
apic apps
apic catalogs
apic drafts
apic orgs
```
V5.0.7 and earlier `apic props`
V5.0.7 and earlier `apic services`
V5.0.5 + `apic spaces`
V5.0.7 + `apic extensions`

Working with drafts

To work directly with draft APIs and Products, use the `apic drafts:action` command, where *action* is the action that you want to perform. For example, to publish a draft Product, and its referenced draft APIs, to a Catalog, use the `apic drafts:publish` command.

To create a local API or Product definition file from a draft API or Product, use the `apic drafts:pull` command.

To create local API or Product definition files from all the draft APIs and Products, use the `apic drafts:clone` command.

Note: You can publish a Product and its referenced APIs directly to a Catalog by using the `apic products:publish` command. You do not have to first create draft Products and APIs.

Working with Catalogs and Spaces

To create a Catalog, use the `apic catalogs:create` command. To view information on a Catalog, use the `apic catalogs:get` command; to list all Catalogs contained in organizations that the currently authenticated user is a member of, use the `apic catalogs` command. V5.0.8 + Beginning with Version 5.0.8.4, you can transfer the ownership of a Catalog to another user by using the `apic catalogs:transfer` command.

V5.0.5 + 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
```

Note: If Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog.
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.

To work directly with Products, apps, and APIs in a Catalog or Space, use the `apic products`, `apic apps`, and `apic apis` commands. For example, to update a Product, use the `apic products:set` command. If you need to specify the Space upon which to act, you must include the `--scope space` option in the command.

Command summary

The following tables summarize `apic` commands. The general command syntax is:

```
apic command:sub-command [argument] [options]
```

where *command* is the command, for example `config`, *sub-command* is the sub-command, where applicable (for example `get`), *argument* is the argument, where applicable (for example,`catalog`), and *options* is one or more options, where applicable (for example,`--local`). Some `apic` commands don't have sub-commands or arguments. For some commands, options are required. The full example command is:

```
apic config:get catalog --local
```

Note: The `create` command has a slightly different syntax:

```
apic create --type sub-command [options]
```

Some of the commands in the following tables are annotated with **Stability: prototype**, which indicates that IBM is in the process of collecting customer feedback on the commands and you should not use them in production scripts.

V5.0.2 + You can use a template to create an API or Product by running the following command:

```
apic create --type [api | product] --template template_filename --title new_title
```

where the *template_filename* is the name of the Handlebars template to use. The template must have an .hbs file name extension. Alternatively, when you create an API or Product interactively, you can specify a template. For more information, see Creating and using API and Product definitions templates.

Table 2. Summary of general-purpose apic commands

Command	Description	Sub-commands

Command	Description	Sub-commands
`apic config`	List and manage configuration variables. For more information, see Using configuration variables. With no sub-command, lists values of defined configuration variables.	• `get` - Get a configuration variable. • `list` - List configuration variables (default). • `set` - Set configuration variables. • `delete` - Delete a configuration variable. • `clear` - Delete all configuration variables
`apic create`	Create project artifacts.	• `--type api` - Create an API. • `--type api --wsdl` ***`filename`*** - Create a SOAP API 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. • `--type product` - Create a Product. • `--type model` - Create a LoopBack model. • `--type datasource` - Create a LoopBack data source. V5.0.2 + Note: You can create an API or Product from an OpenAPI (Swagger 2.0) template file by using the `--template` ***`template-name`*** option.
`apic edit`	Run the API Designer and open in default web browser.	None
V5.0.7 + `apic extensions`	Manage OpenAPI (Swagger 2.0) extensions in a catalog. With no sub-command, lists the extensions in the production catalog. **Stability: prototype**	• `clone` - Pull all extensions from a catalog. • `delete` - Delete an extension in a catalog. • `get` - Get information on an extension in a catalog. • `list` - List all extensions published to a catalog (default). • `publish` - Publish an extension to a catalog. • `pull` - Pull an extension from a catalog.
`apic loopback`	Create LoopBack project and project artifacts. With no sub-command, creates a new LoopBack project. All of these commands are **Stability: prototype**, except for `loopback:app`.	• `acl` - Add access control list specification. • `app` - Create a new LoopBack project (default). • `boot-script` - Add boot script. • `export-api-def` - Generate OpenAPI (Swagger 2.0) definitions from models. • `middleware` - Add middleware function. • `property` - Add property to existing model. • `refresh` - Update Product and API definition from a model. • `relation` - Add relation between models. • `remote-method` - Add remote method • `swagger` - Generate LoopBack project from an OpenAPI (Swagger 2.0) definition.

Command	Description	Sub-commands
V5.0.7 and earlier `apic microgateway`	Create Micro Gateway applications. Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.	None
`apic validate`	Validate API or Product definition YAML file.	None

Table 3. Summary of apic commands to manage and test applications locally

Command	Description	Sub-commands
`apic logs`	Display server logs continuously to console.	None
V5.0.7 and earlier `apic props`	List and manage service properties for a LoopBack application running locally. With no sub-command, lists values of defined service properties.	• **`get`** - Get a property value. • **`list`** - List all property values (default). • **`set`** - Set or update property values. • **`delete`** - Delete a service property. • **`clear`** - Delete all service properties.
V5.0.7 and earlier `apic services`	List and manage services. With no sub-command, lists all services that are currently executing.	• **`list`** - List all services currently executing (default) • **`start`** - Start a service. • **`stop`** - Stop a service. • **`logs`** - Display logs to console. • **`get`** - Get information about a service.
V5.0.7 and earlier `apic start`	If run in LoopBack project directory, start the LoopBack application; otherwise, start the Micro Gateway. Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.	None
V5.0.7 and earlier `apic stop`	Stop the specified service or all services.	None
V5.0.6 + `apic explore`	Opens the API Explore tool. Shows the operations, definitions, and documentation for all of the APIs that are contained in the project directory. To specify a particular API, include the name of the API in the command, for example: `apic explore` ***`apiname`***`.yaml`	Option: • **`-e, --external`** Opens the Explore tool on 0.0.0.0 instead of the default 127.0.0.1. This option binds the server to all IP addresses on the machine, and makes the tool accessible on the wider network.

Table 4. Summary of apic commands to manage APIs, apps, Products, and Catalogs, and publish them to the cloud

Command	Description	Sub-commands
`apic apps`	List, manage, or publish applications. Default sub-command is **`list`**.	• **`get`** - Get information about an application. • **`build`** - Build an application. • **`list`** - List provider apps contained in organizations of which the currently-authenticated user is a member (default sub-command). • **`publish`** - Publish an application to a provider app. • **`set`** - Update an application.

Command	Description	Sub-commands
`apic apis`	List and manage APIs in a Catalog or Space.**Stability: prototype** Default sub-command is `list`.	• `clone` - Pull all APIs from a Catalog or Space - Build an application. • `get` - Get information on an API in a Catalog or Space. V5.0.7 + The output separately lists the production and development endpoints at which an application can call the API. For more information, see Managing the application lifecycle. • `list` - List APIs in a Catalog or Space (default sub-command). • `pull` - Pull an API from a Catalog or Space. • `set` - Update an API in a Catalog or Space.
`apic catalogs`	List and manage Catalogs.	• `catalogs:create` Create a Catalog in an organization • `catalogs:delete` Delete a Catalog in an organization • `catalogs:get` Get information on a Catalog in an organization • `catalogs:list` - List Catalogs contained in organizations of which the currently-authenticated user is a member (default sub-command). • V5.0.8 + `catalogs:transfer` - Beginning with Version 5.0.8.4: Transfers the ownership of a Catalog to another user.
V5.0.4 + `apic devapps`	List and get information about consumer applications. Default sub-command is `list`.	• `get` - Get information about a consumer application. • `list` - List consumer apps in an organization and a Catalog.
`apic drafts`	List and manage APIs and Products in drafts. Default sub-command is `list`.	• `clear` - Delete all API and Product definitions in drafts. • `clone` - Pull all API and Product definitions from drafts. • `delete` - Delete an API or Product definition from drafts. • `list` - List APIs and Products in drafts (default sub-command). • `get` - Get information on an API or Product definition in drafts. • `publish` - Stage and publish a Product and its referenced APIs in drafts to a Catalog. • `pull` - Pull API or Product definitions from drafts. • `push` - Push local API or product definitions to drafts.
`apic login`	Log in to API Manager.	None. Specify server and credentials with the required flags: • `-u, --username` *user_name* • `-p, --password` *password* • `-s, --server` *mgmt_server*. V5.0.7 + You can append the port number to the server name if it is not the default value of 443.
`apic logout`	Log out from API Manager.	None. Specify server with the required flag: • `-s, --server` *mgmt_server*. V5.0.7 + You can append the port number to the server name if it is not the default value of 443.
V5.0.5 + `apic members`	List members in an organization. **Stability: prototype**	None

Command	Description	Sub-commands
V5.0.4 + `apic orgs:get`	Display information on a consumer or provider organization. Use `--type provider|consumer` to specify either provider or consumer organization.	- `get` - Display information on a consumer or provider organization.
`apic organizations` V5.0.4 + `apic orgs`	List and get information about organizations. Default sub-command is `list`. Note: The `organizations` command available in early releases is deprecated in favor of `orgs`.	- `get` - Get information on a provider organization. - `list` - List organizations of which the currently-authenticated user is a member. On IBM Cloud, lists only organizations where the API Management tile is provisioned and added.
`apic policies`	List and manage policies in a Catalog. Default sub-command is `list`.	- `clone` - Pull all policies from a Catalog. - `delete` - Delete a policy from a Catalog. - `get` - Get information on a policy in a Catalog. - `list` - List policies in a Catalog (default sub-command). - `publish` - Publish a policy to a Catalog. - `pull` - Pull a policy from a Catalog.
`apic publish`	Publish a Product and its referenced APIs to a Catalog. V5.0.5 + If Spaces are enabled for a Catalog, Products can be published only to a Space within that Catalog. To publish to a Space, the `--scope space` option must be included in the publish command.	None. This is an alias for `apic products:publish`.
`apic products`	List and manage Products in a Catalog. Default sub-command is `list`.	- `clear` - Clear all Products in a Catalog. **Stability: prototype** - `clone` - Pull all Product definitions from a Catalog. **Stability: prototype** - `delete` - Delete a Product in a Catalog. - `get` - Get information on a Product in a Catalog. - `list` - List Products in a Catalog (default sub-command). - `publish` - Stage and publish Product and referenced APIs to a Catalog. - `pull` - Pull a Product from a Catalog. - V5.0.4 + `replace` - Replace a Product in a Catalog with another Product . **Stability: prototype** - `set` - Update a Product in a Catalog. - V5.0.4 + `set-migration-target` - Set migration target of a Product in a Catalog with another Product. **Stability: prototype** - V5.0.4 + `supersede` - Supersede a Product in a Catalog with another Product. **Stability: prototype**
V5.0.5 + `apic spaces`	List and manage Spaces contained in a Catalog. Default sub-command is `list`.	- `spaces:create` Create a Space in a Catalog - `spaces:delete` Delete a Space in a Catalog. - `spaces:get` Get information on a Space in a Catalog. - `spaces:list` List Spaces contained in a Catalog (default sub-command). - `spaces:set` Set information on a Space in a Catalog.
V5.0.5 + `apic subscriptions`	List and manage subscriptions in a Product or a Catalog. Default sub-command is `list`.	- `subscriptions:get` - Get information on a subscription in an app. - `subscriptions:list` List Spaces contained in a Catalog (default sub-command). **Stability: prototype**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.3 +

Creating new projects with API Designer

You can create new LoopBack® and OpenAPI (Swagger 2.0) projects with the API Designer.

Procedure

To create a new project with API Designer, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the side bar, click the Projects Plus icon ⊕ .
3. Depending on the type of project you want to create, follow the steps in Creating a new LoopBack project or Creating a new OpenAPI project.

Creating a new LoopBack project

Before you begin

To complete this task, you must have internet access, because LoopBack installs dependencies from the public npm repository. Alternatively, you can configure a private npm repository.

About this task

To create a new LoopBack project with the API Designer, first, complete the steps in the preceding section, then follow these steps:

Procedure

1. Click Create LoopBack Project.
   You'll see the Add new LoopBack project dialog.
2. Select a project template: one of api-server, empty-server, hello-world, or notes.
3. For LoopBack Version, select either version 3.x (the current version) or version 2.x.
4. Enter values for the Display Name, Name, and Project Directory fields.
   Note: API Designer creates a new directory called Name to contain the new project within the specified Project Directory. You cannot choose the directory where you ran `apic edit` as the Project Directory; If you do, you will see the error message "Path already in use."
5. Click Add.

Results

The API Designer creates a new LoopBack project and makes the project available for editing.

Creating a new OpenAPI project

About this task

To create a new OpenAPI project with the API Designer, first, complete the steps in the preceding section, then follow these steps:

Procedure

1. Select Create Open API project.
   You'll see the Add new OpenAPI project dialog.
2. Enter values for Display Name, Name, and Project Directory.
   Note: API Designer creates a new directory called Name to contain the new project within the specified Project Directory. You cannot choose the directory where you ran `apic edit` as the Project Directory; If you do, you will see the error message "Path already in use."
3. Click Add.

Results

The API Designer creates a new OpenAPI project and makes the project available for editing.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Creating a database schema from models

Database schemas based on your models can be created and updated through API Designer, for data sources that support the models. This enables you to develop your models first, and create (and update) your database schema to match them. This process is sometimes referred to as "auto-migration" and is supported by data source connectors for MongoDB, MySQL, Oracle, PostgreSQL, and SQL Server.

Before you begin

Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also do the following:

1. Create a LoopBack® project. For more information, see Tutorial: Creating a LoopBack project from the command line.
2. Make the project root directory your working directory; for example:

   ```
   cd acme-bank
   ```

3. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
4. Create a data source of one of the supported types listed in the following section, with at least one model connected to the data source. For more information, see Tutorial: Creating a model and a data source in the API Designer.

About this task

API Designer can create and update a database schema based on existing models. Doing this will create (or modify) a table for each model, and a column in the table for each property in the model, if the data source is a relational database.

Note: This process creates models including Loopback built-in models, as well as any custom models you have created.
Subsequently, if your models change, you can recreate or update (synchronize) the database schemas accordingly if you need to adjust the database to match the models. When updating, this process will alter the database schema based on how the models have changed.

The following data sources support this feature:

- MongoDB. Since MongoDB is a "schema-less" database, the process will only create and update indexes. When updating a schema, if you change existing properties, then existing records in MongoDB will have the old property, but new records will have the new definition with the modified property.
- MySQL
- Oracle
- PostgreSQL
- SQL Server

For more information on how this feature works in LoopBack, see Creating a database schema from models.

Procedure

1. Click Data Sources .
2. Click the data source where you want to create a schema. The data source must use one of the supported connectors listed above.
3. Click Update Schema ⇄. The Update Database Schema dialog opens, listing all the models that use the current data source.
4. Click the models that you want to use to create or update the data source schema.
5. Note: If you update a schema with an existing table, and properties have been deleted from the corresponding model, the columns corresponding to the deleted properties may be removed from the table, and existing data may be destroyed, depending on the specific connector implementation and the underlying database permissions.
   Click Update Schema. If the existing tables already exist, this process will alter the tables based on the structure of the corresponding models.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Discovering models from relational databases

You can use API Designer to create models from existing database tables. This process is called *discovery* and is supported by data source connectors for: MySQL, Oracle, PostgreSQL, and SQL Server.

Before you begin

Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also do the following:

1. Create a LoopBack project. For more information, see Tutorial: Creating a LoopBack project from the command line or .Creating new projects with API Designer.
2. Add the project to API Designer. For more information, see Adding an existing project to API Designer.
3. Create a data source of one of the supported connectors listed in the following section. For more information, see Tutorial: Creating a model and a data source in the API Designer.

About this task

API Designer can create models based on an existing database schema. Doing this will create a model for each table, and a property in the model for each column in the table.

The following data source connectors support this feature:

- MySQL
- Oracle
- PostgreSQL
- SQL Server

For more information on how model discovery works programmatically and the LoopBack discovery API, see Discovering models from relational databases.

Procedure

1. Click Data Sources .
2. If the data source is one of the supported types listed above, the Discover Models option will be available. Click Discover Models. Alternatively, click the data source from which you want to discover models, and in the data source details page, click Discover Models 🔍.
3. The Discover Models page opens. It lists all the tables in the database to which the data source is connected, and the associated schema for each table.

4. Select the check box for each table for which you want to create models. When you select a table, the Select Properties dialog appears, showing the name and type of each property to be created, and whether the property is required and an ID property. By default, each model will be created with a property for each column in the table, so all the properties are selected.
5. In the Select Properties dialog, deselect any properties you don't want to create, and click Select.
6. Follow step 5 for each model you want to create.
7. Click Generate.

Results

API Designer creates a model for each of the selected tables. Each model has a property for each selected column in the table.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 + Adding an existing project to API Designer

You can add existing LoopBack and OpenAPI (Swagger 2.0) projects from the local file system and then edit them withAPI Designer.

About this task

You can add existing LoopBack and OpenAPI (Swagger 2.0) projects from the local file system and then edit them with API Designer.
Note: If the LoopBack project does not include OpenAPI definition files (in the project's definitions directory), you must run the `apic loopback:refresh` command to create them before adding the project.

Procedure

To add a LoopBack project into API Designer, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click ⊕ to the right of Projects.
3. Select Add an existing project.
   You'll see the Add existing project dialog.
4. Browse to the desired project directory.
5. Click OK.
6. Enter Display Name and Name for the project.
7. Click Add existing project.

Results

The project is now available for editing in API Designer and is selected as the current project to edit.
Note: If the LoopBack project does not have OpenAPI definition files (in a definitions directory), you must run the `apic loopback:refresh` command to create them.

Adding an existing OpenAPI project

About this task

You can add an existing OpenAPI (Swagger 2.0) project from the local file system to the API Designer.
Note: Currently adding an OpenAPI project simply creates a new empty directory for the project. However, you can create API and product definitions for the project in the API Designer.

Procedure

To add an OpenAPI project to API Designer, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.

2. Click ⊕ to the right of Projects.
3. Select OpenAPI project.
   You'll see the Add existing project dialog.
4. Browse to the desired project directory.
5. Click Next.
6. Enter Display Name and Name for the project.
7. Click Add existing project.

Results

The project is now available for editing in API Designer and is selected as the current project to edit.

Selecting a project to edit

About this task

Once you have created a new project or added a project as described in Adding an existing project to API Designer and Adding an existing OpenAPI project, you can switch between different projects in API Designer.
Note: When you start API Designer in a project directory, that project is the initial current project.

Procedure

To select a project to edit, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the UI navigation pane, Projects lists all the projects you have created or added. Click the name of the project you want to edit.

Results

The project is now selected as the current project in API Designer and the project name is displayed in the main navigation banner.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Publishing a LoopBack application through the API Designer

Publish a LoopBack® application to an API Connect collective to make the application available for use by your API definitions. You can also publish a Product to a Catalog at the same time. V5.0.5 + The syndication feature in IBM® API Connect means that you can also publish a Product to a Space in a Catalog.

Before you begin

To complete this task, you must have:

- Created a LoopBack application.
- Set up an API Connect collective and added it in the Cloud Manager. For more information, see Installing API Connect Collective.
- Created an App in API Manager. For more information, see Creating an App.
- Permission to publish applications to your API Connect collective.

Important:

- IBM API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

Note:

- LoopBack projects contain both APIs and applications. To enable LoopBack projects to run, you must publish both the APIs (by publishing the Products that contain the APIs) and the associated applications.
- V5.0.5 + 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 IBM API Connect®.

Procedure

1. If you want to add or edit a target, complete the following steps:
   a. In the API Designer, click Publish.
   b. Click Add and Manage Targets.
   c. Click Add a different target.
   d. In the API Connect host address field, enter the host address of your API Manager instance. For example, use `hostname.com` when you would use `https://hostname.com/apim` to access API Manager. Do not use an IP address.
   e. In the Username and Password fields, enter your API Manager user name and password.
   f. Click Sign in.
   g. Click Next.
   h. In the Organization field, select the provider organization to which you want to stage or publish any Products and then click the Catalog to which you want to stage or publish any Products. If you do not want to stage or publish any Products to this target, click None.
      If you have many Catalogs to choose from, you can filter them by using the Search field.
   i. Click Next.
   j. Click the App to which you want to publish your application.
   k. Click Save to add the publishing target.
2. In the API Designer, click Publish.
3. Click the target to which you want to publish your application.
4. Select Publish application, and if you want to stage or publish a Product at the same time, select Stage or Publish products.
5. Optional: If you are staging a Product, select Stage only.
6. Optional: If you do not want to stage or publish all Products in your current project, select Select specific products and then select the Products that you do want to stage or publish.
7. Click Publish.

Results

Your application is published to your API Connect collective. If you staged or published a Product, it is also staged or published.

Related concepts

- Publishing APIs and applications
- Working with Products in the API Designer

Related tasks

- Staging a Product

Related information

- Tutorials for working with LoopBack projects

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 and earlier

Viewing application performance metrics

IBM® API Connect provides two ways to view application performance metrics for Node.js applications running locally: V5.0.7 + the built-in metrics dashboard and sending the data to third-party consoles or log files.

Important: From IBM API Connect Version 5.0.8.7, the `apic start` command referred to on this page is no longer supported by the API Connect toolkit.
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.

- V5.0.7 and earlier **Viewing the application metrics dashboard**
- V5.0.7 and earlier **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 and earlier

Viewing the application metrics dashboard

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.

Important: From IBM API Connect Version 5.0.8.7, the `apic start` command referred to on this page is no longer supported by the API Connect toolkit.
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: Application metrics dashboard 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 a 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 and earlier

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.

Important: From IBM API Connect Version 5.0.8.7, the **apic props** command referred to on this page is no longer supported by the API Connect toolkit.
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.
  For details on how to create and manage Apps, see Working with Apps.
- *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.

For details on how to publish an application, see Publishing APIs and applications and Publishing a LoopBack application through the API Designer.
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 send 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating API definitions by using the API Designer

The API Designer is a graphical user interface within the developer toolkit and provides functions for the creation and configuration of API definitions, running offline.

Running the API Designer

API Designer is run by using the edit command from the command line interface. When the edit command is used, the API Designer opens in your default browser or can be accessed through the local host port displayed when you run the command.

The complete command is `apic edit`.

Permissions and the API Designer

Anybody using the API Designer can perform all actions within it. However, when staging or publishing a Product, permissions will be enforced by API Connect. For more information about user roles and permissions, see Administering user access.

The following topics provide information on using the API Designer to create and configure APIs and Products:

- **Creating API definitions**
  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. In the context of API Manager, applications are typically mobile or web applications, and they use the HTTP protocol.
- **API policies and logic constructs**
  Policies and logic constructs are a pieces of configuration that control a specific aspect of processing in the Gateway server during the handling of an API invocation at run time.
- V5.0.7+ **Extensions commands**
  Use extensions commands to view and manage extensions.
- **Tags in API Connect**
  There are three forms of tagging in API Connect.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating API definitions

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. In the context of API Manager, applications are typically mobile or web applications, and they use the HTTP protocol.

About this task

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 an Invoke (invoke) or Proxy (proxy) policy.

In either case, you can configure your API definition either by using the API Designer or by writing an OpenAPI (Swagger 2.0) definition and publishing it using either API Designer or the command line interface.

Additionally, with IBM® Integration Bus version 10.0.0 or later, you can push a REST API to IBM API Connect. For more information, see Pushing a REST API to an IBM API Management server in the IBM Integration Bus online documentation.

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, through API Manager or through API Designer. For more information on creating SOAP API definitions through API Designer or API Manager, see Adding a SOAP API definition.

OAuth provider endpoint

You create an OAuth provider as the first stage of implementing OAuth 2.0 authentication. A provider endpoint is used to send and receive the tokens and requests that form the OAuth process. You can create an OAuth provider endpoint through the API Designer or from a template through the command line interface of the developer toolkit. For more information about creating a provider endpoint through the API Designer, see Protecting an API with OAuth security definition.

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.
Information about creating and using APIs in the API Designer can be found in the following topics:

- **Adding a REST API definition**
  In the API Designer, you can add a REST API definition either by composing the API definition, and its operations, from scratch, or by importing an OpenAPI (Swagger 2.0) definition. In API Manager you can also discover a REST API definition in a registry.
- **Adding a SOAP API definition**
  If you have an existing SOAP service that you want to expose more widely, you can add a SOAP API to 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.
- **Creating Paths**
  You create Paths, which are the route through which users access REST APIs. Paths consist of one or more HTTP operations such as GET or POST.
- V5.0.6 + **Organizing your APIs and Products into categories**
  You can organize your APIs and Products into categories. The APIs and Products that you categorize in the API Designer or API Manager UI are displayed within the Developer Portal, in their defined categories.
- **The assemble view**
  The API Designer 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.
- **The source view**
  The API Designer features a source view where you can review or edit the OpenAPI (Swagger 2.0) definitions of your APIs.
- **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.
- **Adding an API definition to a new Product**
  If you created an API definition without adding it a Product, and you later want to add the API to a new Product, you can create the Product and add the API to it in a single operation.
- V5.0.4 + **Adding an API definition to existing Products**
  If you created an API definition without adding it to a Product, or if you want to add an existing API definition to additional Products, you can do so while viewing the API details.
- **API properties**
  Your APIs are configured by using properties.
- **Testing an API with the API Designer test tool**
  Even while you are working offline, you can test the API to ensure that is defined and implemented correctly.
- **Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Designer UI)**
  When you configure a REST or SOAP API definition, you can use an OpenAPI (Swagger 2.0) extension to go beyond the standard

OpenAPI (Swagger 2.0) schema. You can also replace an extension with an updated version.

- **Referring to an extension in an API definition**
  When using the command-line tool you must manually add a reference to the extension to your API definition YAML (or JSON) file in the extensions key under x-ibm-configuration.
- **Variable references in API Connect**
  In API Connect you can reference different variables in your API definition.
- **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a REST API definition

In the API Designer, you can add a REST API definition either by composing the API definition, and its operations, from scratch, or by importing an OpenAPI (Swagger 2.0) definition. In API Manager you can also discover a REST API definition in a registry.

Before you begin

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.

About this task

For more information about the ways in which you can add a REST API definition, see the following subtopic:

- **Composing a REST API definition**
  You can create and edit draft REST API definitions by using the API Designer or API Manager user interfaces.
- **Adding a REST API by using an OpenAPI (Swagger 2.0) file**

Related tasks

- Composing a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Composing a REST API definition

You can create and edit draft REST API definitions by using the API Designer or API Manager user interfaces.

Procedure

To compose a REST API definition, complete the following steps:

Note: The API Manager and API Designer user interfaces both include the ability to create and edit APIs. However, the preferred method for these tasks is by using the API Designer user interface, as described in the following steps. Any tasks that are specific to a particular user interface are marked with an icon.

1. Click APIs.

The APIs tab opens.

2. If you have not previously pinned the UI navigation pane then click the Navigate to icon .

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon .
3. Click Drafts in the UI navigation pane, and then click APIs.

   The APIs tab opens.
4. V5.0.6 + Click Add > New API.

   V5.0.4 + Click Add and then click New OpenAPI from scratch.

   V5.0.3 and earlier Click Add and then click API from the Compose section.
5. Specify basic information about the API.
   - The title can include special characters but should be kept short so that it can be easily displayed in the user interface.
   - The name should be kept short and can contain only lowercase alphanumeric characters (a-z and 0-9), underscore characters (_), or hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
   - V5.0.4 + 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 version corresponds to the value of the `info.version` property of the API's OpenAPI (Swagger 2.0) definition. The `version.release.modification` version numbering scheme is recommended; for example `1.0.0`.
6. V5.0.4 + Expand Additional properties to specify additional properties for the API.
   a. From the API template field, select Default if you want to use the template defined as the default, to create the API definition. This can either be the built-in default .hbs template file, or another template file that you have configured as the default by using configuration variables. You can also select OAuth2 Provider to use the template for an OAuth 2.0 API definition, or select a custom template that you created.

      Note:
      - API Designer only Any templates that you have configured for use in the developer toolkit command line interface, including any configured default, are available for selection from this field.
      - API Manager only Only the built-in templates, which are provided with API Manager, are available for selection. Custom templates are not displayed.

      For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.

   b. Specify a target endpoint if known.

   c. Define the security scheme that an application must use to identify itself when calling the API operations. Select None, Client ID, or Client ID and secret. (Securing an API with a client ID and client secret is similar to requiring a user ID and password.)

   d. Select the Enable CORS check box to enable cross-origin resource sharing (CORS) support for your API. If you clear this check box, you can enable CORS support later, as described in Enabling CORS support for an API.

   e. Select Micro and DataPower Gateways, Micro Gateway, or DataPower Gateway as the Gateway server.
7. V5.0.3 and earlier Specify whether your API should be included in a Product, and then create the API definition.

   a. Select one of the following options:
      - To create your API without adding it to a Product, ensure that Don't add to a product is selected.
      - To create a new Product and include your API in that Product, select Add to a new product and provide a name for your Product.
      - To add your API to an existing Product, select Add to an existing product and then select the Product to which you want to add your API.

   b. Click Add.

      The Design tab for the draft of your API definition opens. You can skip to different sections of your API definition by using the page navigation in the side bar. You can view the OpenAPI (Swagger 2.0) definition of your API in the Source tab and, after you have created an assembly, view your policy assembly in the Assemble tab.
8. V5.0.4 + Specify whether your API should be included in a Product, and then create the API definition.
   - To create a new Product and include your API in that Product, complete the following steps:
     - Click Add a product.
     - API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
     - Accept the default values for the Product title, name, and version, or change them as required.
     - To publish the Product to a target Catalog, ensure that the Publish this product to a catalog check box is selected. You can clear this check box and stage or publish the Product later by using the API Designer UI and API Manager UI, as described in Staging a Product and Publishing a Product.
     - V5.0.4 and earlier Select the Catalog that you want to use.
     - V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.Your Product is staged to the Space that you selected.
     - Click Create API.
   - To create your API without adding it to a Product, click Create API.

Tip: You can add the API to one or more Products later, as described in Adding an API definition to existing Products.

The Design tab for the draft of your API definition opens. You can skip to different sections of your API definition by using the page navigation in the side bar. You can view the OpenAPI (Swagger 2.0) definition of your API in the Source tab and, after you have created an assembly, view your policy assembly in the Assemble tab.

9. In the Design tab, edit the Info section.
   a. Optional: Edit any or all of Title, Name, Version, and Description.
   b. Optional: In the Contact section, provide details for any or all of Name, Email, and URL.
   c. In the Terms and License section, provide details for any or all of Terms of Service, License Name, and License URL.
   d. In the External Documentation section, provide a Description for any external documentation you want to refer users to and a URL for where the documentation can be accessed.
10. In the Schemes section, select which transfer protocols you want your API to use.
    Note:
    - If your API is enforced by an IBM API Connect gateway, only the HTTPS protocol is supported. See Step 15.c for instructions of how to enable enforcement.
    - Micro Gateway only If you are using a Micro Gateway, you cannot have both an API that uses HTTP and an API that uses HTTPS. Doing so returns the following error when starting the Micro Gateway:

      ```
      test-gw did not return a port in timeout.
      Error: Service test-gw started but did not initialize within the timeout period. Dumping log
      buffer.
      undefined
      ```

11. Optional: If your API is to be reached by using a host name that is not your gateway cluster, use the Host field in the Host section to define the host name that is to be used. This does not affect the API's implementation, but will affect test tools and the OpenAPI (Swagger 2.0) definition that is made available to developers.
12. In the Base Path section, you can change the path segment that is shared by all operations in your API.
13. 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 Creating Paths.
14. 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 Creating Paths.
15. Configure the Lifecycle section.
    a. Optional: For Phase, use the drop-down menu to change the phase of the life-cycle that your API is in.
       The options are as follows:

       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.

       Realized
          The API is in the implementation phase.

    b. Optional: Set the Testable toggle to the On position to allow the APIs 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 development Catalog.
    c. Set the Enforced toggle to the On position to enforce the API by using the IBM API Connect Gateway.
    d. Set the CORS toggle to the On position to enable CORS access control.
    e. V5.0.8 + To protect your API with a certificate, by using TLS mutual authentication for example, select Authenticate application.
       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 or Adding more Gateway services.

       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.

       Important: The capability to protect an individual API with TLS mutual authentication was introduced in the IBM API Connect V5.0.8.1 release. If you enable the Authenticate application option for an API, the API will fail validation if imported into in a previous release. If you want to re-use the API in a previous release, remove the following lines from the source YAML for the API:

```
application-authentication:
  certificate: true
```

16. In the Policy Assembly section, click Create assembly to create a policy assembly for your API.
    The assemble view opens and displays a blank assembly on the canvas. For more information about the assemble view, see The assemble view.
17. In the Security Definitions section of the design view, manage any security definitions that might be used by the API or its operations. For more information, see Configuring API security
18. In the Security section, select any security definitions that you want to apply to your API. To be available in the Security section, definitions must have been defined in the Security Definitions section.
19. API Manager only V5.0.2 + In the Extensions section, add any vendor extensions you want to use with your API. For more information see, Adding an extension schema to a REST API definition.
20. In the Properties section, define any API properties that you want to use. For more information, see API properties.
21. V5.0.8 + In the Analytics section, add fields to specify your datatypes.
    For convenience, the header and body types of parameter entries that are listed in the Parameters section are available as items in the drop-down list. You can also add fields to the table that are not already specified in the Parameters section.
    a. Click Add Field + to create a field.
    b. Optional: If you are selecting an existing header or body parameter, select one of the listed Parameters.
    c. Optional: If you are creating a field from the beginning, select Add new field.
    d. Verify or modify the following information in the table for the new field:

       Name
          Names the field that you refer to in your visualization.
       Located in
          Identifies where the field is located in the call. The supported options are payload and header.
          Note: If you specify a payload type, the data must be in a JSON Object. Data that is in a JSON Array or other datatype is not recognized. The field can be nested, but cannot be nested in an array. For example, to map the value of *field_3*, you can enter *field_3* as shown in the following example:

          ```
          {"field_1":{"field_2":{"field_3": 123}}}
          ```

       ES type
          Specifies the type of search method that can be used for Elasticsearch. The options are keyword, text, long, integer, short, double, float, and geo-point.
       From
          Identifies what type of call the field is found in. Options are request and response.

22. In the Parameters section, add parameters that are shared by all Paths and operations in the API.
    a. Click the Add Parameter icon ⊕.
    b. In the Name field, provide a name for your parameter.
    c. In the Located In field, select where the parameter is found in the call of your operation.
    d. Optional: In the Description field, provide a description of your parameter.
    e. Use the Required check box to specify whether the parameter is required for a call to be valid.
    f. Optional: From the drop-down list for 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. For more information about creating JSON schemas, see step 24.
23. Add paths to your API. For more information, see Creating Paths.
24. In the Definitions section, create JSON schema definitions.
    You reference these definitions in an operation to provide developers with information about the JSON request they should make or the JSON 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 (validate) policy is used.
    a. Click the Add Definition icon ⊕.
       A new definition is created.
    b. Click the newly created definition to expand its details.
    c. Complete the Name field for your definition.
    d. From the Type drop-down list, select the type of your definition.
    e. Optional: In the Description field for your definition, provide a description of what is defined by the definition.
    f. Complete the details of your definition's properties. Each property requires a name and type and can also have a description.
    g. Optional: To specify that a property is required when the operation is called, select it using the check box in the Required The Required icon column.
    h. Optional: You can add additional properties by clicking Add Property.
    i. Optional: You can delete properties or definitions by clicking the Delete icon 🗑 next to the property or definition.
    j. 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, set Allow additional properties to the On position.

    Note:

- You can include more complex schema definitions in your API by using the Source tab and editing your API's OpenAPI (Swagger 2.0) definition directly. For more information, see the OpenAPI (Swagger 2.0) specification.
- Due to a change in the OpenAPI (Swagger 2.0) specification, references to object definitions created before the application of any fixpacks to IBM API Connect Version 5.0 fail validation upon staging. To rectify this, use the Code view to edit the OpenAPI (Swagger 2.0) definition. Where the object definition is referenced, delete the line `type: Definition_Name` where *Definition_Name* is the name of the referenced object definition.
- If, for a schema definition property of type `long`, you specify a `minimum` or `maximum` parameter in the OpenAPI (Swagger 2.0) source, the highest value you can set is `9007199254740992`; if you exceed this value then, due to JavaScript rounding errors, an incorrect value might be saved.

V5.0.6 + Tip: You can click the Edit inline schema icon ✎ next to a definition to display the Provide a schema window. In this window, you can create a schema definition in YAML format or in JSON format by editing the source in the Schema as YAML or Schema as JSON tabs. You can also generate a schema definition from a sample JSON or XML response, by using the Generate from sample JSON or Generate from sample XML tabs. When you want to generate a schema definition from a sample, enter the sample response into the appropriate generate tab, and click Generate to create the schema definition.

25. API Manager only V5.0.1 + In the Services section, add any web services that you want to use in your API definition. For more information, see Adding an existing web service to your API definition.
26. To add any tags, in the Tags section click the Add Tag icon ⊕. Tags added in this way appear in the OpenAPI (Swagger 2.0) definition of the API but are not used by API Connect for any indexing.
27. Click the Save icon 💾 to save your changes.

Related concepts

- Working with Products in the API Designer

Related tasks

- Creating Paths

Related reference

- API and Product definition template examples

Related information

- Configuring API security

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a REST API by using an OpenAPI (Swagger 2.0) file

You can use an OpenAPI (Swagger 2.0) definition file to add a REST API.

Before you begin

Your file must conform to version 2.0 of the OpenAPI (Swagger 2.0) specification. The format of the file can be JSON or YAML.

Procedure

To add a REST API by loading an OpenAPI (Swagger 2.0) file, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon .

3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. Click Add and then select V5.0.0 ONLY Swagger 2.0, V5.0.1 + OpenAPI (Swagger 2.0), or V5.0.4 + Import an existing OpenAPI from the Import section.
   The Import OpenAPI (Swagger) window opens.
5. Optional: To upload a file from your local file system, click Select file and, in your file system, select the file that you want to use.
   The following file types are supported if they contain a valid OpenAPI (Swagger 2.0) definition: .json, .yml, and .yaml.
6. Optional: To upload a file from a URL, click Or import from URL and then provide the correct URL in the URL field that is presented. If authentication is required to access the URL, provide a user name and password.
   The following file types are supported if they contain a valid OpenAPI (Swagger 2.0) definition: .json, .yml, and .yaml.
7. V5.0.4 + To create a new Product and include your API in that Product, complete the following steps. (If you want to create your API without adding it to a Product, proceed to step 8.)
   a. Select Add a product.
   b. API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
   c. Specify values for the Product title, name, and version.
   d. To publish the Product to a target Catalog, ensure that the Publish this product to a catalog check box is selected. You can clear this check box and stage or publish the Product later by using the API Designer UI and API Manager UI, as described in Staging a Product and Publishing a Product.
   e. V5.0.4 and earlier Select the Catalog that you want to use.
   f. V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.Your Product is staged to the Space that you selected.
8. Click Import.
   A new REST API definition is created, including Paths and HTTP operations.

Results

When the API definition has been imported, it is shown in the list of API definitions in the APIs tab of the Drafts page.

What to do next

You can edit your API definition as you would any other REST API definition. For more information, see Composing a REST API definition.

To finish the creation of your API definition, complete the following tasks.

- Configure security for your API. For more information, see Configuring API security
- Enable other users to add and define API definitions. For more information, see Administering user access.

Related information

- IBM API Connect overview
- Managing your V5 APIs
- OpenAPI (Swagger 2.0) Specification
- What is OpenAPI (Swagger 2.0)?
- API and Product definition template examples

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a SOAP API definition

If you have an existing SOAP service that you want to expose more widely, you can add a SOAP API to 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.

About this task

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:

- You can provide the WSDL as a file or URL.
- You can provide one or more WSDL files in a single .zip file that contains the WSDL files and any necessary schemas.
- API Manager only If the service owner registered that service in a supported service registry, you can search for the SOAP service in a service registry. For more information, see Adding a SOAP API definition by discovering a service from a registry.

The service must support Web Services Basic Profile Version 1.1 - Second Edition.

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.
Use the following tasks to add and configure a SOAP API.
Note: These tasks can be completed only by users who have the ability to edit draft APIs. Users with the ability to view draft APIs cannot create, edit or delete APIs.

- **Adding a SOAP API definition by using a WSDL file**
  If a SOAP service owner provides you with the details of the service in a WSDL file, you can use the WSDL file to add a SOAP API definition. You can also provide a URL link to a location from which you can obtain the WSDL definition.
- **Configuring a SOAP API definition**
  You can configure the details for a SOAP API definition. For example, you can select the service endpoint to which requests are sent.

Related concepts

- Developing your V5 APIs and applications

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a SOAP API definition by using a WSDL file

If a SOAP service owner provides you with the details of the service in a WSDL file, you can use the WSDL file to add a SOAP API definition. You can also provide a URL link to a location from which you can obtain the WSDL definition.

About this task

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.
Note:

- If the WSDL file is a stand-alone file with no external dependencies, you can load the .wsdl file either from a directory or a URL to create the SOAP 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 SOAP API definition.
- If you want to obtain the WSDL definition from a URL, the target file must be a .wsdl file that is accessible from the URL. A .zip file, which contains a WSDL file and its dependent documents, cannot be loaded from a URL.

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.

Procedure

To add a SOAP API definition by loading a WSDL file, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. V5.0.4 + Click Add > New OpenAPI from SOAP service.
   V5.0.3 and earlier Click Add > API from WSDL.
   The "New API from WSDL" window opens.

5. To upload the service information from a stand-alone .wsdl file, a .zip file that contains a WSDL file and its dependent documents, or a registry, use one of the following options:
   - Click Upload file and browse for the .wsdl file or .zip file.
   - (.wsdl file only) Click Load from URL and then complete the WSDL URL field. If the URL is secured, you must provide the user name and password for accessing the URL. Click Next.
   - API Manager only Click Find in registry to find the details for an existing SOAP service in a registry. For more information about using this option, see Adding a SOAP API definition by discovering a service from a registry.

   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.

   All of the services that are contained in the WSDL file are displayed. The WSDL file might contain multiple service definitions. You can view all of those services and the operations for each service.

6. V5.0.3 and earlier Specify the services to include, and then create the API definition.
   a. Select one or more of the services that you want to add as SOAP API definitions and then click Next.
   b. Optional: If you want to create a Product and include your SOAP API in the Product, select Create product and complete the Product name field.
   c. Click Done.

   Your SOAP API definition is created and displayed in the list of API definitions in the APIs tab of the Drafts page. You can edit the SOAP API definition by clicking its name in the list.
7. V5.0.4 + Specify the services to include, and then create the API definition.
   a. Select one or more of the services that you want to add as SOAP API definitions.
      You can then either choose to create a new Product and include your API in that Product, or create your API without adding it to a Product.

   b. If you want to create a Product and include your SOAP API in that Product, complete the following steps:
      i. Click Add a product.
      ii. API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
      iii. Accept the default values for the Product title, name, and version, or change them as required.
      iv. To publish the Product to a target Catalog, ensure that the Publish this product to a catalog check box is selected. You can clear this check box and stage or publish the Product later by using the API Designer UI and API Manager UI, as described in Staging a Product and Publishing a Product.
      v. V5.0.4 and earlier Select the Catalog that you want to use.
      vi. V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.Your Product is staged to the Space that you selected.
      vii. Click Done.
   c. If you want to create your API without adding it to a Product, click Done.
      Tip: You can add the API to one or more Products later, as described in Adding an API definition to existing Products.

   The Design tab for the draft of your API definition opens. You can skip to different sections of your API definition by using the page navigation in the side bar. You can view the OpenAPI (Swagger 2.0) definition of your API in the Source tab and, after you have created an assembly, view your policy assembly in the Assemble tab.

What to do next

- Configure the details of the SOAP API definition, see Configuring a SOAP API definition.
- Configure security for your API definition. For more information, see Configuring API security

Related concepts

- Developing your V5 APIs and applications

Related reference

- API and Product definition template examples

Related information

- IBM API Connect overview
- Adding a SOAP API definition by discovering a service from a registry

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring a SOAP API definition

You can configure the details for a SOAP API definition. For example, you can select the service endpoint to which requests are sent.

Before you begin

You must complete one of the following tasks:

- Adding a SOAP API definition by using a WSDL file
- API Manager only Adding a SOAP API by discovering a service from a registry

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM® Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.

About this task

After you create a SOAP API definition, you can specify API details to ensure the API starts and runs properly.

Procedure

To configure a SOAP API definition, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ≡.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the name of the SOAP API definition that you want to configure.
4. Configure the SOAP API definition as you would a REST API definition.
   Paths, operations, and definitions have already been created. In the assembly view, there is a proxy policy that invokes the service's URL. In the source view, additional configuration for your SOAP API can be found that is not displayed in the design view. For more information about configuring an API definition, see Composing a REST API definition.

Results

The endpoint is associated with the SOAP API.

What to do next

You can now test the invocation of the API by using the embedded test tool in the Assemble tab, see Testing an API with API Manager test tool or Testing an API with the API Designer test tool.

Related concepts

- Developing your V5 APIs and applications

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating Paths

You create Paths, which are the route through which users access REST APIs. Paths consist of one or more HTTP operations such as GET or POST.

About this task

For ways in which you can configure Paths and add them to APIs, see the following topic:

- **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Procedure

To define a Path, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Designer UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the REST API definition that you want to work with.
5. In the Paths section, click the Add Path icon ⊕.

A path is added.

6. In the Path field, add a path segment.
   Note: The full API, which is formed from the base path of the containing API followed by the path segment you define here, 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.

7. By default, the GET HTTP operation is used. To remove an operation, click the Delete Operation icon 🗑 to its right.
8. To add another operation type, click Add Operation. Depending on what function you want the operation to provide, select one of the following HTTP 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 IBM 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..
9. Optional: Add any parameters that you want to include for the Path and all of its operations.
   a. Click Add Parameter in the Path section.
   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 found in the call of your operation.
   d. Optional: If you selected Path in the previous step, you must define the location of the parameter in the Path field in the following form:

      ```
      /Path_Segment_1/{Parameter}/Path_Segment_2
      ```

      where:
      - the *Path_Segment* variables are the names of your path segments.
      - *Parameter* is the name of your parameter. It must match the name used to define it in step 9.b
   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.
   g. Optional: From the drop-down list for 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. For more information about creating JSON schemas, see Step 24 of Composing a REST API.
10. 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.
11. Click the Save icon 💾 to save your changes.

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 in the API Designer. For more information about testing, see Testing an API with the API Designer test tool.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

An operation must have at least one successful response defined.

Procedure

To configure an operation, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Designer UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the REST API definition that you want to work with.
5. In the Paths section, under the Path to which your operation belongs, click the operation that you want to work with.
   The operation's details expand.
6. Optional: In the Add Tag field, add any tags that you want to associate your operation with.
7. Optional: In the Summary field, provide a summary of your operation.
8. Optional: In the Operation ID field, provide an identifier for your operation. The operation ID must not be shared by any other operations.
9. Optional: In the Description field, provide a description of your operation.
10. Optional: Add any parameters that you want to include for only the operation you are configuring.
    a. Click Add Parameter in the operation's section.
    b. In the Name field, provide a name for your parameter.
    c. In the Located In field, select where the parameter is found in the call of your operation.
    d. Optional: In the Description field, provide a description of your parameter.
    e. Use the Required check box to specify whether the parameter is required for a call to be valid.
       Note: The purpose of setting a parameter as required is to set the `required` field for the parameter in the OpenAPI (Swagger 2.0) definition file to `true`, so that the requirement is documented for consumers of the API. The requirement is not enforced

by the IBM® API Connect Gateway, nor is it enforced by the API test tool (allowing you to verify the behavior of the API when the parameter is missing.)
However, the requirement is enforced by the Explore tool, and by the Developer Portal test tool, because the purpose of these tools is to explore the correct operation of the API.

    f. Optional: From the drop-down list for 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. For more information about creating JSON schemas, see Step 24 of Composing a REST API.

11. Add or edit any responses that you want to include.
    These responses are only for the OpenAPI (Swagger 2.0) definition of your API that is provided to developers and are not used for any purposes other than documentation.
    a. Optional: To add a response, click Add Response.
    b. In the Status Code field, provide the HTTP status code that might occur.
    c. In the Description field, provide text to be returned for the specified status code.
    d. Optional: If you have defined any JSON schemas in the Definitions section of the Design tab of your draft API, you can select to reference one of them from the drop-down menu for Schema. For more information on creating JSON schemas, see Step 24 of Composing a REST API.
12. Choose which security definitions apply to your operation. By default, an operation uses the security definitions of the API to which is belongs. You can change this by using the check boxes under Security to select individual security definition.
    Note: The Use API security definitions option implements all of the API's security definitions and must be cleared before you can select other definitions individually.
13. Clear the Use API consume types check box to override the media type with specific configurations in the operation. 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.
14. Clear the Use API produce types check box to override the media type with specific configurations in the operation. 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.
15. Click the Save icon 💾 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

Organizing your APIs and Products into categories

You can organize your APIs and Products into categories. The APIs and Products that you categorize in the API Designer or API Manager UI 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

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 Designer or API Manager UI is different to tagging in the Developer Portal. For more information on tagging, see Providing navigation by tag hierarchy.

Procedure

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon .

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon .
2. Click Drafts, then select the API or Product that you want to categorize.
   The Design window is displayed.
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 the Save icon .
6. Navigate to Drafts and select the Product that you have categorized, or the Product that contains the API that you have categorized.
7. Stage the Product by clicking the Stage icon stage icon.
8. In the API Manager UI, publish the Product that you have staged by following the steps in Publishing a new Product.

Results

You have successfully published a Product that is categorized, or has an API that is categorized.
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 a LoopBack application through the API Designer.

What to do next

In the Developer Portal, you can 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The assemble view

The API Designer 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.

After creating a policy assembly, you can access the assemble view by clicking the Assemble tab while editing a draft API. This view includes a palette, which lists 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.

Create an assembly by clicking Create assembly in the Policy assembly section of the design view of an API.

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 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. Changes that you make in the Assemble tab will be reflected in the Source tab.

API calls are made at the left, unfilled, circle, and returns at the right, filled, circle. You can insert components between the two 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 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 a Catch icon to open the property sheet of all your catches.

You can zoom the view of your canvas in and out by clicking the + and - icons. To fit the canvas to size, click the Zoom to fit icon The zoom to fit icon.

You can specify whether catches are displayed on the canvas by using the Show catches toggle in the upper right corner of the page.

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 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 pin or release the property sheet by clicking the Pin menu icon The Pin menu icon.

Related concepts

- The assemble view

Related tasks

- Creating Paths
- Adding a REST API definition
- Adding components to your assembly

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The source view

The API Designer features a source view where you can review or edit the OpenAPI (Swagger 2.0) definitions of your APIs.

When editing an API draft, you can access the Source tab. In the tab is the OpenAPI (Swagger 2.0) definition of your draft API, which you can edit at will.

OpenAPI (Swagger 2.0) definitions in API Connect can be broadly separated into two parts: a standard OpenAPI (Swagger 2.0) schema that follows all normal syntax, and extensions to the schema that are specific to API Connect.

For more information on these parts, see OpenAPI (Swagger 2.0) and assembly components and Including components in your assembly.

Changes that you make in the main body of the OpenAPI (Swagger 2.0) definition through the Source tab will be reflected in the Design tab and changes in the Design tab will be reflected in the main body of the OpenAPI (Swagger 2.0) definition in the Source tab.

Changes that you make in the Assemble tab are reflected in the API Connect assembly extension to the OpenAPI (Swagger 2.0) definition in the Source tab. Changes that you make in the assembly extension in the Source tab will be reflected in the Assemble tab.

Validation is performed as changes are made and Validation warnings The Validation warnings icon or Validation errors The Validation errors icon icons are displayed in the upper-right corner when appropriate. Click the icon to view details about the warnings or errors. Additional validation is performed when you save your API and, for certain errors, will prevent you from saving your API.

Related concepts

- The assemble view

Related tasks

- Adding a REST API definition

Related information

- Creating API definitions by using the API Designer

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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, you can add and configure components in your assembly. You can also directly add components to the OpenAPI (Swagger 2.0) definition of your API.

- **Adding components to your assembly**
  Create the assembly of your APIs by using the assemble view.
- **Handling errors in the assembly**
  Use the catch section of the assembly to describe the handling of errors thrown during the assembly execution.
- V5.0.2 + **Modifying the assembly to call an application endpoint hosted on a collective**
  In order to call the API of an application that is hosted on an API Connect collective, you need to set the host header and specify the correct URL before invoking the API of the application.
- **OpenAPI (Swagger 2.0) and assembly components**
  An assembly in 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 (Swagger 2.0) specification or extensions to the specification that are specific to API Connect.
- **The behavior of an assembly**
  The assembly executes policies in order and acts on different contexts of the API call.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding components to your assembly

Create the assembly of your APIs by using the assemble view.

About this task

You can use the assembly tool in the API Designer 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, in which case the syntax is as described in execute OpenAPI (Swagger 2.0) extension. For more information about use of the source view, see The source view.

For more information on 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. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Designer UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to work with.
5. Use the assemble view to add a component.
   a. Click the Assemble tab.
   b. If you have not already created an assembly for your API, in the canvas, click Create assembly
   c. Specify which type of gateway you want to use your API with by selecting Micro Gateway policies or DataPower Gateway policies.
      You can show and hide the radio buttons by clicking the Filter policies icon in the palette.
   d. 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 (Swagger 2.0) and assembly components.
   e. 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.
   f. 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.
6. Optional: Repeat Steps 5.d to 5.f for any additional components you want to add.
7. Click the Save icon 💾 to save your changes.

Results

You have added one or more components to your assembly.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

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. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.

The APIs tab opens.

4. Click the API definition that you want to apply a catch to and then click the Assemble tab.
5. Set the Show catches toggle in the menu bar above the palette to the Show position.
6. 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.
7. 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.
8. To add a new catch, click + Catch.
9. 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.
10. Optional: To remove an error case from a catch, click the corresponding cross.
11. Optional: To change the precedence of your catches, use 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.
12. 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 you want to apply the component to.

Results

You have created and added components to a catch for handling errors.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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
: (From API Connect version 5.0.8.3 onward)
  An error occurred while trying to access the request.

UnauthorizedError
: (From API Connect version 5.0.8.3 onward)
  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
: (From API Connect version 5.0.8.3 onward)
  The application making the API request has been disabled or is not active.

ValidateError
: (From API Connect version 5.0.8.5 onward)
  An error occurred while trying to schema validate a message payload.

V5.0.2 +

Modifying the assembly to call an application endpoint hosted on a collective

In order to call the API of an application that is hosted on an API Connect collective, you need to set the host header and specify the correct URL before invoking the API of the application.

Before you begin

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
To complete this task you must:

- have installed an API Connect collective. For more information, see Installing API Connect collective.
- DataPower Gateway only have installed and configured your DataPower Gateway to communicate with your collective. For more information, see Using physical DataPower appliances as Gateway Servers and Configuring your DataPower Gateway and API Connect collective controller to communicate.
- Micro Gateway only have installed your Micro Gateway. For more information, see Installing the Micro Gateway.
- have created and published an application. For information on creating LoopBack® applications, see Tutorials for working with LoopBack projects.
  Important: When you published your LoopBack application you must have recorded the value of host header that is displayed in the output of the command line interface when the application is published. If you did not record the value, you can find the host header value through the API Connect collective admin center, usually found on port 9443 of your controller machine. If you find the host header through the admin center, do not include the `-n` at the end of the string, as this refers to one server, instead of all instances of the application.

Procedure

To set the host headers for an invocation of your application, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. Click the API that you want to use to call the application endpoint. For information on creating REST API definitions, see Composing a REST API definition.
5. Click the Assemble tab.
6. If you have not already specified which type of gateway you want to use the API with, select DataPower Gateway or Micro Gateway as appropriate.
7. From the palette, click invoke and drag it across to the canvas. Place it at the point in the assembly that you want to invoke the endpoint.
8. Click your newly created invoke policy and configure its properties in the property sheet.
   - DataPower Gateway only If you want to invoke the endpoint directly, using the base path and Path names of the API definition you are working in, in the URL field, enter `http://ODR-DYN$(request.path)$(request.search)`.
   - Micro Gateway only If you want to invoke the endpoint directly, using the base path and Path names of the API definition you are working in, in the URL field, enter `$(runtime-url)$(request.path)$(request.search)`.
   - If you want to invoke the endpoint from an API definition other than that which was created as part of a LoopBack project, or if you want to invoke the endpoint of a different application, enter its URL in the URL field.
     Note: If you are using a DataPower Gateway, you will still need to use **`http://ODR-DYN`** for the host section of the URL.
9. From the palette, click set-variable and drag it across to the canvas. Place it immediately before your invoke policy from step 7.

10. Click your newly created set-variable policy to open the property sheet.
11. Optional: In the Title field, enter `Set Host Header`
12. Click + Action.
13. Configure the properties of the action according to the following table:

    Table 1. Set action

Property	Value
Action	Set
Set	`message.headers.host`
Type	String
Value	`Host_Header.$(api.org.name)`

    where *Host_Header* is the value that you recorded when you published your application.
14. Click the Save icon 💾 to save your changes.

Results

Your API definition can now call your application's endpoint correctly.

Related reference

- Invoke (invoke)
- Set Variable (set-variable)

Related information

- Tutorials for working with LoopBack projects
- V5.0.2 + Configuring your DataPower Gateway and API Connect collective controller to communicate

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

OpenAPI (Swagger 2.0) and assembly components

An assembly in 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 (Swagger 2.0) specification or extensions to the specification that are specific to API Connect.

The full OpenAPI (Swagger 2.0) specification can be found at the OpenAPI (Swagger 2.0) website.

For Draft APIs with an OpenAPI (Swagger 2.0) definition file, the developer toolkit 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 (Swagger 2.0) specification. The following table shows the policy and logic construct extensions that can be added, and provides links to more detailed information.

The "OpenAPI (Swagger 2.0) information" column provides links to topics detailing the OpenAPI (Swagger 2.0) implementation of different components.

The "Built-in policy and logic construct information" column provides links to the topics that detail the purpose of the different policies and logic constructs, and explain how you can configure them by using the API Designer assembly editor.

Important: If you are using IBM® API Connect for IBM Cloud, you must apply only policies that can be run on the DataPower® Gateway.

Table 1. IBM OpenAPI (Swagger 2.0) extensions

Component	Description	OpenAPI (Swagger 2.0) information	Built-in policy and logic construct information

Component	Description	OpenAPI (Swagger 2.0) information	Built-in policy and logic construct information
DataPower Gateway only activity-log	Log the activity that passes through this component in the assembly.	activity-log	Activity Log (activity-log)
DataPower Gateway only gatewayscript	Include a GatewayScript program.	gatewayscript	GatewayScript (gatewayscript)
if	Perform a part of the assembly only when a condition is fulfilled.	if	if
invoke	Call a service.	invoke	Invoke (invoke)
Micro Gateway only javascript	Include a JavaScript program.	javascript	JavaScript (javascript)
DataPower Gateway only json-to-xml	Convert payload from JSON to XML.	json-to-xml	JSON to XML (json-to-xml)
DataPower Gateway only map	Map and transform variables.	map	Map (map)
operation-switch	Perform different actions depending on which operation has been called.	operation-switch	operation-switch
DataPower Gateway only proxy	Proxy a service.	proxy	Proxy (proxy)
DataPower Gateway only redact	Redact a field from transmitted data.	redact	Redaction (redact)
set-variable	Set the value of a variable.	set-variable	Set Variable (set-variable)
throw	Throw a specified error.	throw	throw
DataPower Gateway only validate	Perform REST validation.	validate	Validate (validate)
DataPower Gateway only xml-to-json	Convert payload from XML to JSON.	xml-to-json	XML to JSON (xml-to-json)
DataPower Gateway only xslt	Apply an XSLT transform to the payload.	xslt	XSLT (xslt)

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The behavior of an assembly

The assembly executes 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 left, filled, circle to the right, 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a new version of an API definition

You can create multiple versions of an API definition and edit the versions independently.

Procedure

To create a new version of an API definition, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.

3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to create a new version of.
   The details of your API definition are displayed.
5. Click the More Actions icon ⋮ and then click Save as a new version.
   The "Save as a new version" window opens.
6. In the Version field, enter your new version number.
   Note: The version corresponds to the `info.version` property value of the API's OpenAPI (Swagger 2.0) definition. The `version.release.modification` version numbering scheme is recommended, for example `1.0.0`.
7. API Designer only V5.0.4 + Accept or change the default file name as required.
8. Click Save as a new version.

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 in the APIs tab of the Compose page of the API Designer.

Related tasks

- Adding a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding an API definition to a new Product

If you created an API definition without adding it a Product, and you later want to add the API to a new Product, you can create the Product and add the API to it in a single operation.

Procedure

To create a new Product and add an existing API definition to the Product, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to add to a new Product.
   The details of your API definition are displayed.
5. Click the More Actions icon ⋮ and then click Generate a default product.
6. API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
7. Specify a title, name, and version for the Product.
   Note: The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), and hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
8. If you want to publish the Product to a Catalog immediately, select Publish this product to a catalog and select the target Catalog.
   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 a LoopBack application through the API Designer.
9. Click Create product.
10. To work with the new Product, click All APIs > Products, then click the new Product.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.4 +

Adding an API definition to existing Products

If you created an API definition without adding it to a Product, or if you want to add an existing API definition to additional Products, you can do so while viewing the API details.

Procedure

To add an API definition to an existing Product, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to add to a Product.
   The details of your API definition are displayed.
5. Click the More Actions icon ⋮ and then click Add to existing products.
6. From the "Add to existing products" window, select one or more Products to which you want to add the API definition.
7. Click Add.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API properties

Your APIs are configured by using 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 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. API properties for the invoke, map, proxy, and rate limit 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
V5.0.6 + x-ibm-gateway-decode-request-params	V5.0.6 + No	V5.0.6 + **From API Connect version 5.0.6.2 onward:** If set to a value of `true`, any request parameters that are referenced by a variable definition on an invoke target-url are URL-decoded. This replicates the behavior from before release 5.0.6.2. From release 5.0.6.2 and beyond, the default behavior is to not decode any parameters, thereby sending them to the target URL without alteration.	V5.0.6 + Boolean
V5.0.6 + x-ibm-gateway-invoke-suppress-clientid	V5.0.6 + No	V5.0.6 + **From API Connect version 5.0.6.3 onward:** When set with a value of `true`, 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 `false`, the X-IBM-Client-Id HTTP header is no longer suppressed from being sent to the invoke target URL.	V5.0.6 + Boolean

Property	Required	Description	Data type
V5.0.8 + x-ibm-gateway-queryparam-encode-plus-char	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** If set to a value of `true`, 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 `false`. In previous releases, "+" were always encoded to "%2F". Now, the default behavior is to **not** do the encoding.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-api-enforce-response-limits	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** If set to a value of `true`, 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.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-invoke-emulate-v4-soap-error	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** 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 `true` only in the case where a gateway extension is expecting to handle a SOAP error in a post error rule. The default value is `false`. Note: **From API Connect version 5.0.8.8 onward:** This property is deprecated in favor of x-ibm-gateway-invoke-emulate-v4-invoke-error.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-invoke-keep-payload	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.5 onward:** If set to a value of `true`, 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 `false`.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-invoke-emulate-v4-invoke-error	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.8 onward:** 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 `true` 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 `false`.	V5.0.8 + 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
V5.0.7 + x-ibm-gateway-map-array-first-element-value	V5.0.7 + No	V5.0.7 + **From API Connect version 5.0.7.2 onward:** 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 `true` to only return the first array element value.	V5.0.7 + Boolean
V5.0.7 + x-ibm-gateway-map-resolve-apic-variables	V5.0.7 + No	V5.0.7 + **From API Connect version 5.0.7.2 onward:** By default, any API Connect variable that is found in the map configuration is resolved. For example, `$(request.headers.content-type)` 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 `false`. 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.	V5.0.7 + Boolean
V5.0.8 + x-ibm-gateway-map-create-empty-array	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.1 onward:** This property controls how the map policy handles the output of an empty array; it can have the following values: • `all`: Output all empty arrays, including empty children arrays. This is the default value if the property is not configured or has an invalid value. • `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.	V5.0.8 + String

Property	Required	Description	Data type
V5.0.8 + x-ibm-gateway-optimize-schema-definition	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** Set the value of this property to **`true`** 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 **`true`** 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 **`false`**, maintaining the existing behavior and performance.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-map-null-value	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.4 onward:** Set the value of this API property to **`true`** to allow a property from a map policy's input data with a value of **`null`** to be mapped to the output document. By default, a property from a map policy's input data with a value of **`null`** is not mapped to the output document.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-map-resolve-xmlinput-datatypes	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.5 onward:** 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 **`false`**, XML input elements are always mapped as a string. If you set the value to **`true`**, numeric or boolean XML input elements are mapped as the corresponding data type from the input schema. The default value is **`false`**.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-map-xml-empty-element	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.5 onward:** 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: • **`string`**: 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. • **`null`**: 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 **`true`**. • **`none`**: The empty XML element is ignored. • **From API Connect version 5.0.8.10 onward:** **`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. • **From API Connect version 5.0.8.10 onward:** **`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 API property **`x-ibm-gateway-map-null-value`** is also specified with a value of **`true`**.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-schema-definition-reference-limit	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.5 onward:** 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.	V5.0.8 + String

Property	Required	Description	Data type

V5.0.8 + x-ibm-gateway-map-emulate-v4-default-required-properties	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.7 onward:** Set the value of 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:
		• 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 set the `x-ibm-gateway-map-emulate-v4-default-required-properties` API property to `true`, 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 `x-ibm-gateway-map-emulate-v4-default-required-properties` API property is set to `true`, 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 `x-ibm-gateway-map-emulate-v4-default-required-properties` API property is set to `true`, 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 `x-ibm-gateway-map-emulate-v4-default-required-properties` API property is not specified or does not have a value of `true`, these required properties are **not** created in the output with their default values.
			V5.0.8 + Boolean

V5.0.8 + ibm-gateway-map-post-process-json-output	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.8 onward:** Set the value of this property to `true` 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 `false` for no post processing of mapped JSON output. The default value is `false`.	V5.0.8 + Boolean

Property	Required	Description	Data type

V5.0.8 + x-ibm-gateway-map-emulate-v4-empty-json-object	V5.0.8 +	V5.0.8 + **From API Connect version 5.0.8.7 onward:** 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 `true` 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 `x-ibm-gateway-map-emulate-v4-empty-json-object` API property is set to `true`, the output is as follows:
		`{`
		`  "a": {`
		`    "b": {`
		`    }`
		`  }`
		`}`
		Properties `a` and `b` are created but the value of `b` is an empty object.
		The default value is `false`.
			V5.0.8 +
			Boolean

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
V5.0.6 + x-ibm-gateway-proxy-suppress-clientid	V5.0.6 + No	V5.0.6 + **From API Connect version 5.0.6.3 onward:** A setting of `false` activates the injection of the X-IBM-Client-Id HTTP header (if it is specified on the API request), or the `client_id` query parameter in the request URL, to the proxy target-url. If set with a value of true, suppresses the sending of this header or parameter on the proxy target-url. If not specified, this header is suppressed, but the parameter is not suppressed on the proxy target-url. In previous releases, the client ID parameter was always suppressed.	V5.0.6 + Boolean
V5.0.8 + x-ibm-gateway-optimize-invoke	V5.0.8 + No	V5.0.8 + If set to `false`, prevents the replacement of the last invoke in a policy with proxy. Any value other than `false` (case insensitive) will result in the last invoke in a policy possibly being replaced by proxy when the API is executed in the gateway.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-queryparam-encode-plus-char	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** If set to a value of `true`, all "+" characters in the query parameter values of the target-url of Invoke and Proxy policies are encoded to "%2F". The default value is `false`. In previous releases, "+" were always encoded to "%2F". Now, the default behavior is to **not** do the encoding.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-api-enforce-response-limits	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.3 onward:** If set to a value of `true`, 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.	V5.0.8 + 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
V5.0.8 + x-ibm-gateway-emulate-v4-plan-rate-limit	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.4 onward:** By default in IBM API Connect Version 5.0, 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 **individually** for **each** operation in the API. To change the Version 5.0 behavior to emulate the Version 4.0 behavior, set this API property to a value of `true`.	V5.0.8 + Boolean

A list of API properties that control the behavior of multiple policies.

Table 5. Properties controlling multiple policies

Property	Required	Description	Data type
V5.0.8 + x-ibm-gateway-sourcecode-resolve-apic-variables	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.5 onward:** If set to `true`, API Connect variable references are resolved. Set to `false` if you want the policy to ignore API Connect variable references. The default value is `true`. This property applies to the following policies: - GatewayScript - XSLT - Map - if - switch Note: This property setting is overridden by the x-ibm-gateway-map-resolve-apic-variables API property setting for the Map policy.	V5.0.8 + Boolean
V5.0.7 + x-ibm-gateway-api-json-parse-error-handling	V5.0.7 + No	V5.0.7 + **From API Connect version 5.0.8.7 iFix3 onward:** If an API request or response payload includes valid JSON content that contains characters that cannot be represented in the JSONX XML internal syntax that is used by the DataPower Gateway, set this property to `escape-unicode` 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. This property applies to the API request payload, and to the API response payload when `x-ibm-gateway-api-enforce-response-limits` is enabled.	V5.0.7 + String
V5.0.8 + x-ibm-gateway-framework-preserve-escaped-reverse-solidus	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.8 onward:** By default, the string `\\` in a policy property is converted to a single `\` character. Set this property to `true` to preserve the string `\\`.	V5.0.8 + Boolean
V5.0.8 + x-ibm-gateway-inspect-request-headers	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.10 onward:** 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: - `default`: 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. - `sanitize`: Any illegal XML characters in header values are replaced with a `?` character. The API processing will continue. Any API that attempts to read `request.headers.<headername>` will see the `?` character in the value. However, the original protocol headers representing `message.headers` will still have the original character, which will be sent to an invoke or proxy backend server. - `bad-request`: 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. The default value is `default`.	V5.0.8 + Boolean

A list of API properties that control the behavior of custom policies.

Table 6. Properties controlling custom policies

Property	Required	Description	Data type
V5.0.8 + x-ibm-gateway-custom-policy-with-gws-action	V5.0.8 + No	V5.0.8 + **From API Connect version 5.0.8.4 onward:** If set to `true`, the `request.body` and `message.body` context variables will be populated for access by an `apim.getvariable('request.body')` or `apim.getvariable('message.body')` 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 `false` or do not specify it. The default value is `false`.	V5.0.8 + Boolean

- **Setting API properties**
  You define your API through the use of API properties.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Setting API properties

You define your API through the use of API properties.

About this task

API properties include property name, value, and the Catalog to which the property applies. For a list of API properties relating to invoke and proxy policies, see API properties.

Procedure

To set API properties, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to manage.
5. In the Properties section, click the Add property icon ⊕, then click the new property to expand its details.
6. In the Property Name field, enter the property name.
   Note: The property name must begin with either a letter or a _ (underscore) character.
7. In the Description field, enter a description for the property.
8. Select the Encode check box 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 Encode check box, the original property value is restored in its unencoded form.
9. In the Default Catalog's Value field, specify the default property value.
10. Optional: Add a value for another Catalog.
    a. Click Add value.
    b. API Designer only In the new value's Catalog field, enter the name of the Catalog for which you want to set the property value. This must match the name of a Catalog in API Manager and cannot be used when testing offline.
    c. API Manager only In the new value's Catalog field, select the Catalog for which you want to set the property value.
    Note: The value `apic-dev` in the list refers to the Catalog name that is reserved for testing offline in the developer toolkit. For instructions, see Configuring API definitions for container run times at ../com.ibm.apic.install.doc/tapim_migrating_to_containers.html.
    d. In the new value's Value field, enter the property value that is specific to the selected Catalog.
11. Click the Save icon 💾 to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Testing an API with the API Designer test tool

Even while you are working offline, you can test the API to ensure that is defined and implemented correctly.

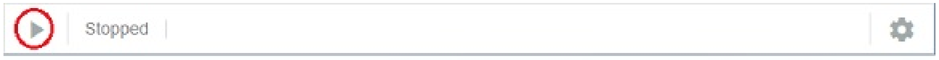
About this task

The API Designer user interface has an integrated test tool. During the test, the API is hosted on a local gateway.

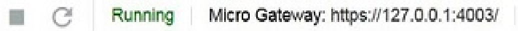
Procedure

To test an API, complete the following steps.

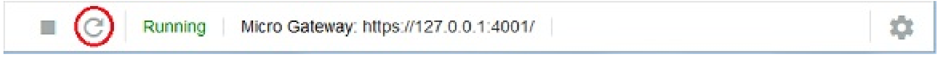
1. V5.0.0 ONLY Start the local test servers by completing the following steps:
   a. In the API Designer, click Run.
   b. Click Start to run your Micro Gateway locally and host your APIs on it. A node.exe window opens when your Micro Gateway is running; leave the window open until you are finished testing API.
      Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking Restart.
   c. Wait until the **`Running`** message is displayed.
2. V5.0.1 + Start the local test servers.In the test console at the bottom of the screen, click the Start the servers icon:

   A completed service start displays `Running` next to the gateway type, URL, and the port where it is available.

   Your project configuration and other running processes can produce a different gateway type, URL, or port number than is displayed in the image.
   Test changes that are for a running gateway by clicking the Restart the servers icon

   to start again with the new settings.
3. Click the APIs tab.
4. Click the name of the API that you want to test.
5. Click the Assemble tab.
6. Click the Test icon ▶.
7. Choose the Operation (endpoint) to test.
8. Provide values for the parameters you set in your operation with the Parameters section of the test tool, or click Generate to create random values based on the data type that is set for each parameter.
   Note: The test tool does not enforce the **required** setting for any parameters in the definition of the operation.
9. Optional: If you want to run the test multiple times, complete the following steps:
   a. Select the Repeat check box.
   b. Use the Stop after field to specify how many times to run the test.
   c. If you want the test to stop when an error occurs, select the Stop on error check box.
10. Click Invoke.
    The test result is displayed in the Response section. You can continue to test different field values as necessary.
    V5.0.8 + Note: Modern web browsers prevent local resources from being accessed by the API Designer, and display a CORS (cross origin resource sharing) alert on the page. To enable local API testing in CORS compliant browsers, select the Enable Proxy check box to send test messages from the local server that hosts API Designer rather than from the browser. The proxied request appears as in the example.

    ```
    Request URL: http://localhost:port/proxy/proxyService
    Request Method: POST
    Request Payload: The test content
    ```

    Responses are returned directly to the user interface. For more information about CORS, see: https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS.

Results

You successfully tested an API.
Note: You can also use the API Explore tool within API Designer to test API endpoints. Ensure that your local test servers are running, then click Explore. The API Explore tool opens, and shows the operations, definitions, and documentation for all of the APIs that are contained in your Drafts view. The left pane of the Explore window can be used to select an operation to test. The center pane displays summary information about the endpoint, including its parameters, model instance data, and response codes, and the right pane provides template code to call the endpoint.

Related tasks

- Configuring an operation
- Testing APIs with the IBM DataPower Gateway
- Creating a Product
- Staging a Product

Related information

- Creating and configuring Catalogs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Designer UI)

When you configure a REST or SOAP API definition, you can use an OpenAPI (Swagger 2.0) extension to go beyond the standard OpenAPI (Swagger 2.0) schema. You can also replace an extension with an updated version.

About this task

To make an extension available for adding to an API definition, create a schema definition file (in YAML or JSON format) for the extension in your project folder on your file system. Then use the API Designer user interface to import the extension into an API definition. If you wish to use the command-line tool, there is also a corresponding `apic extensions` command to publish and manage extensions in a catalog.

When you add an extension to an API definition, you can set values for any of the properties that are defined in the schema definition file for the extension. The values that you specify are subject to any validation criteria that are defined in the schema definition.

The following example shows an extension schema 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
```

For more information on extensions, see OpenAPI (Swagger 2.0) Swagger Extensions.

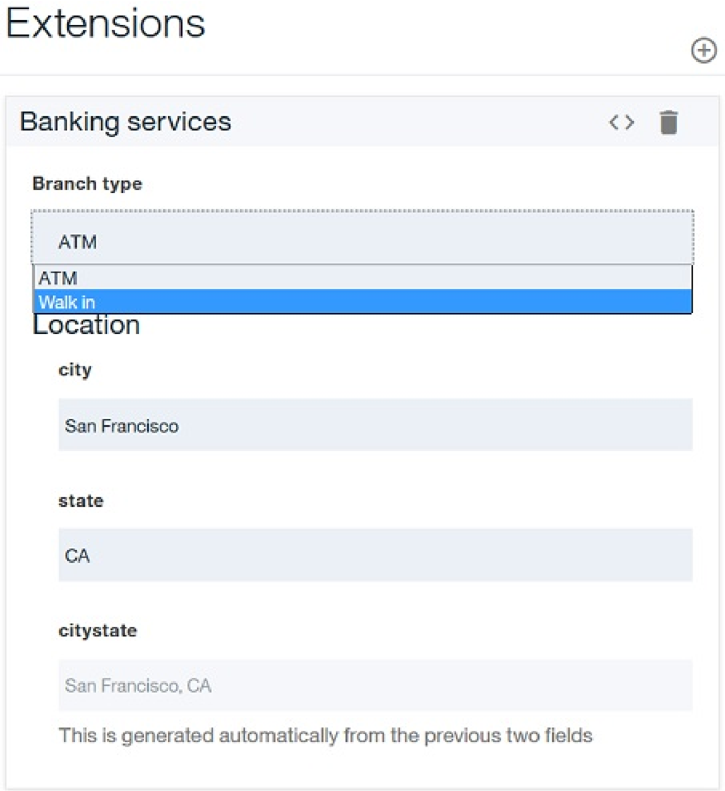
Procedure

To add an extension schema to an API definition, complete the following steps:

1. To open the API Designer user interface (UI), enter the following command at the command line:

   ```
   apic edit
   ```

   The API Designer UI opens in your default browser.
   Note: If you are connected to a network, you are asked to sign in with IBM Cloud. Enter your IBMid credentials and click LOG IN. If this is your first time using API Designer, the Draft APIs information page opens, click Got it!, and continue with the following steps.
2. In the API Designer, click the APIs tab.
3. Click the API that you want to work with.
   The API details page opens.
4. Navigate to the Extensions section, and click the Add Extension icon ⊕.
   The Add Extension window opens. The window lists the extension schema YAML files that are in your project folder.
5. Select the extension that you want to add, then click Done.
   You can enter a search string to locate the required extension.
6. To set a value for any of the properties that are defined in the schema, specify the value in the appropriate field. Any validation criteria that are defined in the schema are applied to the value that you specify. The following screen capture shows the property setting pane for the previous bank branch example:



Extensions

Banking services

Branch type

ATM

ATM
Walk in

Location

city

San Francisco

state

CA

citystate

San Francisco, CA

This is generated automatically from the previous two fields

7. To replace an extension with a new version, add an extension that has the same value for the **name** property as an extension that has already been added, but a different value for the **version** property. The Change Version of Extension window opens; click OK to replace the existing version with the new version, or Cancel to retain the existing version. For example, for the bank branch extension described earlier, the **Branch type** property might have an additional type added to the enumeration list.

Related tasks

- Referring to an extension in an API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Referring to an extension in an API definition

When using the command-line tool you must manually add a reference to the extension to your API definition YAML (or JSON) file in the extensions key under x-ibm-configuration.

Procedure

1. Edit the API definition YAML file and add the reference. Using the above example extension:

   ```
   x-ibm-configuration:
     ...
     extensions:
       banking: 1.0.0
   ```

2. (Optional) Create a YAML or JSON file containing the property definition. For example, create a file named extension.yaml:

   ```
     Branch type: ATM
     location:
       city: San Francisco
       state: CA
       citystate: 'San Francisco, CA'
   ```

3. In your API definition file, add a property whose name must begin with `x-` and references the property extension you created. The following example uses the banking extension:

   ```
   ...
   tags: []
   x-banking:
     $ref: extension.yaml
   ```

   If you don't want to create an extension file, add the property explicitly; for example:

   ```
   ...
   tags: []
   x-banking:
     location:
       properties:
         city:
           type: string
         state:
           type: string
         citystate:
           type: string
   ```

Related tasks

- Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Designer UI)

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

For more information, see Setting API properties.

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:
  - DataPower Gateway only Through the GatewayScript policy, run the **`apim.getvariable()`** API.
  - DataPower Gateway only 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.
  - DataPower Gateway only Through the GatewayScript policy, use the **`apim.setvariable()`** API.
  - DataPower Gateway only 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 and JavaScript policies. For more information, see GatewayScript (gatewayscript) and JavaScript (javascript).
- **`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 (invoke) or Proxy (proxy).
- A Map policy. For more information, see Map (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.

Related concepts

- The behavior of an assembly

Related reference

- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 SOAP proxy API; see Adding a SOAP API definition by using a WSDL file.
- Adding a target WSDL service to an API definition; see Adding an existing web service to your 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
`suppressExamples`	`true` or `false`	`false`	Suppress auto-generation of examples.
`wssecurity`	`true` or `false`	`true`	Enable generation of definitions for WS-Security headers.
`implicitHeaderFiles`	Array of XSD file locations		Additional schema files, not referenced in the WSDL, that are used to define additional SOAP headers.
V5.0.8 + `port`	V5.0.8 + The `name` attribute for a `wsdl:port` in a `wsdl:service` definition.	V5.0.8 +	V5.0.8 + 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 `port` field is present in the options file, API Connect creates the API by using the first SOAP port that it finds in the `wsdl:service` definition.

Example

```
suppressExamples: true
wssecurity: false
implicitHeaderFiles:
  - xsdDir/schema1.xsd
  - xsdDir/schema2.xsd
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API policies and logic constructs

Policies and logic constructs are a 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 in the API Designer.

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 Designer 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 (Swagger 2.0) definition file. For more information, see IBM extensions to the OpenAPI (Swagger 2.0) 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 Designer 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 (Swagger 2.0) definition file. For more information, see IBM extensions to the OpenAPI (Swagger 2.0) 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 IBM 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 User-defined policies.
Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), support for implementing your own policies is *not* available.

Information about API policies and logic constructs can be found in the following topics:

- **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.
- **User-defined policies**
  Make a user-defined policy available to a Catalog, and apply that policy to a REST or SOAP API.

Related reference

- execute

Related information

- Creating API definitions by using the API Designer
- Including components in your assembly

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 and the Micro Gateway, some policies are restricted to a particular Gateway. The following icons indicate which Gateway each policy can be used with:

- DataPower Gateway Indicates that the policy can be run on the DataPower Gateway.
- Micro Gateway Indicates that the policy can be run on the Micro Gateway.

Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
Important: If you are using IBM API Connect for IBM Cloud, you must apply only policies that can be run on the DataPower Gateway.
Built-in policies are configured in the context of an API. You can use the API Designer 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 (Swagger 2.0) definition file. For more information, see Creating an OpenAPI (Swagger 2.0) definition file.

The following table shows the list of built-in policies that are available, and whether they are restricted to a particular Gateway or are available on both. The table contains links to configuration information for both the built-in policy definitions, and the OpenAPI (Swagger 2.0) policy definitions. The policies are the same, but they are created in different ways.

Table 1. Built-in policies

Built-in policy	OpenAPI (Swagger 2.0) policy	Description	DataPower Gateway	Micro Gateway
Activity Log[1]	activity-log	Use the Activity Log policy 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.	✓	✗
GatewayScript	gatewayscript	Use the gatewayscript policy to execute a specified DataPower GatewayScript program.	✓	✗
V5.0.5+ Generate LTPA Token	V5.0.5+ ltpa-generate	V5.0.5+ Use the Generate LTPA Token security policy in IBM API Connect to generate a Lightweight Third Party Authentication (LTPA) token.	V5.0.5+ ✓	V5.0.5+ ✗
Invoke	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.	✓	✓
JavaScript	javascript	Use the JavaScript policy to execute a specified JavaScript program.	✗	✓
JSON to XML	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.	✓	✗
V5.0.1+ Generate JWT	jwt-generate	Use the Generate JWT security policy in IBM API Connect to generate a JSON Web Token (JWT).	✓	✗
V5.0.1+ Validate JWT	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.	✓	✗
Map	map	Use the Map policy to apply transformations to your assembly flow and specify relationships between variables.	✓	✗
Proxy	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.	✓	✗
Redaction	redact	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.	✓	✗
Set Variable	set-variable	Use the Set Variable policy to set a runtime variable to a string value, or to clear a runtime variable, or to add a header variable.	✓	✓
Validate	validate	Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.	✓	✗
V5.0.3+ Validate	V5.0.3+	V5.0.3+ Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema. Micro Gateway You can now also use the validate policy with the Micro Gateway to validate the payload in an assembly flow against a JSON schema.	V5.0.3+ ✓	V5.0.3+ ✓
V5.0.2+ Validate Username Token	validate-usernametoken	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.	✓	✗
XML to JSON	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).	✓	✗
XSLT	xslt	Use the XSLT policy to apply an XSLT transform to the payload of the API definition.	✓	✗

Related tasks

- Including components in your assembly

1

Note: The Micro Gateway does not support the Activity Log policy. However, the Micro Gateway does collect the basic analytic statistics. The statistics that the Micro Gateway gathers are equivalent to what an Activity Log policy in the DataPower Gateway with `Content:activity` settings gathers with some exceptions:

- For the following fields, the Micro Gateway does not collect the information and sends empty payload: **`requestHttpHeaders`**, **`responseHttpHeaders`**, and **`debug`**.
- When the Micro Gateway starts with an APIMANAGER environment variable that specifies a valid Management server, the Micro Gateway automatically collects the basic analytic statistics. There is no mechanism to turn the collection function on or off at runtime.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Activity Log (activity-log)

Use the Activity Log policy 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.

Restriction:

- The Activity Log policy can be used only with the DataPower® Gateway.
- Activity log policies that call for logging of Analytics data upon success do not apply for the OAuth provider API. The OAuth provider API logs Analytics data for failure cases, but does not log successful cases.

Note: The Micro Gateway does not support the Activity Log policy. However, the Micro Gateway does collect the basic analytic statistics. The statistics that the Micro Gateway gathers are equivalent to what an Activity Log policy in the DataPower Gateway with `Content:activity` settings gathers with some exceptions:

- For the following fields, the Micro Gateway does not collect the information and sends empty payload: **`requestHttpHeaders`**, **`responseHttpHeaders`**, and **`debug`**.
- When the Micro Gateway starts with an APIMANAGER environment variable that specifies a valid Management server, the Micro Gateway automatically collects the basic analytic statistics. There is no mechanism to turn the collection function on or off at runtime.

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

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 1. Activity Log policy properties

Property label	Required	Description	Data type
Title	Yes	A title for the policy is required, but a default value, **`activity-log`** is provided.	string
Description	No	A description of the policy.	string

Property label	Required	Description	Data type
Content	Yes	Defines the type of content to be logged when the operation is successful. Valid values: - **none**: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal. - **activity**: Logs invocation only (only the resource URI is recorded). - **header**: Logs activity and header. - **payload**: Logs activity, header, and payload (the original request, if any, and the final response). The default value is **activity**.	string
Error content	No	Indicates what content to log if an error occurs. Valid values: - **none**: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal. - **activity**: Logs invocation only (only the resource URI is recorded). - **header**: Logs activity and header. - **payload**: Logs activity, header, and payload (the original request, if any, and the final response). The default value is **payload**.	string

Related tasks

- Composing a REST API definition

Related information

- API Analytics

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

GatewayScript (gatewayscript)

Use the gatewayscript policy to execute a specified DataPower® GatewayScript program.

Restriction: The gatewayscript policy can be used only with the DataPower Gateway.

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 1. gatewayscript policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is `gatewayscript`.	string
Description	No	A description of the policy.	string

Source	Yes	The GatewayScript source code to execute. For example:
		`var message = [ 'Hello', 'World!' ];`
		`console.debug(message.join(' '));`
			string

Examples

The following examples show how the full OpenAPI (Swagger 2.0) 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.

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

- Composing a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

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 default common name `apim` to call the methods. You can change the common name to one of your choice by using the function `var name = require('./apim.custom.js');`; for example, setting

  ```
  var apic = require('./apim.custom.js');
  ```

  would set the calls to start with the common name `apic`. However, the use of `require()` adds latency.
- If you are using GatewayScript within a user-defined policy, there are additional configuration requirements to those listed here. For example, the output from a user-defined policy must be an XML node-set or a JSONx message. For information about how to create an implementation for a user-defined policy, see Implementing your policy, and for example GatewayScript code snippets, see Implementation code examples.

Access the assembly context

The following code snippets show examples of how to access the assembly context. The examples use the default common name of `apim`. 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.

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');
```

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.

V5.0.8 +

(From API Connect version 5.0.8.8 onward) 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` and you use `apim.output` to set the output type, you must call `apim.setvariable` **before** `apim.output`.

DataPower Gateway only V5.0.5 +

Generate LTPA Token (ltpa-generate)

Use the Generate LTPA Token security policy in IBM® API Connect to generate a Lightweight Third Party Authentication (LTPA) token.

Restriction:

- The Generate LTPA Token policy is deprecated and is not supported in API Connect releases later than Version 5.
- The Generate LTPA Token policy can be used only with the DataPower® Gateway.

About

Lightweight Third Party Authentication (LTPA) is an IBM protocol that provides a cookie or binary security token based authentication mechanism in WebSphere® Application Server. It supports single sign-on (SSO) technology, and is intended for distributed, multiple application server and machine environments.

Apply the Generate LTPA Token policy to your assembly so that your API can securely authenticate with applications or services that are hosted onWebSphere Application Server. At run time, the LTPA token that is generated by the policy is sent to the WebSphere Application Server back-end services, either in an HTTP cookie header (the default option), or for SOAP and XML payloads the token is wrapped in a WS-Security header inside a SOAP message.

Note: Lotus® Domino® (Domino) token types and keys are not supported in IBM API Connect. However, you can configure Lotus Domino to accept a WebSphere Application Server LTPA token, and then a Generate LTPA Token policy can be used to authenticate with Lotus Domino. You can attach this policy to the following API flows:

- REST
- SOAP

Prerequisites

The following prerequisites apply:

- Before you can apply a Generate LTPA Token policy to your API definition, an LTPA key must be imported from the LTPA peer (that is, the WebSphere Application Server) into API Manager. For more information, see LTPA keys.
- The minimum level of IBM DataPower is Version 7.5.1.

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 1. Generate LTPA Token policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is `ltpa-generate`.	string
Description	description	No	A description of the policy.	string

Property label	Property name	Required	Description	Data type
LTPA Key	key	Yes	The name of the LTPA key that you want to use to generate the LTPA token. API Designer only Enter the name of the LTPA key by using one of the following syntax options: • *`my-ltpa-key`* - entering the name with no version number means that at run time the policy selects version 1.0.0 of the LTPA key. • *`my-ltpa-key:2.0.0`* - entering the name with a version number means that at run time that specific version of the key is used. • *`my-ltpa-key:latest`* - entering the name with **`latest`** means that at run time the policy selects the latest version of the LTPA key to use. API Manager only Select the LTPA key from the drop-down menu. Each LTPA key has a non-version specific option, for example my-ltpa-key, latest version. Select this option if you want the policy at run time to select the latest version of the LTPA key to use. Otherwise, select a specific version. Note: The automatic version selection feature relies on the LTPA key being configured with a version number that conforms to the version.release.modification version numbering scheme.	LTPAKey
Authenticated User Name	authenticatedUserName	Yes	The runtime variable that contains the authenticated user name. The LTPA token is generated with this property as the user. For example, if the API is configured with a basic authentication security definition, then the authenticated user name can be specified as **`$(client.app.id)`**. If the API is configured with an OAuth security definition, then the authenticated user name can be specified as **`$(oauth.resource-owner)`**. Alternatively, before the Generate LTPA Token policy, you can configure a set-variable policy to set a runtime variable with a particular user name, and then specify this runtime variable as the authenticated user name.	string
Token Version	tokenVersion	Yes	The version of the LTPA token. Select from the following values: • WebSphereVersion1 • WebSphereVersion1-FIPS • WebSphereVersion2 • WebSphere70Version2 The default value is **`WebSphereVersion2`**.	string
Token Output	tokenOutput	Yes	Define where in the output source the policy should place the generated LTPA token. Select from the following options: • In Cookie Header • In WSSec Header[1] The default option is In Cookie Header. Note: In WSSec Header should be selected only if the message has an XML or SOAP media type.	enum
Token Expiry	tokenExpiry	Yes	The length of time (in seconds) that is added to the current date and time, in which the LTPA key is considered valid. The default value is **`600`**.	integer

Example

```
- ltpa-generate:
    title: ltpa-generate
    tokenVersion: WebSphereVersion2
    tokenOutput: in-cookie-header
    tokenExpiry: 600
    key: 'my-first-ltpa-key:1.0.0'
```

Errors

The following error can be thrown while the policy is being executed:

- `LTPAGenerateError` - an error that captures all the errors that occur during the execution of the policy. Upon failure, the detailed error message is assigned to the runtime variable ltpa-generate.error-message, so it can be retrieved via catch.

If a catch is not configured, in the case of a failure the Generate LTPA Token policy returns an HTTP code 500 failure. The detailed error message can be found in the system log.
Tip: If there is an error, make the following checks:

- Verify that the IBM DataPower firmware is at Version 7.5.1 or later.
- Check that the password that is set in the LTPA key is correct.
- If In WSSec Header is selected for the Token Output property, verify that the payload of the message contains an XML or SOAP media type.

Related information

- LTPA
- LTPA versions and token formats
- Single sign-on for authentication using LTPA cookies

[1] If the In WSSec Header option is selected, the following conditions apply:

- If the input is XML, the policy creates a SOAP envelope, and places the LTPA token in the SOAP security header, and places the input XML in the SOAP body.
- If the input is SOAP, but without a SOAP security header, the policy creates a SOAP security header and places the LTPA token in this header. The rest of the SOAP message is untouched.
- If the input is SOAP and there is already a binary security token in the SOAP security header, the policy overwrites the existing token with the newly generated LTPA token. The rest of the SOAP message is untouched.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Invoke (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.

About

You can attach this policy to the following API flows:

- REST
- SOAP

V5.0.8 + From IBM® API Connect V5.0.8.0 and beyond, 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 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.

Note that a proxy policy ignores the Stop on error property, and that a replacement with a proxy does not occur if you have configured any catches on the invoke policy. For more information about the proxy policy, see Proxy (proxy). 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.

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.

Note: The Invoke policy does not support responses with multipart form data, that is, when the response is set to `Content-Type: multipart/related`.

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 1. Invoke policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is **`invoke`**.	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. V5.0.2 + Additional configuration is required to call applications that are hosted on collectives. For more information, see Modifying the assembly to call an application endpoint hosted on a collective.	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 **`60`**.	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: • Keep Note: This is only a valid option when you are using it with an IBM DataPower appliance. If you are using microgateway, you must use a different value. • GET • POST • PUT • DELETE • PATCH • HEAD • OPTIONS The default value is **`GET`**. However, if set to **`Keep`**, 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
DataPower Gateway only Cache Type	No	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: Protocol 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. 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. No Cache 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. Time to Live 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. The default value is Protocol.	string
DataPower Gateway only 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 `Time to Live`. Enter a value in the range 5 - 31708800. The default value is `900`.	integer
DataPower Gateway only 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 this check box to stop the flow and trigger a catch flow when a particular type of error is thrown. The following list shows the type of errors that can be selected: • ConnectionError • SOAPError • OperationError The check box is selected by default, but you must also select one or more error types in the search errors field, or the flow will not stop and no catch flows will be triggered. If the check box is not selected, or no error types are selected, the flow will continue when an error is thrown during the policy execution.	boolean

Property label	Required	Description	Data type
DataPower Gateway only Response object variable	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 *message*. Use this property to specify an alternate location to store the invoke response. This variable can then be referenced in other actions, such as Map. Note: If you want the response to be saved in *message*, leave the Response object variable property blank, do **not** supply the value `message`.	string

Example

```
- invoke:
  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
```

Related tasks

- Composing a REST API definition
- Handling errors in the assembly

Related information

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Micro Gateway only

JavaScript (javascript)

Use the JavaScript policy to execute a specified JavaScript program.

Restriction: The JavaScript policy can be used only with the Micro Gateway.
Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

About

You can attach this policy to the following API flows:

- REST
- SOAP

You can use the JavaScript policy to execute a snippet of JavaScript code, however, the policy has the following limitations:

- The **`require()`** parameter is not available.
- Global objects of node.js are not available.
- The use of strict mode (**`"use strict"`**) inside the JavaScript code is not supported. Therefore, block-scoped declarations cannot be used.

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 1. JavaScript policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `javascript`.	string
Description	No	A description of the policy.	string
Source	Yes	The JavaScript source code to execute.	string

Examples

The following snippet of JavaScript code shows how the properties of a context object can be accessed or modified directly:

```
if (request.verb === 'POST') {
  //perform some business logic when the request is POST
}
```

The following example shows how to throw an error object that contains the error information, and changes the flow after:

```
if (request.body.order === undefined) {
  throw { name : 'IncorrectOrder', message: 'the payload should contain valid order' };
}
```

The following example shows how the error object from the previous example can be caught by a `catch` assembly:

```
catch:
  - errors:
      - 'IncorrectOrder'
    execute:
      - set-variable:
        actions:
          - set: 'message.body'
            value: '{ "error" : "found an incorrect order" }'
```

Errors

The following errors 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.
- A custom error.

Related tasks

- Composing a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

JSON to XML (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.

Restriction: The JSON to XML policy can be used only with the DataPower® Gateway.

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 API Designer 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`.

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 1. Policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `json-to-xml`.	string
Description	No	A description of the policy.	string
Root XML Element Name	Yes	The root element name of the resultant XML document. This property is only used 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 `json`.	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 `false`.	boolean

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.1+

Generate JWT (jwt-generate)

Use the Generate JWT security policy in IBM® API Connect to generate a JSON Web Token (JWT).

Restriction: The Generate JWT policy can be used only with the DataPower® Gateway.

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:

- IBM API Connect Version 5.0.1 or later.
- 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 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 1. Generate JWT policy properties

Property label	Property name	Required	Description	Data type
Title	title	No	The title of the policy. The default value is **`jwt-generate`**.	string
Description	description	No	A description of the policy.	string
JSON Web Token (JWT)	jwt	No	Runtime variable in which to place the JWT that is generated. The default value is: **`generated.jwt`**. However, if not set, the JWT that is generated is written to the Authorization Header as a Bearer token.	string
JWT ID Claim	jti-claim	No	Indicates whether a JWT ID (jti) claim should be added to the JWT. If selected, the property is set to **`true`**, and a UUID is generated and set as the JTI claim value.	boolean
Issuer Claim	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: **`iss.claim`**	string
Subject Claim	sub-claim	No	Runtime variable from which the Subject (sub) claim string can be retrieved.	string
Audience Claim	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
Validity Period	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 **`3600`**.	integer

Property label	Property name	Required	Description	Data type
Private Claims	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	jws-jwk	No	Runtime variable that contains the JWK that is used to sign the JWT.[1]	string

Cryptographic Algorithm	jws-alg	No	The cryptographic algorithm to use. Valid values are:
			- HS256
			- HS384
			- HS512
			- RS256
			- RS384
			- RS512
			- ES256
			- ES384
			- ES512
			- V5.0.8 + PS256
			- V5.0.8 + PS384
			- V5.0.8 + PS512
			Note: The following algorithms are not available in the drop-down list and must be added to the OpenAPI source manually:
			- PS256
			- PS384
			- PS512
			For example:
			`- jwt-generate:`
			`    title: jwt-generate`
			`    iss-claim: iss.claim`
			`    exp-claim: 3600`
			`    version: 1.0.0`
			`    jws-alg: PS256`
			For more information on specifying the OpenAPI source for the Generate JWT policy, see jwt-generate.
				string

Sign Crypto Object	jws-crypto	No	The cryptographic object to use to sign the JWT.[1]	string
Encryption Algorithm	jwe-enc	No	The encryption algorithm to use. Valid values are: - A128CBC-HS256 - A192CBC-HS384 - A256CBC-HS512	string
Encrypt JWK variable name	jwe-jwk	No	Runtime variable that contains the JWK to use to encrypt the JWT.	string
Key Encryption Algorithm	jwe-alg	No	The key encryption algorithm to use. Valid values are: - RSA1_5 - RSA-OAEP - RSA-OAEP-256 - dir - A128KW - A192KW - A256KW	string
Encrypt Crypto Object	jwe-crypto	No	The cryptographic object to use to encrypt the claim.	string

Example

```
- jwt-generate:
    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

[1] 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.1+

Validate JWT (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.

Restriction: The Validate JWT policy can be used only with the DataPower® Gateway.

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.

Prerequisites

The following prerequisites apply:

- IBM® API Connect Version 5.0.1 or later.
- 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) 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 1. Validate JWT policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is **`jwt-validate`**.	string
Description	description	No	A description of the policy.	string
JSON Web Token (JWT)	jwt	Yes	Context or runtime variable that contains the JWT to be validated. The default value is: **`request.headers.authorization`**. However, if this property is not set, the policy looks for the JWT in the **`request.headers.authorization`** location by default. Note: The format of the authorization header must be: **`"Authorization: Bearer jwt-token"`** where *jwt-token* is the encoded JWT.	string
Output Claims	output-claims	Yes	Runtime variable to which the full set of claims that are contained in the JWT is assigned. The default value is: **`decoded.claims`**.	string
Issuer Claim	iss-claim	No	The Pearl Compatible Regular Expression (PCRE) to use to validate the Issuer (iss) claim.	string
Audience Claim	aud-claim	No	The PCRE to use to validate the Audience (aud) claim.	string
Decrypt Crypto Object	jwe-crypto	No	The cryptographic object (a shared key or certificate) to use to decode the claim.[1]	string
Decrypt Crypto JWK variable name	jwe-jwk	No	Runtime variable that contains the JWK to use to decrypt the JWT.[1]	string
Verify Crypto Object	jws-crypto	No	The cryptographic object (a shared key or certificate) to use to verify the signature.[2]	string
Verify Crypto JWK variable name	jws-jwk	No	Runtime variable that contains the JWK to use to verify the signature.[2]	string

Example

```
- jwt-validate:
    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.

Related information

- Introduction to JSON Web Tokens

[1] 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.
[2] A JWK 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Map (map)

Use the Map policy to apply transformations to your assembly flow and specify relationships between variables.

Restriction: The Map policy can be used only with the DataPower® Gateway.

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 1. 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 **`map`**.	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

- Composing a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The Map policy structure

The Map policy uses a structure within its OpenAPI (Swagger 2.0) 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: With the exception of client ID and client secret, the passing of form input as a parameter into an API is not supported.

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.

V5.0.4 + **`options`**

V5.0.4 + Applicable to the output XML of the Map policy. The following property options are available for you to select:

- Include empty XML elements. Select this option to control whether empty elements are produced for non-required elements that have no mapped value.
- Inherit XML namespaces. Select this option to inherit the namespace from the parent element rather than specifying it on every required element. Specifying a namespace of `null` or `''` indicates no inheritance from the parent element.
- Inline namespace declarations. 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.
- V5.0.8 + Severity level for input data log messages (From API Connect Version 5.0.8.7); This property specifies the severity level for log messages that relate to input data. The following choices are available:
  - error
  - warn
  - info

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 `$(0)`. In each iteration, the current value of *out.total* and the current array entry of *in.input* are summed, and the value of *out.total* is set as this summation.
V5.0.1+ When using a foreach to operate on an array, if the elements of the array do not have named fields, you can use `$(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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the Map policy in the user interface

The assemble view in API Designer 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. Optional: Provide a Title and Description for your Map policy.
3. Click the Edit inputs icon ✎ in the Input column.
4. 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
      - V5.0.4+ Generate from sample JSON
      - V5.0.4+ Generate from sample XML

      .
      V5.0.3+ 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.

5. Optional: To remove a variable, click the corresponding Remove input icon 🗑.
6. After you have added all of your input variables, click Done.
7. Click the Edit outputs icon ✎ in the Output column.
8. 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
   - V5.0.4+ Generate from sample JSON
   - V5.0.4+ Generate from sample XML

   V5.0.3+ 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.

9. Optional: To remove a variable, click the corresponding Remove output icon 🗑 .
10. After you have added all of your output variables, click Done.
11. Optional: V5.0.3+ 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.
12. 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.
13. 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.
14. 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.
    V5.0.3+ 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.
15. 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.
16. 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.
17. Optional: To delete all mappings to the output, click Delete.
18. When you have configured your outputs, click OK.
19. Optional: V5.0.4+ To control the XML output of the map policy, click the Settings icon in the Map column.
    a. In the Advanced XML options section, select one or both of the following options:

       Include empty XML elements
          This option is selected by default and means that 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.

       Inherit XML namespaces
          This option is selected by default and means that XML namespaces are inherited from the parent element. Clear the check box if you want the map policy to use explicit namespaces.

       Inline namespace declarations
          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.

       Note: These options effect the XML output only and have no effect on the JSON data.
    b. V5.0.8+ (From API Connect Version 5.0.8.7) 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
    c. Click Done.

Results

You have configured a Map policy to transform and map variables in your assembly flow.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Map policy examples

Examples of the OpenAPI (Swagger 2.0) 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 is used
                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'
```

V5.0.4 +

Using the advanced XML options

The following example:

- includes empty elements in the XML output for every element in the schema that has no mapped value.
- indicates that all namespace declarations will be placed on the root XML element

```
        actions:
            - set: output, two
              from: input, one
        options:
            includeEmptyXMLElements: false
            inlineNamespaces: false
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Proxy (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.

Restriction: The Proxy policy can be used only with the DataPower® Gateway.

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.

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM® Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.

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 1. Proxy 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 `proxy`.	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. V5.0.2 + Additional configuration is required to call applications that are hosted on collectives. For more information, see Modifying the assembly to call an application endpoint hosted on a collective.	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 `60`.	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: • Keep • GET • POST • PUT • DELETE • PATCH • HEAD • OPTIONS The default value is `Keep`. By using `Keep`, 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	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: Protocol 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. 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. No Cache 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. Time to Live 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. The default value is Protocol.	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 `Time to Live`. 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
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

Property label	Required	Description	Data type

X-Forwarded header	No	This header can be provided by
		1. If
		`X-Forwarded-Host`
		exists, processing continues. If it does not exist prior to calling the proxy policy, it is set with the value of the `Host` header.
		2. The
		`X-Forwarded-For`
		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.
		3. If all three of
		`X-Forwarded-Host`
		,
		`X-Forwarded-Port`
		, and
		`X-Forwarded-Proto`
		headers are missing at the time of calling the proxy policy, they are set automatically. To prevent this, set
		`X-Forwarded-Host`
		header to some value before calling the proxy policy.
			string

Related tasks

- Composing a REST API definition

Related information

- TLS profiles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Redaction (redact)

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.

Restriction: The Redaction policy can be used only with the DataPower® Gateway.

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 1. Redaction policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is `redact`.	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 Constructing XPath expressions to redact fields.	string
Action	Yes	Specifies whether you want to remove or redact the field. Valid values: • `remove`: Completely removes the specified field. • `redact`: Redacts (obfuscates with "*"s) the field to block out the data. The default value is `redact`. Note: If a numerical value is being redacted, the redacted value is depicted as `******` and the type is changed to `string`.	string
From	Yes	Specifies where to remove or redact the specified field from. Valid values: • `all`: Removes or redacts the specified field from the Request body, the Response body, and the activity logs. • `request`: Removes or redacts the specified field from the Request body. • `response`: Removes or redacts the specified field from the Response body. • `logs`: Removes or redacts the specified field from the activity logs. The default value is `all`. Optionally click Add item to specify additional values.	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

- Composing a REST API definition

Related reference

- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Constructing XPath expressions to redact fields

To define a field for redaction, 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
`//`	search anywhere in the XML structure
`//streetAddress`	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
`//*`	find any element anywhere in the structure
`//*[@name='secondaryAddress']`	find an element anywhere in the structure that is named "secondaryAddress"

Expression	Meaning
`//*[@name='secondaryAddress']/*[@name='streetAddress']`	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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Set Variable (set-variable)

Use the Set Variable policy to set a runtime variable to a string value, or to clear a runtime variable, or to add a header 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 1. Set Variable policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is **`set-variable`**.	string
Description	No	A description of the policy.	string
Action	Yes	Defines what action to apply on a runtime variable. Valid values: • **`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. • **`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. • **`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. The default value is **`Set`**.	string

Property label	Required	Description	Data type
Variable name	Yes	Specifies the name of the variable that you want to set to a string value, or that you want to add or clear.	string
Value	Yes*	Allocates this value to the specified variable. Can be a literal value, or another variable. * Value is required only when **`Set`** or **`Add`** is specified as the action. For example, to *set* a named variable of **`billing-hostname`** to a literal value, you can specify the Value as **`acme.com`**. As another example, to *set* a named variable to the value of the Content-Type header in a request, you can specify the Value entry as **`$(request.headers.content-type)`**. Note: You can only set single string elements. Values are retrieved as strings and therefore you cannot clone a complete nodeset.	string

Related tasks

- Composing a REST API definition

Related reference

- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Validate (validate)

Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.
Restriction:

- The schema that represents the XML can reference only one XML namespace.
- The schema cannot reference polymorphic XML elements.
- The OpenAPI **`discriminator`** field is **not** supported by the Validate policy.
- The Validate policy can be used only with the DataPower® Gateway, not with the Micro Gateway.
- Validate 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.

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 (Swagger 2.0) 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 1. Validate policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `validate`.	string
Description	No	A description of the policy.	string
Definition	Yes	Specifies the schema definition to be used for validating the payload. Valid values: • `request`: 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 Configuring an operation. • `response`: 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 Configuring an operation. • `#/definitions/`*`definition_name`*: Select this value for a previously defined schema to be used to validate the payload that is returned from other invoke actions or tasks in the assembly flow.	string

Related tasks

- Defining Paths for a REST API
- Composing a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.2 +

Validate Username Token (validate-usernametoken)

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.
Restriction: The Validate Username Token policy can be used only with the DataPower® Gateway.

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 1. 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.

Important: If you are using IBM® API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.

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 Username Token policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is **`validate-usernametoken`**.	string
Description	description	No	A description of the policy.	string
Authentication type	auth-type	Yes	The authentication type to use to validate the UsernameToken. Valid values: • **`Authentication URL`**: Select this value to validate the user credentials against an authentication URL. • **`LDAP registry`**: Select this value to validate the user credentials against an LDAP user registry. The default value is: **`Authentication URL`**.	string
Authentication URL	auth-url	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 **`Authentication URL`**.	string
TLS profile	tls-profile	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 **`Authentication URL`**.	string
LDAP registry name	ldap-registry	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 **`LDAP registry`**.	string
LDAP search attribute[1]	ldap-search-attribute	Yes	The name of the LDAP user password attribute. Note: This property is required only if Authentication type is set to **`LDAP registry`**.	string

Examples

The following example shows an LDAP user registry authentication:

```
- validate-usernametoken:
    title: "validate-usernametoken"
    auth-type: "LDAP Registry"
    ldap-registry: "wstest"
    ldap-search-attribute: "userPassword"
```

The following example shows an Authentication URL definition:

```
- validate-usernametoken:
    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

- Composing a REST API definition

Related information

- Creating an LDAP registry
- Authenticating by using your enterprise user registry

[1] 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

XML to JSON (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).

Restriction: The XML to JSON policy can be used only with the DataPower® Gateway.

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 Designer 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`.

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 1. Policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `xml-to-json`.	string
Description	No	A description of the policy.	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" } }
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

XSLT (xslt)

Use the XSLT policy to apply an XSLT transform to the payload of the API definition.

Restriction: The XSLT policy can be used only with the DataPower® Gateway.

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 1. XSLT 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 **xslt**.	string
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
Source	Yes	The XSLT transform source to execute.	string

For examples of the OpenAPI (Swagger 2.0) definitions of XSLT policies, see XSLT policy examples.

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

- Composing a REST API definition

Related reference

- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

XSLT policy examples

Examples of the OpenAPI (Swagger 2.0) 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:msl="http://www.ibm.com/xmlmap"
        exclude-result-prefixes="fn dp dp map xalan msl"
        version="1.0">
      <xsl:output method="xml" encoding="UTF-8" indent="no"/>

      <!-- root wrapper template  -->
      <xsl:template match="/">
        <msl:datamap>
          <xsl:choose>
            <xsl:when test="not(msl: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="msl: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>
        </msl: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 (Swagger 2.0) 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_5.0.0_com.ibm.apic.toolkit.doc_local:_isp_po
licy_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: []
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 Designer 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

if

Use the if construct to apply a section of the assembly when a condition is fulfilled.

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 (Swagger 2.0) 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 1. 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, **`if`** is used by default.
Description	No	A description of your construct, it is not displayed on the canvas.
Condition	Yes	Use JavaScript (if using the Micro Gateway) or GatewayScript (if using the DataPower® Gateway) to provide conditions. A list of context variables that you can use to generate conditions can be found in API Connect context variables.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

operation-switch

Use the operation-switch component to apply a section of the assembly to a specific operation.

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 (Swagger 2.0) 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 1. 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, `operation-switch` 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

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 JavaScript or 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.

To reference variables in the Condition field, 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 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 1. 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, `if` 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 JavaScript (if using the Micro Gateway) or GatewayScript (if using the DataPower® Gateway) to provide conditions. A list of context variables that you can use to generate conditions can be found in API Connect context variables.

Property	Required	Description
otherwise	No	Add an otherwise case if you want to execute a section of the assembly when no other cases are fulfilled.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

throw

Use the throw policy to throw an error when it is reached during the execution of an assembly flow.

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 1. 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, **`throw`** 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

User-defined policies

Make a user-defined policy available to a Catalog, and apply that policy to a REST or SOAP API.

A policy is a piece of configuration that controls a specific aspect of processing in the Gateway server during the handling of an API invocation at run time. IBM® API Connect provides built-in policies for many capabilities, including logging, redaction, proxy, and transformations, but you can also create user-defined policies to provide more processing control.

Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), support for implementing your own policies is *not* available.
The following information outlines how to create and manage user-defined policies:

Creating a user-defined policy

> A user-defined policy is created outside of IBM API Connect by using a standard development tool, but the policy must conform to a particular schema. For information about how to create a policy, see Authoring policies.
>
> DataPower Gateway only Note: If you are running your policies on a DataPower® Gateway, you must ensure that the DataPower processing to be controlled by the user-defined policy, is supported by the level of IBM DataPower Gateway that the policy will be run on. When using the GatewayScript actions, the minimum DataPower firmware that should be used is version 7.2.0.
> The last invoke

Importing a user-defined policy into IBM API Connect

> To make a user-defined policy available to an API, the policy must be imported into one or more Catalogs in the API Manager. If you also want to be able to apply the policy to your policy assembly in API Designer, you will have to import that policy into the developer toolkit environment.
>
> API Designer only Make your policy accessible to API Designer by using the developer toolkit. For detailed instructions, see Packaging and importing your policies into IBM API Connect®.

API Manager only Make your policy available by using the API Manager UI. For detailed instructions, see:

- DataPower Gateway only Importing a user-defined policy into a Catalog. If you want to create a new Catalog and import a user-defined policy as part of the configuration, follow the instructions in Creating and configuring Catalogs.
- Micro Gateway only Packaging and importing your policies into IBM API Connect.

Note: You must import a user-defined policy into every Catalog in API Manager that you require the policy to run in.

Applying a user-defined policy to an API

Apply a user-defined policy to an API definition by using the assemble view in API Designer or API Manager, to add the component to your assembly and configure its properties. For detailed instructions, see Including components in your assembly.

Deleting a user-defined policy

Delete a user-defined policy from a Catalog, by clicking the Delete icon next to the policy in the associated Catalog in API Manager.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Extensions commands

Use extensions commands to view and manage extensions.

You can extend the OpenAPI (Swagger 2.0) 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, then added to the API schema. For more information about defining extensions, see Adding an Open API (Swagger 2.0) extension to an API definition (API Manager UI).

Table 1.

Command name	Action	Syntax
extensions clone	The **extensions:clone** command makes a copy of the specified extensions. **Parameters** --catalog or -c <*catalog_name*> Specifies a single catalog with the catalog name. --organization or -o <*organization_name*> Specifies a single organization with the organization name. --server or -s <*management_server_endpoint*> Specifies the server endpoint.	`apic extensions:clone [{--catalog	-c} <catalog_name>][{--organization	-o}<organization_name>][{--server	-s}<management_server_endpoint>]` Example: `apic extensions:clone --catalog catalog1 --organization orgmain --server endpoint1` This example clones the extensions that are in *catalog1* of the *orgmain* organization, and are paired with *endpoint1*.
extensions delete	The **extensions:delete** command removes specified extensions from one or more catalogs. **Parameters** --catalog or -c <*catalog_name*> Specifies a single catalog with the catalog name. --organization or -o <*organization_name*> Specifies a single organization with the organization name. --server or -s <*management_server_endpoint*> Specifies the server endpoint.	`apic extensions:delete extension_name[:version_number_of_extension][{--catalog	-c}<catalog_name>][{--organization	-o}<organization_name>][{--server	-s}<management_server_endpoint>]` Example: `apic extensions:delete myextension:1.0.0 --catalog catalog1 --organization orgmain --server endpoint1` This example deletes *myextension* version *1.0.0* that is in *catalog1* of the *orgmain* organization, and is paired with *endpoint1*.

Command name	Action	Syntax

extensions get	The **extensions:get** command displays information about extensions in a catalog.
	Parameters
	--catalog or -c <*catalog_name*>
	Specifies a single catalog with the catalog name.
	--organization or -o <*organization_name*>
	Specifies a single organization with the organization name.
	--server or -s <*management_server_endpoint*>
	Specifies the server endpoint.
		`apic extensions:get extension_name[:version_number_of_extension][{--catalog | -c} <catalog_name>][{--organization | -o} <organization_name>][{--server | -s} <management_server_endpoint>]`
		Example:
		`apic extensions:get myextension:1.0.0 --catalog catalog1 --organization orgmain --server endpoint1`
		This example gets *myextension* version *1.0.0* that is in *catalog1* of the *orgmain* organization, and is paired with *endpoint1*.

extensions list	The **extensions:list** command lists the extensions that are available. This is the default command if you enter only `apic extensions`.
	Parameters
	--catalog or -c <*catalog_name*>
	Specifies a single catalog with the catalog name.
	--all-catalogs or -C
	Specifies all catalogs.
	--organization or -o <*organization_name*>
	Specifies a single organization with the organization name.
	--all-organizations or -O
	Specifies all organizations.
	--server or -s <*management_server_endpoint*>
	Specifies the server endpoint.
		`apic extensions:list [{--catalog | -c} <catalog_name>|{--all-catalogs | -C}][{--organization | -o} <organization_name> | {--all-organizations | -O}][{--server | -s} <management_server_endpoint>]`
		Example:
		`apic extensions:list --catalog catalog1 --all-organizations --server endpoint1`
		This example lists the extensions that are paired with *endpoint1*, are in *catalog1*, and in any organization.

extensions publish	The **extensions:publish** command publishes information about extensions in a catalog.
	Parameters
	--catalog or -c <*catalog_name*>
	Specifies a single catalog with the catalog name.
	--organization or -o <*organization_name*>
	Specifies a single organization with the organization name.
	--server or -s <*management_server_endpoint*>
	Specifies the server endpoint.
		`apic extensions:publish extension_name[:version_number_of_extension][{--catalog | -c} <catalog_name>][{--organization | -o} <organization_name>][{--server | -s} <management_server_endpoint>]`
		Example:
		`apic extensions:publish myextension:1.0.0 --catalog catalog1 --organization orgmain --server endpoint1`
		This example publishes *myextension* version *1.0.0* that is in *catalog1* of the *orgmain* organization, and is paired with *endpoint1*.

extensions pull	The **extensions:pull** command pulls information about extensions in a catalog.
	Parameters
	--catalog or -c <*catalog_name*>
	Specifies a single catalog with the catalog name.
	--organization or -o <*organization_name*>
	Specifies a single organization with the organization name.
	--server or -s <*management_server_endpoint*>
	Specifies the server endpoint.
		`apic extensions:pull extension_name[:version_number_of_extension][{--catalog | -c} <catalog_name>][{--organization | -o} <organization_name>][{--server | -s} <management_server_endpoint>]`
		Example:
		`apic extensions:pull myextension:1.0.0 --catalog catalog1 --organization orgmain --server endpoint1`
		This example pulls *myextension* version *1.0.0* that is in *catalog1* of the *orgmain* organization, and is paired with *endpoint1*.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tags in API Connect

There are three forms of tagging in API Connect.

- You can create tags in an API that are visible in its OpenAPI (Swagger 2.0) 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.
- You can mark items as favorites in API Manager or the API Designer to make them visible in the Favorites section of the navigation pane.

Tags within an API

The OpenAPI (Swagger 2.0) 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 in this way can be used to filter the operations of an API in the Developer Portal. These tags are visible to anyone who can view any of the Products to which the API belongs.

For more information, see Composing a REST API definition.

Note: Though you can tag the API itself, instead of an operation, 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 search function in the Developer Portal.

For more information, see Editing tags for a specific item.

Categorizing APIs, Catalogs, and Products as favorites

In API Manager or the API Designer, you can mark APIs, Catalogs, and Products as favorites, causing them to appear in the Favourites expandable section of the navigation pane. Favorites marked in this way are only visible in the user interface where they were marked, and never in the Developer Portal

You can mark an object as a favorite by clicking the Add to favorites icon The Add to favorites icon for the object. You can remove an object from your list of favorites by clicking the Remove from favorites icon The Remove from favorites icon.

Related concepts

- Working with Products in the API Designer

Related tasks

- Creating API definitions

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring API security by using the API Designer

You configure security for an API by creating one or more security definitions 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.
- V5.0.8 + **Securing your APIs with OpenID Connect**
  OpenID Connect (OIDC) is built on top of the OAuth 2.0 protocol and focuses on identity assertion. OIDC provides a flexible framework for identity providers to validate and assert user identities for Single Sign-On (SSO) to web, mobile, and API workloads.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

API Connect is secured to forbid external references in XML managers. For example, if you configure a custom login form or consent form for OAuth provider, it cannot refer to an external DTD.

You can create security definitions of the following types:

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 an LDAP user registry or an authentication URL 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.
- **Protecting an API with OAuth**
  In API Connect, you can secure API with 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.
- **Scope**
  Per IETF RFC 6749, the value of the scope parameter is expressed as a list of space-delimited, case-sensitive strings.
- **Tokens**
  If you are using OAuth authentication, you can enable refresh tokens.

- **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a basic authentication security definition

When you create a basic authentication security definition in an API, you provide details of an LDAP user registry or an authentication URL to be used to authenticate access to the API operations.

About this task

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.

Note: The API Manager UI also includes the ability to create and edit security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.
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.

Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.

Procedure

To create a basic authentication security definition, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. 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 API definitions.
5. Navigate to the Security Definitions section.
6. In the Security Definitions section, click the Add Security Definition icon ⊕.
7. Select Basic.
   A Basic security definition is added to the Security Definitions section.
8. Enter a name for the security definition, to replace the default name, and, optionally, a description.
9. To authenticate users with an authentication URL, complete the following steps:
   a. Select Authentication URL, and specify a URL.
      When establishing authentication, API Connect makes a GET call to your authentication URL. When the call is made, it includes in its authorization header the user name and password it has collected from the user. Confirm that these are correct and respond with an HTTP success code such as `200 OK` if you want to allow the application access, or with an HTTP error code such as `401 Unauthorized` if you want to deny access.
   b. API Designer only To apply a TLS profile, click TLS Profile and enter the name of the required TLS profile.
   c. API Manager only To apply a TLS profile, click TLS Profile and select the required TLS profile.

Note: The TLS profile must be created on the Management server by using the API Manager user interface. For more information, see TLS profiles.

10. To authenticate users with an LDAP user registry, complete the following steps:
    a. Select User registry.
    b. API Designer only Enter the name of the required registry.
    c. API Manager only Select the required registry.

    Note: The LDAP user registry must be created on the Management server by using the API Manager user interface. For more information, see Creating an LDAP registry.
11. Click the Save icon 💾 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.

- V5.0.5 + **Authentication URL**
  You can use an Authentication URL to specify a REST authentication service that manages user authentication, and optionally provide additional meta data to be embedded in the token.
- **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Authentication URL

You can use an Authentication URL to specify a REST authentication service that manages user authentication, and optionally provide additional meta data to be embedded in the token.

This support can optionally enable any of the following:

- Provide the authenticated credential to API Connect. For example, the user logs-in with user name: `spoon`, and password: `passw0rd`. 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.
- Provide metadata support. Allow extra metadata to be kept in the **`access_token`**.
- V5.0.7 + Override 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, 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
API-Authenticated-Credential: cn=spoon,o=eatery
```

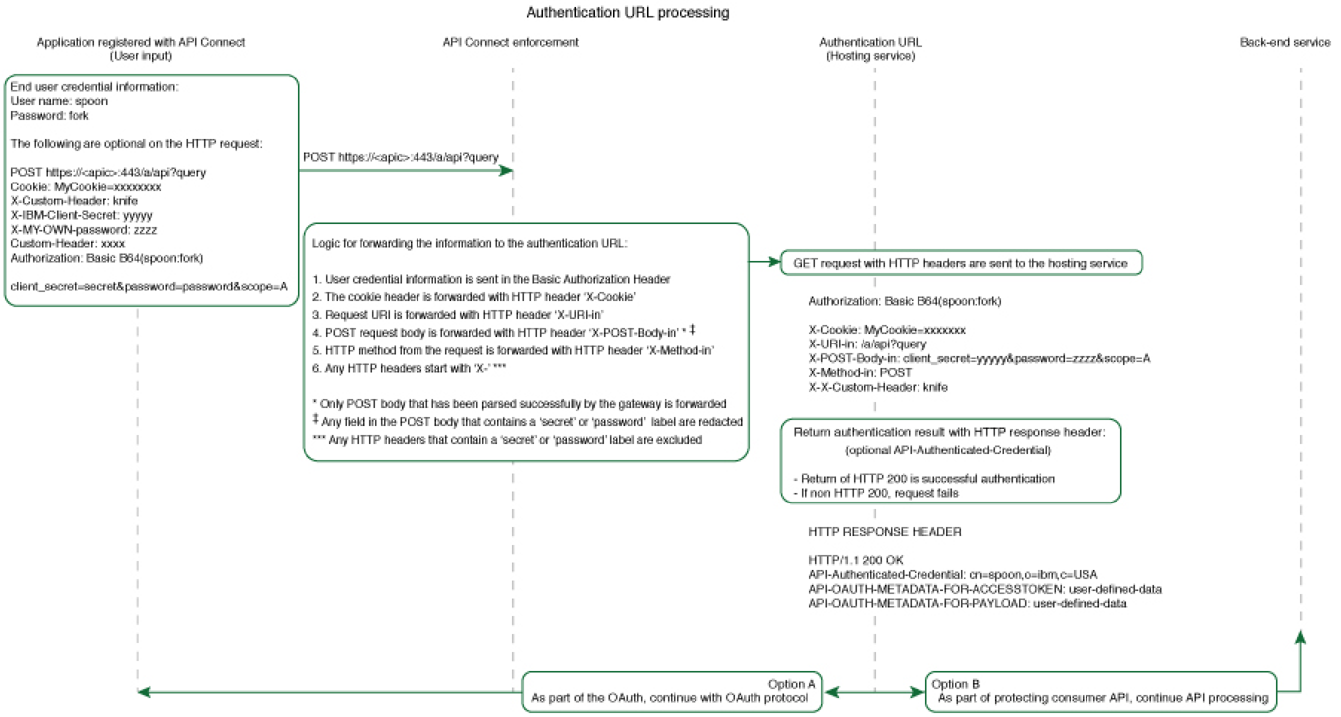
API Connect considers any non-200 HTTP response code a failed user authentication attempt.

V5.0.6 + When Authentication URL 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 V5.0.6 + OAuth metadata URL and authentication URL. The two metadata response headers are:

```
API-OAUTH-METADATA-FOR-ACCESSTOKEN
API-OAUTH-METADATA-FOR-PAYLOAD
```

V5.0.7+ From version 5.0.7.3 onward, when Authentication URL is invoked, an HTTP response header is available to override the requested `scope` from the application. For more information, see OAuth scope. The response header is:

```
x-selected-scope
```



Authentication URL processing
Application registered with API Connect (User input)
API Connect enforcement
Authentication URL (Hosting service)
Back-end service
POST https://<apic>:443/a/api?query
Option A
Option B

End user credential information:
User name: spoon
Password: fork

The following are optional on the HTTP request:

```
POST https://<apic>:443/a/api?query
Cookie: MyCookie=xxxxxxxx
X-Custom-Header: knife
X-IBM-Client-Secret: yyyyy
X-MY-OWN-password: zzzz
Custom-Header: xxxx
Authorization: Basic B64(spoon:fork)

client_secret=secret&password=password&scope=A
```

Logic for forwarding the information to the authentication URL:

1. User credential information is sent in the Basic Authorization Header
2. The cookie header is forwarded with HTTP header 'X-Cookie'
3. Request URI is forwarded with HTTP header 'X-URI-in'
4. POST request body is forwarded with HTTP header 'X-POST-Body-in' * ‡
5. HTTP method from the request is forwarded with HTTP header 'X-Method-in'
6. Any HTTP headers start with 'X-' ***

* Only POST body that has been parsed successfully by the gateway is forwarded
‡ Any field in the POST body that contains a 'secret' or 'password' label are redacted
*** Any HTTP headers that contain a 'secret' or 'password' label are excluded

GET request with HTTP headers are sent to the hosting service

```
Authorization: Basic B64(spoon:fork)

X-Cookie: MyCookie=xxxxxxx
X-URI-in: /a/api?query
X-POST-Body-in: client_secret=yyyyy&password=zzzz&scope=A
X-Method-in: POST
X-X-Custom-Header: knife
```

Return authentication result with HTTP response header:
(optional API-Authenticated-Credential)

- Return of HTTP 200 is successful authentication
- If non HTTP 200, request fails

```
HTTP RESPONSE HEADER

HTTP/1.1 200 OK
API-Authenticated-Credential: cn=spoon,o=ibm,c=USA
API-OAUTH-METADATA-FOR-ACCESSTOKEN: user-defined-data
API-OAUTH-METADATA-FOR-PAYLOAD: user-defined-data
```

Option A
As part of the OAuth, continue with OAuth protocol

Option B
As part of protecting consumer API, continue API processing

Related concepts

- OAuth metadata URL and authentication URL

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 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_5.0.0_com.ibm.apic.toolkit.doc_con_ldap_requi
rements_, 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 LDAP referral.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 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.

Note: The API Manager UI also includes the ability to create and edit security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

Procedure

To create an API key definition, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. 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 API definitions.
5. Navigate to the Security Definitions section.
6. In the Security Definitions section, click the Add Security Definition icon ⊕.
7. Select API Key.
8. Enter a name for the security definition, to replace the default name, and, optionally, a description.
9. Enter the Parameter name.
   If your API is enforced by the IBM® API Connect gateway, enter one of the following values depending on where the client credentials are to be located, and the type of credentials that are required:

   Table 1. Client ID and Client secret parameter name values

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

   If your API is not enforced by the IBM API Connect gateway, enter the parameter name required by your gateway.

   When you change the location of an API key security definition's credentials, the parameter name changes appropriately.

   When you first create an API, default API key security definitions are 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.
10. 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.
11. Click the Save icon 💾 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Protecting an API with OAuth

In API Connect, you can secure API with 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.

- **OAuth user scenario**
  Potential users of OAuth with API Connect have a number of methods to secure their API. The following scenario provides an overview of the available options.
- **Creating an OAuth provider API**
  An OAuth provider API contains the authorization and token endpoints of an OAuth flow.
- **Protecting an API with OAuth security definition**
  When you create an OAuth security definition in an API, you provide settings for controlling access to the API operations through the OAuth authorization standard.
- V5.0.6 + **OAuth metadata URL and authentication URL**
  You can use the Metadata URL or Authentication 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.
- **OAuth responses**
  API Connect returns various responses to requests during the OAuth 2.0 process.
- V5.0.6 + **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

OAuth user scenario

Potential users of OAuth with API Connect have a number of methods to secure their API. The following scenario provides an overview of the available options.

Micro Gateway only V5.0.3+ OAuth 2.0 is only supported by the Micro Gateway from version 5.0.3 and onwards.

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 IBM® 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating an OAuth provider API

An OAuth provider API contains the authorization and token endpoints of an OAuth flow.

Before you begin

To use your DataPower® Gateway to manage the revocation and introspection of access tokens, you must be running IBM® DataPower version 7.5.1.1 or later.

About this task

An OAuth provider API can serve multiple APIs that are employing OAuth security definitions. It provides operations that are the authorization and token endpoints of an OAuth flow.

In API Connect, scopes are defined in the provider API and listed as requirements by the secured API. All scopes that are listed by the security definition of the secured API must be granted by the access token. You have the opportunity to override the value of your scope during each phase of the OAuth process. For more information, see Scope for API Connect.

Each token grants access to a specific site for specific resources for a defined duration. By using an OAuth token, a user can grant a third-party site access to a range of information, which is stored with another service provider, without needing to share their personal credentials. For more information, see Tokens.

A scope cannot be restricted to a single user, and instead, you should configure your secured API to behave differently by referencing the `oauth.resource-owner` context. For more information, see IBM API Connect context variables.

If the user changes their mind and decides that they do not want a third-party site to continue to have access to their information, the user can revoke the token access. If the token revocation URL is specified, the token revocation list is always checked before access is granted to the user information. For more information, see OAuth revocation URL.

Your OAuth provider API can support multiple OAuth flows, each of which corresponds to an OAuth grant type. Security definitions, which are applied to other APIs to use OAuth, use only one of the flows. The different types of OAuth flow, and the corresponding OAuth grant types, are shown in the following table:

Table 1. OAuth security definition types

OAuth flow	Corresponding OAuth grant type
Implicit	Implicit
Password	Resource Owner Password Credentials
Application	Client Credentials
Access Code	Authorization Code

You can secure your APIs with a third-party OAuth provider. For more information, see Integrating third party OAuth provider.

Note:
The OAuth provider API logs failure cases in Analytics data, 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 API.

Procedure

To create an OAuth provider API, complete the following steps:

Note: The API Manager and API Designer user interfaces both include the ability to create and edit APIs. However, the preferred method for these tasks is by using the API Designer user interface, as described in the following steps. Any tasks that are specific to a particular user interface are marked with an icon.

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. Click Add and then click New OAuth 2.0 Provider API.
5. Complete the fields that are presented.
   - The title can include special characters but should be kept short so that it can be easily displayed in the user interface.
   - The name should be kept short and can contain only lowercase alphanumeric characters (a-z and 0-9), underscore characters (_), or hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
   - V5.0.4 + 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 version corresponds to the value of the `info.version` property of the API's OpenAPI (Swagger 2.0) definition. The `version.release.modification` version numbering scheme is recommended; for example `1.0.0`.
6. Specify whether your provider API is included in a Product, and create the API.
   - To create a new Product and include your provider API in that Product, complete the following steps:
     - Click Add a product.
     - API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file that is provided with the developer toolkit, or another template file that you configure as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates.
     - Accept the default values for the Product title, name, and version, or change them.
     - To publish the Product to a target Catalog, ensure that the Publish this product to a catalog check box is selected and then select the Catalog. You can clear this check box and stage or publish the Product later by using the API Designer UI and API Manager UI, as described in Staging a Product and Publishing a Product.
     - If Spaces have been enabled, select the Catalog and Space that you require.
     - Click Create API.

- To create your provider API without adding it to a Product, click Create API.

The Design tab for the draft of your provider API definition opens. You can skip to different sections of your API definition by using the page navigation in the side bar. You can view the OpenAPI (Swagger 2.0) definition of your API in the Source tab and after you create an assembly, view your policy assembly in the Assemble tab.

7. In the Design tab, edit the Info section.
   a. Optional: Edit any or all of Title, Name, Version, and Description.
   b. Optional: In the Contact section, provide details for any or all of Name, Email, and URL.
   c. In the Terms and License section, provide details for any or all of Terms of Service, License Name, and License URL.
   d. In the External Documentation section, provide a Description for any external documentation you want to refer users to and a URL for where the documentation can be accessed.
8. In the Schemes section, select which transfer protocols you want your provider API to use.
   Note: If your provider API is enforced by the API Connect gateway, only the HTTPS protocol is supported. See Step 9 for instructions of how to enable enforcement.
9. If your provider API is to be enforced by a gateway other than API Connect, use the Host field in the Host section to define the gateway URL that is to be used.
10. Configure the OAuth 2 section.
    a. Use the Client type drop-down menu to specify whether the OAuth provider API uses a Public or Confidential flow.
    b. Required: Define scopes by providing a Scope Name and an optional Description.

       At least one scope is required. You can create a scope by clicking the Add scope icon ⊕ and you can delete a scope by clicking the Remove scope icon 🗑. A scope that is defined becomes an option in the request for an access token from the provider API. In the security definition of a secured API, describe the scopes for which a token must be valid to grant access to the secured API. When an access token is requested from the provider API, multiple scopes must be separated by spaces.

       In the Advanced Scope Check section, use the Enable Application Scope Check slider to enable or disable the option to provide additional scope verification, resulting in what the scope application is allowed to have. This process happens after API Connect successfully verifies the application credential, and before API Connect attempts to authenticate the resource user. For more information, see OAuth Scope.

       In the Advanced Scope Check section, use the Enable Owner Scope Check slider to enable or disable the option to provide additional scope verification, resulting in what scope the authenticated user is allowed to have. This process happens after user has been authenticated successfully during the OAuth protocol exchange. For more information, see OAuth Scope.

    c. In the Grants section, select which OAuth 2.0 flows you want to use.
       Note: If you are using a Public flow, you cannot have an Application grant type; and if you are using a Confidential flow, you cannot have an Implicit grant type. You must have at least one grant type to use the provider API.
    d. In the Identity extraction section, use the Collect credentials using drop-down menu to specify how a user's credentials are collected during the authorization process.
       - Select Default form where authentication is assumed.
       - Select Basic to use basic authentication.
       - Select Custom form to use a custom HTML form that you provide to API Connect. If you select this option, provide a URL at which your custom form can be found in the Custom form field. For more information, see Creating a custom sign-in form.
       - Select Redirect to use an externally hosted service for authentication. If you select this option, provide the URL at which users start your authentication process in the Redirect URL field. For more information, see Authenticating and authorizing through a redirect URL.
    e. In the Authentication section, use the Authenticate application users using drop-down menu to select how a user is authenticated.
       - To authenticate users with an LDAP registry, select User registry, then complete the following action according to which user interface you are using:
         - API Manager only Select the registry from the User registry list.
         - API Designer only Enter the name of the registry.

         Note: The LDAP user registry must be created on the Management server by using the API Manager user interface. For more information, see Creating an LDAP registry.
       - Select Authentication URL to authenticate users by sending credentials that are collected by API Connect to this URL. When establishing authentication, API Connect makes a GET call to your authentication URL. When the call is made, it includes in its authorization header the user name and password it has collected from the user. Confirm that these are correct and respond with an HTTP success code such as `200 OK` if you want to allow the application access, or with an HTTP error code such as `401 Unauthorized` if you want to deny access.
         Note: If you are authenticating and authorizing users through a redirect URL, you must supply an authentication URL. For more information, see Authentication URL.
         To apply a TLS profile for communication with the authentication URL, complete one of the following actions according to the user interface you are using:
         - API Manager only Select the TLS profile from the TLS Profile list.

  - API Designer only Enter the name of the TLS profile.

    Note: The TLS profile must be created on the Management server by using the API Manager user interface. For more information, see TLS profiles.

    To change the value of the resource owner to differ from the value sent to the authentication URL, the authentication URL should return a header that is named `API-Authenticated-Credential`, with its value set to the new resource owner. For example:

    ```
    --header "API-Authenticated-Credential: Resource_Owner"
    ```

    where *Resource_Owner* is the value to which you want to set the **`oauth.resource-owner`** context.

f. In the Authorization section, use the Authorize application users using drop-down menu to select how authorization should be granted.
   - Select Default form to use the default form that is provided by API Connect.
   - Select Custom form to use your own custom HTML form. If you select this option, provide a URL at which your form can be found in the Custom form field. You can also select a TLS profile from the drop-down menu to use for communications with this URL. For more information, see Creating a custom authorization form.
   - Select Authenticated to automatically grant authorization.
     Note: If you are authenticating and authorizing users through a redirect URL, you must automatically grant authorization.

g. In the Tokens section, use the Time to live (seconds) field to specify for how many seconds an access token remains valid, for a minimum of `1` and a maximum of `63244800`.

h. In the Tokens section, use the Enable refresh tokens slider to enable or disable the use of refresh tokens by applications.
   If you enable refresh tokens, the following fields are available for you to use:

   Count

   Use the Count field to specify how many times a refresh token can be requested. You can request a refresh token up to `4096` times per *permission set*. (A permission set is an instance of application, owner, and permission)

   Time to live (seconds)

   Use the Time to live field to specify how many seconds a refresh token remains valid. The time range available is `2` to `252979200` seconds.

i. In the Tokens section, use the Time to live (seconds) field under Maximum Consent, to specify for how many seconds the combination of any number of access and refresh token remain valid. The time range available is `0` to `2529792000` seconds. Note, this feature is only present if refresh tokens are enabled. Setting the value to `0` disables this feature.

j. In the Tokens section, use the Enable revocation slider to enable or disable the use of a list of blocked applications.
   - If you want to use your DataPower Gateway to manage revocation of access tokens, select Use DataPower Gateway. Use the Enable users to view and revoke permissions switch to specify whether more operations are made available to applications that can use the provider API so that they can view and revoke access tokens.
     The list of revoked applications is shared between all provider APIs. If you do not enable the additional operations, then to revoke an application you need to use a second provider API that has them enabled.

     Important:
     If you select the Allow users to view and revoke permissions option, and the Client type is set to Public, you must manually edit the OpenAPI source for your OAuth Provider API to remove the client secret header setting; complete the following steps:

     - Select the Source tab.
     - Locate the following section:

       ```
       /oauth2/issued:
           get:
             .
             .
             .
             security:
               - clientIdHeader: []
                 clientSecretHeader: []
       ```

     - Replace

       ```
             security:
               - clientIdHeader: []
                 clientSecretHeader: []
       ```

       with

       ```
             security: []
       ```

     - Locate the following section:

```
/oauth2/issued:
    delete:
      .
      .
      .
      security:
        - clientIdHeader: []
          clientSecretHeader: []
```

- Replace

  ```
        security:
          - clientIdHeader: []
            clientSecretHeader: []
  ```

  with

  ```
        security: []
  ```

- Save your OAuth provider API.

In addition to revoking all tokens for an application, you can revoke a single token; use the Allow application to revoke its token switch to enable this capability. To revoke a single token for an application, send one or other of the following requests to the DataPower Gateway (the **curl** command is used by way of illustration):

```
curl -X POST \
  https://Datapower_Gateway_Hostname/oauth2/revoke \
  -H ‘authorization: Basic base64_encoded_application_credential’ \
  -H ‘content-type: application/x-www-form-urlencoded’ \
  -d “token_type_hint=access_token&token=Access_Token”
```

```
curl -X POST \
  https://Datapower_Gateway_Hostname/oauth2/revoke \
  -H ‘authorization: Basic base64_encoded_application_credential’ \
  -H ‘content-type: application/x-www-form-urlencoded’ \
  -d “token_type_hint=refresh_token&token=Refresh_Token”
```

Replace *Datapower_Gateway_Hostname*, *base64_encoded_application_credential*, *Access_Token*, and *Refresh_Token* with the appropriate values.
For further information, see Managing tokens with the DataPower Gateway.

Important:

- If you select the Allow application to revoke its token option, and the Client type is set to Public, you must manually edit the OpenAPI source for your OAuth Provider API to remove the client secret header setting; complete the following steps:

  - Select the Source tab.
  - Locate the following section:

    ```
     /oauth2/revoke:
       post:
         .
         .
         .
         security:
           - clientSecretHeader: []
             clientIdHeader: []
    ```

  - Replace

    ```
          security:
            - clientIdHeader: []
              clientSecretHeader: []
    ```

    with

    ```
          security: []
    ```

  - Save your OAuth provider API.
- 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.

To configure DataPower to synchronize OAuth data, complete the following steps:

- Ensure that the DataPower quota enforcement server is configured. For more information, see Configuring the quota enforcement server.
- Enable revocation in the OAuth provider.

Restriction: The Enable revocation option is not supported if you are using API Connect with IBM Cloud public.

- If you want to use a revocation URL, select Revocation URL and provide the URL in the Revocation URL field. You can also select a TLS profile from the drop-down menu to use for communications with this URL. For more information, see OAuth revocation URL.

k. Use the Enable token introspection slider to enable or disable token introspection.
Enabling introspection creates a new operation that can be called, which returns all information about an access token that is passed to it, such as its scope and validity. For more information, see Integrating third party OAuth provider.
Important:
If you select the Enable token introspection option, and the Client type is set to Public, you must manually edit the OpenAPI source for your OAuth Provider API to remove the client secret header setting; complete the following steps:

i. Select the Source tab.
ii. Locate the following section:

```
  /oauth2/introspect:
    post:
      .
      .
      .
      security:
        - clientIdHeader: []
          clientSecretHeader: []
```

iii. Replace

```
      security:
        - clientIdHeader: []
          clientSecretHeader: []
```

with

```
      security: []
```

iv. Save your OAuth provider API.

l. Specify the metadata URL where request headers are sent to retrieve extra content for the OAuth transaction. For more information, see OAuth metadata URL and authentication URL.

11. Optional: In the Consumes section, select which types of media your provider API accepts when calls are made to it. Add other supported media types in addition to JSON and XML, by using the Add Media Type field.
Important: The provider API must accept `application/x-www-form-urlencoded` content and changes to this field do not affect the behavior of the OAuth flow, only the documentation available through the Developer Portal.
12. Optional: In the Produces section, select which types of media your provider API returns when calls are made to it. Add other supported media types in addition to JSON and XML, by using the Add Media Type field.
Important: The provider API always produces `application/json` content and changes made to this field do not affect the behavior of the OAuth flow, only the documentation is available through the Developer Portal.
13. Optional: Configure the Lifecycle section.
    a. Optional: For Phase, use the drop-down menu to change the phase of the lifecycle that your provider API is in.
    The options are as follows.

    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.

    Realized
    The API is in the implementation phase.

    b. Optional: Set the Testable toggle to the On position to allow the provider APIs operations to be tested using the test tool in the Developer Portal.
    Note: For the test tool to work, your provider API must be included in a Plan in a Product that is staged in a development Catalog.
    c. Optional: Set the Enforced toggle to the On position to enforce your provider API by using the API Connect Gateway.
    d. Optional: Set the CORS toggle to the On position to enable CORS access control.
14. Optional: If you want to perform transformations or other actions when the provider API is called, create an assembly by clicking Create assembly in the Policy Assembly section.

For more information, see The assemble view.

15. Optional: In the Security Definitions section, manage any security definitions that might be used by the API or its operations. For more information, see Configuring API security
16. Optional: In the Security section, select any security definitions that you want to apply to your provider API. To be available in the Security section, definitions must have been defined in the Security Definitions section.
17. API Manager only In the Extensions section, add any vendor extensions you want to use with your API.
18. Optional: In the Properties section, define any API properties that you want to use. For more information, see API properties.
19. Optional: Add paths to your API. For more information, see Defining Paths for a REST API.
    Important: Deleting the existing Paths in your provider API will interrupt the correct functioning of the OAuth flow.
20. Optional: In the Parameters section, add parameters that are shared by all Paths and operations in the API.

    a. Click the Add Parameter icon ⊕.
    b. In the Name field, provide a name for your parameter.
    c. In the Located In field, select where the parameter is found in the call of your operation.
    d. Optional: In the Description field, provide a description of your parameter.
    e. Use the Required check box to specify whether the parameter is required for a call to be valid.
    f. Optional: From the drop-down list for 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. For more information about creating JSON schemas, see step 21.
21. Optional: In the Definitions section, you can create JSON schema definitions.
    You reference these definitions in an operation to provide developers with information about the JSON request they should make or the JSON 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 (validate) policy is used.
    Important: Deleting the existing definitions in your provider API interrupts the correct functioning of the OAuth flow.

    a. Click the Add Definition icon ⊕.
       A new definition is created.
    b. Click the newly created definition to expand its details.
    c. Complete the Name field for your definition.
    d. From the Type drop-down list, select the type of your definition.
    e. Optional: In the Description field for your definition, provide a description of what is defined by the definition.
    f. Complete the details of your definition's properties. Each property requires a name and type and can also have a description.
    g. Optional: To specify that a property is required when the operation is called, select it using the check box in the Required The Required icon column.
    h. Optional: You can add additional properties by clicking Add Property.
    i. Optional: You can delete properties or definitions by clicking the Delete icon 🗑 next to the property or definition.
    j. 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, set Allow additional properties to the On position.

    Note:
    - You can include more complex schema definitions in your API by using the Source tab and editing your API's OpenAPI (Swagger 2.0) definition directly. For more information, see the OpenAPI (Swagger 2.0) specification.
22. Optional: To add any tags, in the Tags section click the Add Tag icon ⊕. Tags added in this way appear in the OpenAPI (Swagger 2.0) definition of the provider API but are not used by API Connect for any indexing.
23. Click the Save icon 💾 to save your changes.

Results

You created and configured an OAuth 2.0 provider API.

- V5.0.7+ **Integrating third party OAuth provider**
  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. API Connect can use this feature along with the mentioned provider to protect access to the API.
- **OAuth custom forms and protection against XSS attacks**
  When the OAuth security definition uses Implicit Flow, Password Flow, or Authorization flow, you can present HTML forms to users during the extract identity and authorization stages. Because HTML forms can include external and inline sources, , such as images or JavaScript, these sources can be the origin of cross-site scripting (XSS) attacks.

Related reference

- API and Product definition template examples

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

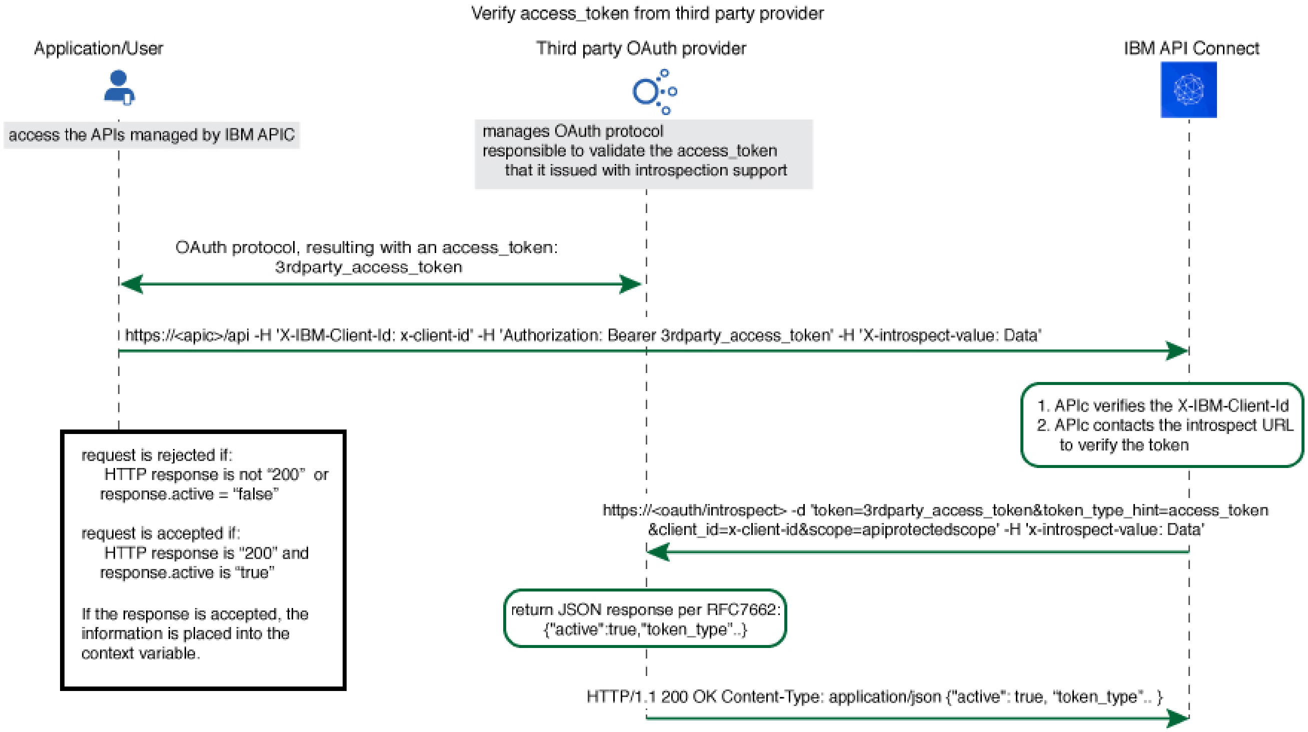
Integrating third party OAuth provider

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®. API Connect can use this feature along with the mentioned provider to protect access to the API.

You can use IBM API Connect to protect an API that is secured by using the third-party OAuth access token in accordance with the 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.

The **`x-introspect-basic-authorization-header`** is available to provide a user configured HTTP Basic authorization header.

The sequence diagram depicts the overall flow of request and response.


Verify access_token from third party provider
Application/User
Third party OAuth provider
IBM API Connect
access the APIs managed by IBM APIC
manages OAuth protocol
responsible to validate the access_token
that it issued with introspection support
OAuth protocol, resulting with an access_token:
3rdparty_access_token
https://<apic>/api -H 'X-IBM-Client-Id: x-client-id' -H 'Authorization: Bearer 3rdparty_access_token' -H 'X-introspect-value: Data'
1. APIc verifies the X-IBM-Client-Id
2. APIc contacts the introspect URL
to verify the token
request is rejected if:
HTTP response is not "200" or
response.active = "false"
request is accepted if:
HTTP response is "200" and
response.active is "true"
If the response is accepted, the
information is placed into the
context variable.
https://<oauth/introspect> -d 'token=3rdparty_access_token&token_type_hint=access_token
&client_id=x-client-id&scope=apiprotectedscope' -H 'x-introspect-value: Data'
return JSON response per RFC7662:
{"active":true,"token_type"..}
HTTP/1.1 200 OK Content-Type: application/json {"active": true, "token_type".. }

The Introspection URL can be configured in the OAuth security definition of the API. Entering a valid URL and optional TLS profile creates the following section in the API definition, conforming to the Open API (Swagger 2.0) specification:

```
x-tokenIntrospect:
 url: your_introspection_URL
 tls-profile: tls_profile_to_use
```

Note: Use **`x-tokenIntrospect`** **only** when connecting to a third party OAuth provider. Using **`x-tokenIntrospect`** with a native OAuth provider increases the time taken to complete the security validation, and is unnecessary provided that the OAuth provider is not using an OAuth token that was generated by a different OAuth provider.
When an API is protected by this feature, API Connect extracts the bearer token and issues an HTTP POST request to the introspection endpoint specified by **`x-tokenIntrospect`**.

The GET request is protected by APIC with this feature.

A header prefix of **`x-Introspect-`** can be used to pass information to the third-party provider as in the example.

```
     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
```

API Connect issues this POST request to the introspection endpoint that you specified in **`x-tokenIntrospect`**:

```
    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
```

When the request is successful, the third-party OAuth/OIDC provider responds with `HTTP 200`, and the token payload information. API Connect accepts 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 not valid.

```
    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 you contact the introspection endpoint, API Connect uses **`client_id`**/**`appId`** and **`client_secret`**/**`appSecret`** to construct the HTTP Basic authorization header. If only **`client_id`** is provided, it will be sent to the third-party provider in the form body.

If **`x-introspect-basic-authorization-header`** exists, its value is used for the HTTP Basic authorization header when the introspection endpoint is contacted. API Connect verifies that the HTTP Basic authorization header value is Base64 encoded before it is sent. If the header is already encoded it is sent unmodified, otherwise the header value is encoded as shown in the following 2 examples.

```
HTTP/1.1
x-introspect-basic-authorization-header: user:password
GET /petstore/pet/123 HTTP/1.1 Host: apiconnect.com x-introspect-basic-authorization-header: 3rd-party-
client_id:3rd-party_client_secret
```

API Connect sends the following.

```
POST ..
Authorization: Basic
M3JkLXBhcnR5LWNsaWVudF9pZDozcmQtcGFydHlfY2xpZW50X3NlY3JldA==token_type_hint=access_token&token= ..
```

Note: If you need to contact the introspection end point with a credential that cannot be specified with any of the methods previously described, a proxy policy can be used to override the Basic Authorization Credential to the proper value. When you use a proxy policy you can export and share the OAuth provider swagger while maintaining the security of sensitive credential information.
V5.0.7+ 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.
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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

OAuth custom forms and protection against XSS attacks

When the OAuth security definition uses Implicit Flow, Password Flow, or Authorization flow, you can present HTML forms to users during the extract identity and authorization stages. Because HTML forms can include external and inline sources, , such as images or JavaScript, these sources can be the origin of cross-site scripting (XSS) attacks.

The sign-in and authorization form are presented to users during different OAuth processing stages based on the configuration of the OAuth 2.0 provider API.

- The sign-in form is used during the extract identity stage. During OAuth processing, this form is presented so that users can sign in to the service provided by the API. Out-of-box processing does not prevent the direct use of external or inline sources. To protect against XSS attacks, you can add an XSLT Action to the assembly of the OAuth 2.0 processing API to set the value of the `Content-Security-Policy` response header. Use the information below to set a suitable header value for your environment.
- The authorization form is used during the authorization stage. During OAuth processing, this form is presented so that users can grant permission to an application that accesses their data through the API on their behalf. Out-of-box processing prevents the direct use of external or inline sources. To allow direct use of external or inline sources, add an XSLT Action to the assembly for the OAuth 2.0 processing API. This XSLT changes the out-of-box behavior by changing the value of the `Content-Security-Policy` response header. Use the information below to set a suitable header value for your environment.

If the assembly for the OAuth 2.0 provider API includes an XSLT Action, the value of the `Content-Security-Policy` response header applies to any defined custom form.

Independent of whether you use the default or custom forms, review and update as appropriate the XSS response headers for processing assembly.

`Content-Security-Policy`

The HTTP response header controls which resources the user agent can load for a specific page to protect against XSS attacks.

- The processing of the sign-in form can return this header. This header is returned only when the XSLT action in the assembly sets this header. In other words, out-of-box processing does not use this header.
- The processing of the authorization form returns this header independent of the XSLT action in the assembly. Out-of-box processing uses a setting of `Content-Security-Policy: default-src 'self'; style-src 'unsafe-inline'`.

For more information, see Content-Security-Policy.

`X-XSS-Protection`

The HTTP response header that stops the loading of pages when an XSS attack is detected.

- The processing of the sign-in form returns this header.
- The processing of the authorization form returns this header.

The default setting is `X-XSS-Protection: 1; mode=block`.

For more information, see X-XSS-Protection.

To use external and inline sources with the authorization form or to protect against XSS attacks in the sign-in form, you must add an XSLT action to the assembly for the OAuth 2.0 provide API. This XSLT sets the value of the `Content-Security-Policy` response header. Use the following XSLT to set this header.

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:dp="http://www.datapower.com/extensions"
    extension-element-prefixes="dp"
    exclude-result-prefixes="dp">

  <xsl:template match="/">
    <dp:set-http-response-header name="'Content-Security-Policy'" value="'DESIRED_VALUE'" />
  </xsl:template>

</xsl:stylesheet>
```

To protect against XSS attacks from external or inline sources in the sign-in form, use the `dp:set-http-response-header` element to define the `Content-Security-Policy` response header in the following way.

```
<dp:set-http-response-header name="'Content-Security-Policy'" value="'default-src 'self'; style-src
'unsafe-inline''" />
```

- **Creating a custom sign-in form**
  You can create a custom sign-in form for the OAuth identity extraction stage.
- **Creating a custom authorization form**
  You can create a custom authorization form for the OAuth authorization stage.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Creating a custom sign-in form

You can create a custom sign-in form for the OAuth identity extraction stage.

Before you begin

- You need an existing API. For more information, see Creating API definitions.
- You need an OAuth security definition that uses Implicit Flow, Password Flow, or Authorization Code Flow. For more information, see Protecting an API with OAuth security definition.

About this task

During OAuth processing, you can configure the OAuth 2.0 provider API to present users with a form to sign in to the service provided by the API. You can present a custom form or the default form. A custom form must fulfill certain requirements.

The names of the fields that inject information into your form are case-sensitive.

Procedure

To create a custom sign-in form for your OAuth definition, complete the following steps:

1. Create a well formed HTML document that will be parsed and transformed by IBM® API Connect to inject hidden fields.
2. For your HTML 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 IBM 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_5.0.0_com.ibm.apic.toolkit.doc_task_apionp
   rem_Create_a_custom_login_form_login_button" 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: Based on the value of the **Content-Security-Policy** response header, add elements that are loaded from external sources, such as images or JavaScript.
   For example, <script src="http://www.example.com/example.js" />
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_5.0.0_com.ibm.apic.toolkit.doc_task_apionp
    rem_Create_a_custom_login_form_login_button" 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 entry 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 OAuth 2.0 provider API to use custom forms for authentication and provide the URL at which your form is available. For more information, see Step 10.d of Creating an OAuth provider API.

Related tasks

- Protecting an API with OAuth security definition
- Creating a custom authorization form

Related information

- Configuring API security

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

Creating a custom authorization form

You can create a custom authorization form for the OAuth authorization stage.

Before you begin

- You need an existing API. For more information, see Creating API definitions.
- You need an OAuth security definition that uses Implicit Flow, Password Flow, or Authorization Code Flow. For more information, see Protecting an API with OAuth security definition.

About this task

During the OAuth processing, you can configure the OAuth 2.0 provider API to present users with a form that grants permission to an application that accesses their data through the API on their behalf. You can present a custom form or the default form. A custom form must fulfill certain requirements.

The names of the fields that inject information into your form are case-sensitive.

Procedure

1. Create a well formed HTML document that is parsed and transformed by IBM® API Connect to inject hidden fields.
2. For your HTML 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 IBM API Connect will replace with input fields necessary for the completion of the OAuth processing.
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: Have an error message displayed 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. Include additional hidden fields to provide context for the OAuth processing.
   Although you will likely want to include additional text to provide context for the user, use the following commands to include information relevant to the OAuth flow.
   - `<input type="hidden" name="dp-state" value="A"/>`
   - `<input type="hidden" name="resource-owner" value="A"/>`
   - `<input type="hidden" name="dp-data" value="A"/>`
   - `<input type="hidden" name="redirect_uri" value="A"/>`
   - `<input type="hidden" name="scope" value="A"/>`
   - `<input type="hidden" name="original-url" value="A"/>`
   - `<input type="hidden" name="client_id" value="A"/>`
   - `<input type="hidden" name="miscinfo" value="A"/>`
7. Optional: Based on the value of the **`Content-Security-Policy`** response header, add elements that are loaded from external sources, such as images or JavaScript.
   For example, `<script src="http://www.example.com/example.js" />`
8.
9. Insert spacing and additional elements as you require. Completing Steps 1 through to 7 results in a form similar to the following example.
   V5.0.6 and earlier

```
<html lang="en" xml:lang="en">
  <head>
    <title>Request for permission</title>
  </head>
  <body class="login">
    <div>
      <div>
        <form method="post" enctype="application/x-www-form-urlencoded" action="authorize">
          <input type="hidden" name="dp-state" value="A"/>
          <input type="hidden" name="resource-owner" value="A"/>
          <input type="hidden" name="dp-data" value="A"/>
          <input type="hidden" name="redirect_uri" value="A"/>
          <input type="hidden" name="scope" value="A"/>
          <input type="hidden" name="original-url" value="A"/>
          <input type="hidden" name="client_id" value="A"/>
          <input type="hidden" name="miscinfo" value="A"/>
          <AZ-INJECT-HIDDEN-INPUT-FIELDS/>
          <p>Greeting </p><DISPLAY-RESOURCE-OWNER/>
          <p>This application </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>
```

V5.0.7+

```
<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>
```

10. Make your form available at a URL of your choice.
11. If you have not already done so, configure your OAuth 2.0 provider API to use custom forms for authorization and provide the URL at which your form is available. For more information, see Step 10.f of Creating an OAuth provider API.

Related tasks

- Protecting an API with OAuth security definition
- Creating a custom sign-in form

Related information

- Configuring API security

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Protecting an API with OAuth security definition

When you create an OAuth security definition in an API, you provide settings for controlling access to the API operations through the OAuth authorization standard.

Before you begin

If you have not already done so, create an OAuth provider API. For more information, see Creating an OAuth provider API. The same provider API can support multiple security definitions and APIs.

Micro Gateway only V5.0.3+ OAuth 2.0 is supported by the Micro Gateway only from version 5.0.3 and onwards.

About this task

Each token grants access to a specific site for specific resources for a defined duration. By using an OAuth token, a user can grant a third-party site access to a range of information, which is stored with another service provider, without needing to share their personal credentials. For more information, see Tokens.

Note: The API Manager UI also includes the ability to create and edit security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

Procedure

To create an OAuth security definition, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. 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 API definitions.
5. Navigate to the Security Definitions section.
6. In the Security Definitions section, click the Add Security Definition icon ⊕.
7. Select OAuth.
   A new OAuth definition is created.
8. Configure your new security definition.
   a. In the Name field, provide a name for your OAuth definition to replace the default name.
   b. Optional: In the Description field, provide a description of your OAuth definition.
   c. For the Flow drop-down menu, select Implicit, Password, Application, or Access Code depending on the required OAuth security definition grant type.
   d. Specify the authorization url endpoint.
   e. V5.0.7+ If you select a grant type of Access Code, you are provided with an additional entry field in the user interface, named token URL endpoint.
      Note: The authorization url endpoint and token URL endpoint are maintained only for informative purposes, no validation or other action is applied to them by API Connect.
   f. V5.0.6 and earlier Complete one or both of the following steps depending on the OAuth security definition grant type:
      - For Password, Access Code, and Application flows: in the Token URL field, provide the URL of your OAuth provider API's token Path.
        The token path takes the following form:

        ```
        Host_Address/Organization_Segment/Catalog_Segment/API_Base_Path/Token_Path
        ```

        where:
        - *Host_Address* is the address of your gateway host.
        - *Org_Segment* is the path segment of the organization that you want to use.
        - *Catalog_Segment* is the path segment that is used by the Catalog you intend to publish your API in.
        - *API_Base_Path* is the base path of the OAuth provider API.
        - *Token_Path* is the path of either of your token operations. Typically, this is `oauth2/token`.
      - For Access Code and Implicit flows: in the Authorization URL field, provide the URL of your OAuth provider API's authorization Path.
        The authorization path takes the following form:

        ```
        Host_Address/Org_Segment/Catalog_Segment/API_Base_Path/Auth_Path
        ```

        where:
        - *Host_Address* is the address of your gateway host.
        - *Org_Segment* is the path segment of the organization that you want to use.
        - *Catalog_Segment* is the path segment that is used by the Catalog you intend to publish your API in.
        - *API_Base_Path* is the base path of the OAuth provider API.
        - *Auth_Path* is the path of your authorization operation. Typically, this is `oauth2/authorize`.

9. You can use either scope or introspect URL to verify an access token. If the access token is generated by the IBM® API Connect platform, use `scope`. If the access token is generated by a third party entity, use the introspect URL.
   Scope validation

   - Define any scopes that you want to cover access to your API.
     - In the Scopes section, click the Add scope icon ⊕.
     - Enter the name of the scope.
     - Enter a description.
   - Set Advanced Scope Check.
     - Use the Enable Scope Check slider to set the option to provide extra scope verification in addition to previously defined Scope settings. For more information, see OAuth Scope.

     Note:
     All scopes listed by the security definition of the secured API must be granted by the access token issued by your API Connect OAuth provider.

   - V5.0.7 + Define the introspect endpoint by entering an introspect URL. An introspect endpoint is what API Connect uses to verify a token issued by a third party OAuth provider.
     Note: The introspect URL conforms to the OAuth 2.0 specification: https://tools.ietf.org/html/rfc7662 with regards to requests and responses. This feature also provides a header to pass additional information to your provider, see Integrating third party OAuth provider.
10. Configure your security settings.
    a. In the Security section, click the Add icon to create additional options.
11. Click the Save icon 💾 to save your changes.

What to do next

Creating an OAuth provider API

Create an application. For more information, see Create an application.

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.

Configure a metadata URL for sending request headers to. For details, see OAuth metadata URL and authentication URL

Related information

- The OAuth 2.0 Framework

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.6 +

OAuth metadata URL and authentication URL

You can use the Metadata URL or Authentication 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 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 APIConnect 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 a metadata within the access token.
- A confirmation code that needs to be sent to the client application along with access token.

Configuring Metadata URL or Authentication URL in API Connect to obtain metadata

Metadata can be set by using either or both of the following URLs:

- Metadata URL - When you call the Metadata 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 Metadata URL is configured in the API Connect UI when you configure OAuth-2 Provider API. When you save the API, the OpenAPI (Swagger 2.0) file is updated with a new section under `oauth2:`

```
metadata:
   metadata-url:
      url: 'your_Metadata_URL'
      tls-profile:
```

Both of the URLs can send the following two HTTP headers in their response:

`API-OAUTH-METADATA-FOR-PAYLOAD`

`API-OAUTH-METADATA-FOR-ACCESSTOKEN`

The response header value from `API-OAUTH-METADATA-FOR-PAYLOAD` is placed in the response payload and indicated as `metadata`.

The response header value from `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...cL0Mv6ctl37w7ZU",
"metadata":  "m:metadata-for-payload_content"
"expires_in":  3600,
"scope":  "read",
"refresh_token":  "AAEnj5SynCMybF...oEZ6JjxYax_HdNg",
}
```

This example output from 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 Metadata URL

The following request headers are sent to the Metadata 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
...
```

Retrieving Metadata in APIConnect

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. The Creating an OAuth provider API topic has instructions to enable token introspection.

Refresh tokens and metadata

Authentication URL (if configured for authentication) is invoked first, during authentication of the resource owner. Metadata URL is invoked as the last step, just before the generation of the access token. The only exception, is when you generate access tokens from a refresh token. In cases where refresh tokens are used to generate new access tokens, the Metadata 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

There are two places from where metadata can be set, the metadata is prefixed with keywords to differentiate between the two sources.

- Metadata from the Metadata 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 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 payload does not have a specific size restriction, except for when you use the Authorization code grant type, as described in the following sections.

Characters 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 below include `authorization code`, `implicit`, and `client credentials`.

- **Authorization code grant type**:
  See the following section for details of the payload and token size limits for this grant type.

  Example of the `authorization 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/introspectp1/oauth2/token
{ "token_type":"bearer",
"access_token":"AAEkOThlZDhhNjYtYTQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ20ZN5Tl_TqYFeIfB7BFf6HFgibEoOjWX
XEA84oFsWuE4NY-RRZVdnGSaXAIYJz7s5vczfk5EV-BIb_6P_1YKm3ahrfhR5kI3sPO0uADEoseIP5-
O9anUpEM5yhsayXvZbJ_6VDYz-hyXSJHTNqVj-PHBialoRkBD5qca6kO0fV2M", "metadata":"a:[Authorization Code-
Test-auth-url-payload]", "expires_in":3600, "scope":"scope_introspect",
"refresh_token":"AAFAg1EVMbwicr_L0fTZ4q6HZ-
RcQygniXFC9zbSKO4wd3hcniC4KQX21X0fL2c8cnmzCZgws8xxLzNyfjQhUJNGl5C1GbIe3dwhXJdiWA0Go-
dduhVtCbG26sJRRXyYrMeRxWnJsylBETPI8HQEN4a_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/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=AAEkOThlZDhhNjYtYTQ1ZS00YTMzLWE0N2QtZmE2OGZkMzQ0NzQ2buS2KfWdq-
  nYBJSi4nmPxQBtLae17tKBPRMzwP5BC386nlxpoOTE1G748ZVH6Mq_TJL3GeV3PtXTVIkWOLBJi_7tljiQfpnVfrNkovvZkhUex
  YmFkcDsmLSdaWxZ6PcIMPAC4ojT8qV1sYV-
  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.

Authorization Code grant type size limitations

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.

Behavior when retrieving metadata with both Metadata URL and Authentication URL

If a Metadata 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 Metadata URL.

If a Metadata URL is configured, but the connection to Metadata URL fails, then a failure message of “`error on metadata url`” is written for metadata in both the payload and the access token.

If a Metadata 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: Metadata URL overwrites existing values from Authentication URL. This includes blank values.
If no Metadata URL is configured, the metadata that is obtained from the Authentication URL is retained as the final value.

Sample gatewayscript for simulating a Metadata URL endpoint

This sample gatewayscript can be used in the `gws policy` in the assembly of an API in API Connect for simulating a Metadata URL endpoint.

```
// Get the input headers that contain the exiting headers from the Authentication URL if any.
var existingToken = apim.getvariable('request.headers.x-existing-metadata-for-access-token');
var existingPayload = apim.getvariable('request.headers.x-existing-metadata-for-payload');
// Append metadata to the ones obtained from auth URL
apim.setvariable('message.headers.API-OAUTH-METADATA-FOR-ACCESSTOKEN', existingToken + ' token-from-
metadata-url');
apim.setvariable('message.headers.API-OAUTH-METADATA-FOR-PAYLOAD', existingPayload + ' payload-from-
metadata-url');
```

Related concepts

- V5.0.6 + Authentication URL

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

OAuth responses

API Connect returns various responses to requests during the OAuth 2.0 process.

Micro Gateway only V5.0.3 + OAuth 2.0 is only supported by the Micro Gateway from version 5.0.3 and onwards.

This topic lists possible responses for the following requests:

- Request through URL (implicit flow)
- Request through URL (access code flow)
- Request for access token (incorrect grant type)
- Request for access token (public access code flow)
- Request for access token (confidential access code flow)
- Request for an access token (application flow)
- Request for an access token (public password flow)
- Request for an access token (confidential password flow)

Request through URL (implicit flow)

When requesting an access token during an implicit flow, the application directs the user to the following URL:

```
Authorization_Endpoint_URL?response_type=token&
redirect_uri=Redirect_URI&scope=Scope&
client_id=Client_ID
```

where:

- *Authorization_Endpoint_URL* is the authorization endpoint of the API from which the application is requesting access.
- *Redirect_URI* is the redirect URI of the application.
- *Scope* is the scope of access the application is requesting.
- *Client_ID* is the client ID of the application.

Requesting a token can result in the following responses:

- If authorization is granted by the user and the request is successful, the user is redirected to the following URL:

```
Redirect_URI#access_token=Access_Token&expires_in=3600&
scope=Scope&token_type=bearer
```

where *Access_Token* is the access token requested.

- If authorization is not granted by the user and the request would otherwise be successful, the user is redirected to the following URL:

  ```
  Redirect_URI#error=access_denied
  ```

- If the *Redirect_URI* is incorrect, the user is not redirected and is presented with the following message:

  ```
  OAuth Error
  An error occurred while processing the OAuth request.
  Error:invalid_request
  Error Description:invalid redirect_uri
  ```

- If the *Redirect_URI* is incorrect in the request and present in the application, or if it is missing in the application, the user is presented with the following message:

  ```
  { "error":"unknown" }
  ```

  In this situation, the flow can still proceed if no other errors occur.

- If the *Scope* is incorrect or missing, the user is redirected to the following URL:

  ```
  Redirect_URI#error=invalid_scope
  ```

- If the *Client_ID* is incorrect, the user is presented with the following message:

  ```
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

Request through URL (access code flow)

When requesting an access token during an access flow, the application directs the user to the following URL:

```
Authorization_Endpoint_URL?response_type=code&
redirect_uri=Redirect_URI&scope=Scope&
client_id=Client_ID
```

where:

- *Authorization_Endpoint_URL* is the authorization endpoint of the API from which the application is requesting access.
- *Redirect_URI* is the redirect URI of the application.
- *Scope* is the scope of access the application is requesting.
- *Client_ID* is the client ID of the application.

Requesting an authorization code can result in the following responses:

- If authorization is granted by the user and the request is successful, the user is redirected to the following URL:

  ```
  Redirect_URI?code=Authorization_Code
  ```

  where *Authorization_Code* is the requested code.

- If the *Redirect_URI* is incorrect, the user is not redirected and is presented with the following message:

  ```
  OAuth Error
  An error occurred while processing the OAuth request.
  Error:invalid_request
  Error Description:invalid redirect_uri
  ```

- If the *Scope* is incorrect or missing, the user is redirected to the following URL:

  ```
  Redirect_URI#error=invalid_scope
  ```

- If the *Redirect_URI* is incorrect in the request and present in the application, or if it is missing in the application, the user is presented with the following message:

  ```
  { "error":"unknown" }
  ```

  In this situation, the flow can still proceed if no other errors occur.

- If the *Client_ID* is incorrect, the user is presented with the following message:

  ```
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

- If authorization is not granted by the user and the request would otherwise be successful, the user is redirected to the following URL:

```
Redirect_URI#error=access_denied
```

Request for access token (incorrect grant type)

If an incorrect grant type is provided, API Connect cannot identify the grant type, and therefore the flow, that was intended to be used.

When requesting an access token during an access code, application, or password flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

Part of the request header sent by the application includes the following information:

```
grant_type=Grant_Type
```

where *Grant_Type* corresponds to the OAuth flow in use and can take the following values:

- `authorization_code`
- `client_credentials`
- `password`

If *Grant_Type* is not one of the previous options, the following response is returned:

```
400 Processed
{ "error":"invalid_request" }
```

Request for access token (public access code flow)

When requesting an access token during a public access code flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

The application includes the following information in its request header:

```
grant_type=authorization_code
code=Authorization_Code
redirect_uri=Redirect_URI
client_id=Client_ID
```

where

- *Authorization_Code* is the code obtained when the user granted authorization.
- *Redirect_URI* is the redirect URI of the application.
- *Client_ID* is the client ID of the application.

Requesting an access token can result in the following responses:

- If the request is successful and refresh tokens are not enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/Scope}
  ```

  Where *Access_Token* is the requested access token and *Scope* is the scope of access that is allowed by the token.

- If the request is successful and refresh tokens are enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/scope,"refresh_token":Refresh_Token}
  ```

  where *Refresh_Token* is a token that can be used to obtain another access token once the original has expired.

- If *Authorization_Code* is invalid, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_grant" }
  ```

- If *Redirect_URI* is missing, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_request" }
  ```

  Note:

When an application requests an access token (after obtaining an authorization code) during a public or confidential access code flow (`grant_type=authorization code`), and the request header is missing the Redirect URI, the method by which the application was created makes a difference in the API Connect OAuth response behavior.

An application can be created through the Developer Portal (see [Applications in the Developer Portal]()), or created in an API Manager community (see [Developing your V5 APIs and applications]()):

  - If the application is created through the Developer Portal, there is no default Redirect URI. In this case, if the application does not supply a Redirect URI during an authorization request, the response is: `400 Processed {`
`"error":"invalid_request"`
`}`.
  - If the application is created through the API Manager community, API Connect automatically provides a default Redirect URI of [https://localhost](). In this case, if the application does not supply a Redirect URI, the default URI is used, and the response code is `200 OK`. The default URI also applies to authorization and token endpoint calls.

- If *Redirect_URI* is incorrect, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_grant" }
  ```

- If *Client_ID* is incorrect or missing, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

Request for access token (confidential access code flow)

When requesting an access token during a confidential access code flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

The application includes *Client_ID* as a user name and *Client_Secret* as a password. The application also includes the following information in its header:

```
grant_type=authorization_code
code=Authorization_Code
redirect_uri=Redirect_URI
```

where

- *Authorization_Code* is the code obtained when the user granted authorization.
- *Redirect_URI* is the redirect URI of the application.
- *Client_ID* is the client ID of the application and is used as a user name.
- *Client_Secret* is the client secret of the application and is used as a password.

Requesting an access token can result in the following responses:

- If the request is successful and refresh tokens are not enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/Scope}
  ```

  Where *Access_Token* is the requested access token and *Scope* is the scope of access that is allowed by the token.

- If the request is successful and refresh tokens are enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/scope,"refresh_token":Refresh_Token}
  ```

  where *Refresh_Token* is a token that can be used to obtain another access token once the original has expired.

- If *Authorization_Code* is invalid, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_grant" }
  ```

- If *Authorization_Code* is missing, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_request" }
  ```

- If *Redirect_URI* is missing, the following response is returned:

```
400 Processed
{ "error":"invalid_request" }
```

For more information on the response for a missing *Redirect_URI*, see the Note in Request for access token (public access code flow).

- If *Redirect_URI* is incorrect, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_grant" }
  ```

- If *Client_ID* or *Client_Secret* is incorrect or missing, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

Request for an access token (application flow)

When requesting an access token during an application flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

The application uses *Client_ID* as a user name and *Client_Secret* as a password. The application also includes the following information in its request header:

```
grant_type=client_credentials
scope=Scope
```

where

- *Client_ID* is the client ID of the application.
- *Client_Secret* is the client secret of the application.
- *Scope* is the scope of access requested.

Requesting an access token can result in the following responses:

- If the request is successful, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/Scope}
  ```

  Where *Access_Token* is the requested access token and *Scope* is the scope of access that is allowed by the token.

- If *Client_ID* or *Client_Secret* is incorrect or missing, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

- If *Scope* is incorrect or missing, the following response is returned:

  ```
  401 Processed
  { "error":"invalid_scope" }
  ```

Request for an access token (public password flow)

When requesting an access token during a public password flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

The application includes the following information in its request header:

```
grant_type=password
client_id=Client_ID
username=User_Name
password=Password
scope=Scope
```

where

- *Client_ID* is the client ID of the application.
- *User_Name* is the user name of the user whose credentials the application is using.
- *Password* is the password of the user whose credentials the application is using.
- *Scope* is the scope of access requested.

Requesting an access token can result in the following responses:

- If the request is successful and refresh tokens are not enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/Scope}
  ```

  Where *Access_Token* is the requested access token and *Scope* is the scope of access that is allowed by the token.

- If the request is successful and refresh tokens are enabled, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/scope,"refresh_token":Refresh_Token}
  ```

  where *Refresh_Token* is a token that can be used to obtain another access token once the original has expired.

- If *Client_ID* or *Client_Secret* is incorrect or missing, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

- If *Scope* is incorrect or missing, the following response is returned:

  ```
  401 Processed
  { "error":"invalid_scope" }
  ```

- If *Password* or *User_Name* is missing, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_request" }
  ```

- If *Password* or *User_Name* is incorrect, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_grant" }
  ```

Request for an access token (confidential password flow)

When requesting an access token during a confidential password flow, the application makes a POST call to the following URL:

```
Token_Endpoint_URL
```

The application uses *Client_ID* as a user name and *Client_Secret* as a password. The application also includes the following information in its request header:

```
grant_type=password
username=User_Name
password=Password
scope=Scope
```

where

- *Client_ID* is the client ID of the application.
- *Client_Secret* is the client secret of the application.
- *User_Name* is the user name of the user whose credentials the application is using.
- *Password* is the password of the user whose credentials the application is using.
- *Scope* is the scope of access requested.

Requesting an access token can result in the following responses:

- If the request is successful, the following response is returned:

  ```
  200 OK
  {"token_type":"bearer","access_token":"Access_Token",
  "expires_in":3600,"scope":"/Scope}
  ```

  Where *Access_Token* is the requested access token and *Scope* is the scope of access that is allowed by the token.

- If *Client_ID* or *Client_Secret* is incorrect or missing, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_client", "error_description":"client_id unauthorized" }
  ```

- If *Scope* is incorrect or missing, the following response is returned:

  ```
  401 Processed
  { "error":"invalid_scope" }
  ```

- If *Password* or *User_Name* is missing, the following response is returned:

  ```
  400 Processed
  { "error":"invalid_request" }
  ```

- If *Password* or *User_Name* is incorrect, the following response is returned:

  ```
  401 Unauthorized
  { "error":"invalid_grant" }
  ```

Related concepts

- OAuth user scenario

Related information

- Tutorial: Securing an API by using OAuth 2.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

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:

1. OAuth provider API returns the original request rather than the requested access token
2. Enable an extended error description
3. OAuth 2.0 support in the developer portal test tool
4. OAuth configuration with introspection URL can cause incorrect response on token revocation

OAuth provider API returns the original request rather than the requested access token

If an access token request that is sent to an OAuth provider API has an invalid URL, the OAuth provider API can become corrupted and thereafter return only the original request rather than the access token, even if the URL is valid.

For example, this problem occurs if the basepath for the OAuth Provider API is set to / and the access token request URL `https://hostname/organization_name/catalog_name//oauth2/token` is used. Subsequent calls with the correct URL, `https://hostname/organization_name/catalog_name/oauth2/token`, return the original request rather than the access token.

To fix this problem, take the OAuth provider API offline, then put it back online; complete the following steps:

1. In the API Manager user interface, open the Catalog to which the containing Product has been published.
2. Select the Products tab.
3. Click the containing Product to expand it.
4. Alongside the OAuth Provider API, toggle the Offline/Online slider control.

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 swagger definition. You can use the test tool that is included with API Connect to test the OAuth 2.0 provider API. For more information, see Testing an API using the Developer Portal test tool, and Testing an API with the API Manager test tool.

OAuth configuration with introspection URL can cause incorrect response on token revocation

Consumer APIs that are protected with an OAuth security definition, when configured to use an introspection URL which points to DataPower Gateway or API Connect based introspection, may result in an incorrect response if a token is revoked using the **`/revoke`** endpoint. In this configuration, the introspection response is the ultimate factor for API authentication.

Workaround:

- If you want to use **`/oauth2/revoke`**, remove the introspection URL from the Consumer API.
- If an introspection URL to DataPower is required, use **`/oauth2/issued DELETE`** to revoke the token.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 API Connect, scopes have no inherent meaning. Instead, scopes are defined in the provider API so that an application can request an access token that is valid for one or more of the scopes that are defined in the provider API. 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 of the secured API must be granted by the access token. When an access token is requested from the provider API, multiple scopes are separated by spaces.

OAuth provider

For illustration purposes, the example provider API swagger file that is named provider.yaml defines three scopes that an application can request: **`checking`**, **`saving`**, and **`mutual`**.

`provider.yaml`

```
x-ibm-configuration:
    testable: true
    enforced: true
    phase: realized
    oauth2:
      client-type: confidential
      scopes:
        checking: Checking Account
        saving: Saving Account
        mutual: Mutual Fund
```

To provide more refined support for the OAuth scope handling, API Connect allows the Authentication URL extension to modify the scope value.

V5.0.8 + When you define an OAuth provider, two more extensions allow the flexibility to check and override what scope the application is allowed. The optional extensions are Enable Application Scope Check and Enable Owner Scope Check. The extensions can be set independently.

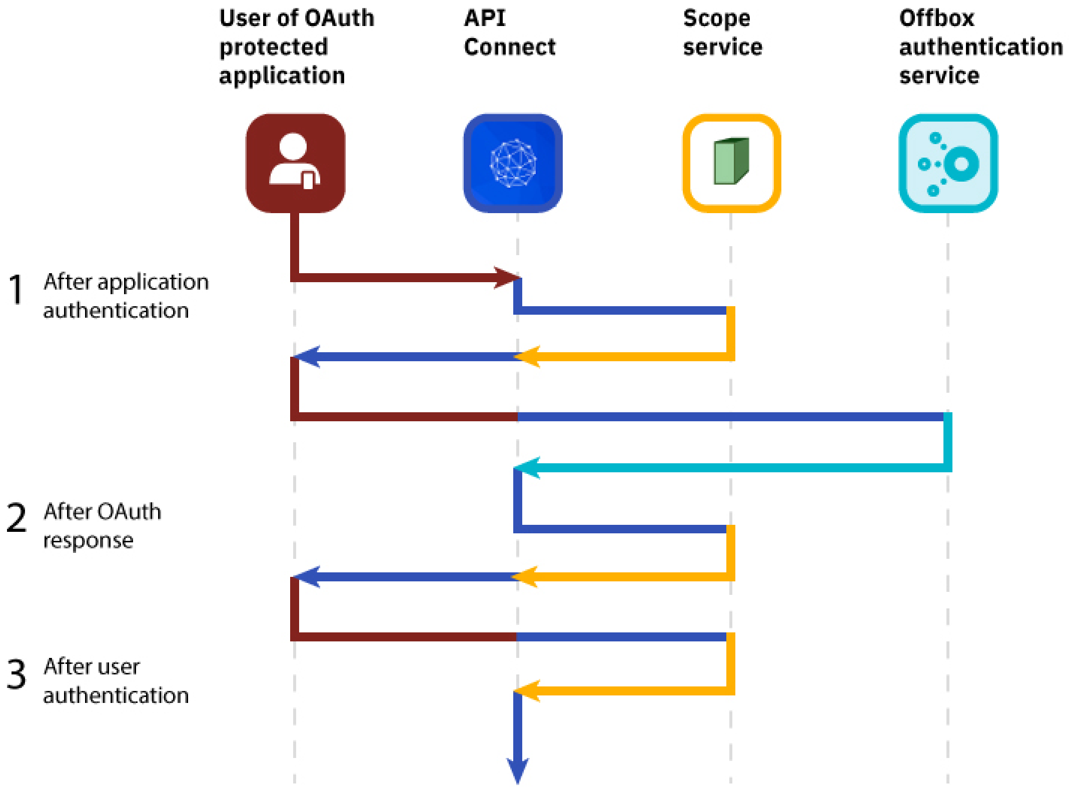
V5.0.8 + The scope that is eventually received by the application is determined by the interactions that are described in the three following paragraphs. Processing follows the sequence of paragraphs 1, 2, then 3, offering three distinct opportunities to override the scope value. Figure 1. provides an overview of the process.

1. From version 5.0.7.2, after the application successfully authenticates, and if OAuth 2 > Advanced Scope Check > Application Scope Check is configured, API Connect makes a call to allow extra verification and use 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.

V5.0.8 + 2. If OAuth 2 > Authentication > Authentication URL is enabled and configured with a valid URL, API Connect makes a call, as documented in [Authentication URL](). 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 paragraph 1. In the response header, `x-selected-scope` is an optional element.

V5.0.8 + 3. After the user successfully authenticates, and if OAuth 2 > 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. V5.0.8 +

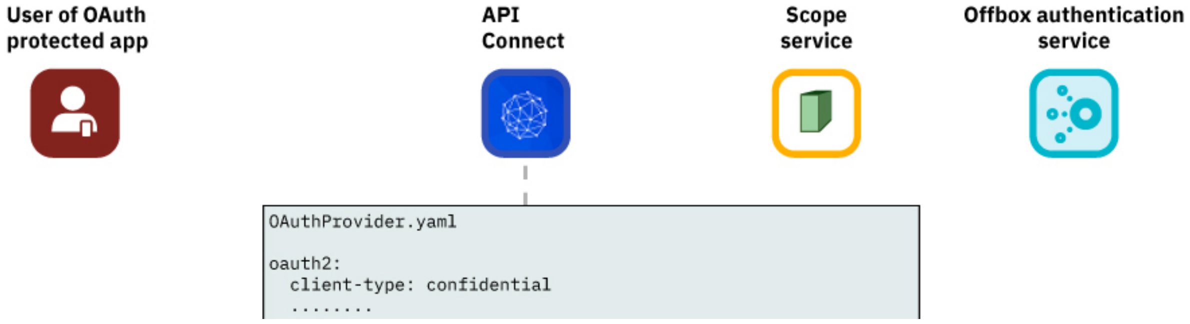
Figure 1. OAuth advanced scope overview


User of OAuth protected application
API Connect
Scope service
Offbox authentication service
1 After application authentication
2 After OAuth response
3 After user authentication

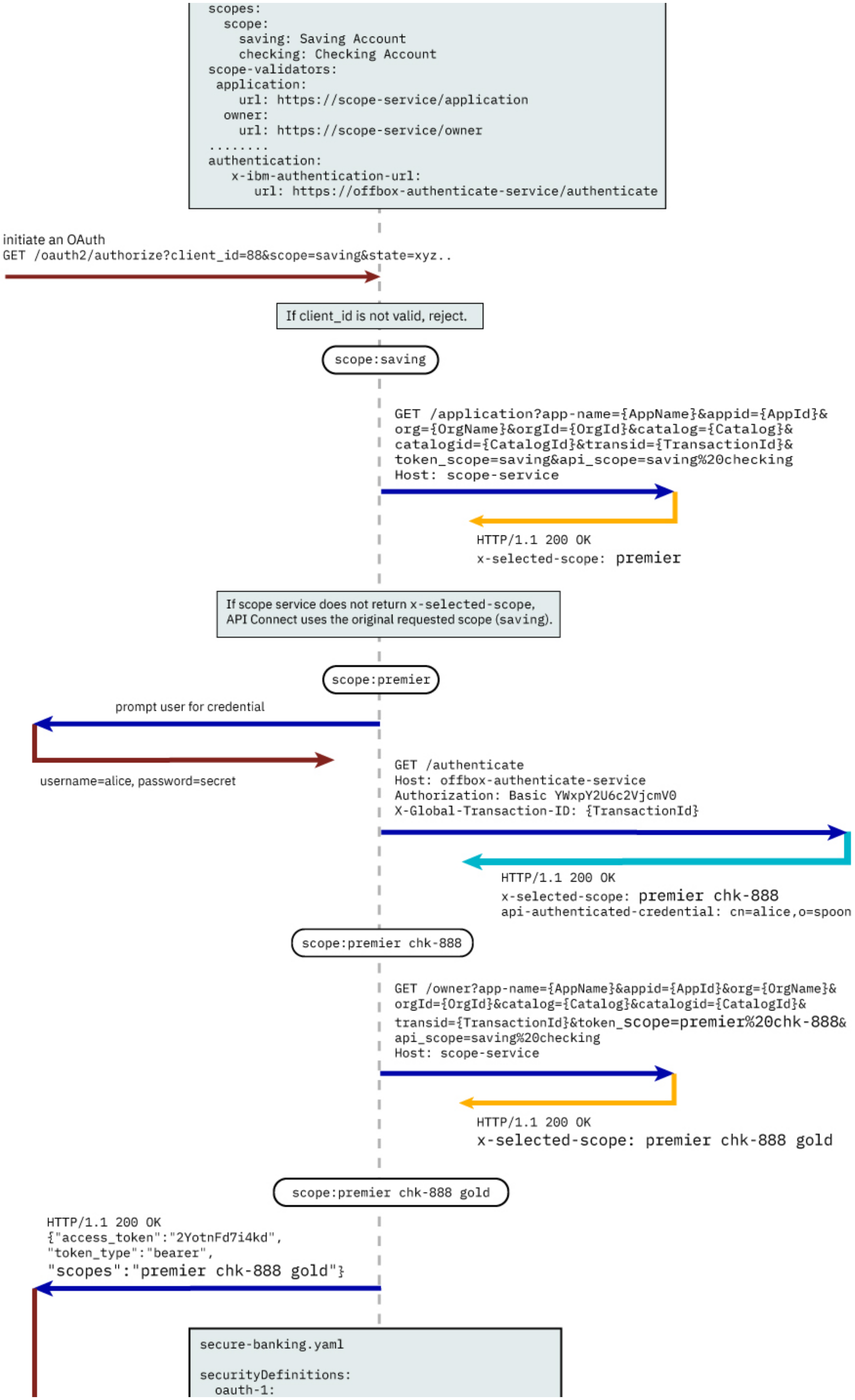
V5.0.8 + The final scope permission that is granted by the access token is the result of the described processing. Figure 2. shows a more detailed view of the transaction flow with examples that show when `x-selected-scope` provides a new scope value.

V5.0.8 +

Figure 2. OAuth advanced scope detail



```
OAuthProvider.yaml

oauth2:
  client-type: confidential
  ........
```



```
scopes:
  scope:
    saving: Saving Account
    checking: Checking Account
scope-validators:
 application:
    url: https://scope-service/application
  owner:
    url: https://scope-service/owner
........
authentication:
   x-ibm-authentication-url:
      url: https://offbox-authenticate-service/authenticate
```

```
GET /oauth2/authorize?client_id=88&scope=saving&state=xyz..
```

```
GET /application?app-name={AppName}&appid={AppId}&
org={OrgName}&orgId={OrgId}&catalog={Catalog}&
catalogid={CatalogId}&transid={TransactionId}&
token_scope=saving&api_scope=saving%20checking
Host: scope-service
```

```
HTTP/1.1 200 OK
x-selected-scope: premier
```

If scope service does not return x-selected-scope,
API Connect uses the original requested scope (saving).
scope:premier
prompt user for credential
username=alice, password=secret

```
GET /authenticate
Host: offbox-authenticate-service
Authorization: Basic YWxpY2U6c2VjcmV0
X-Global-Transaction-ID: {TransactionId}
```

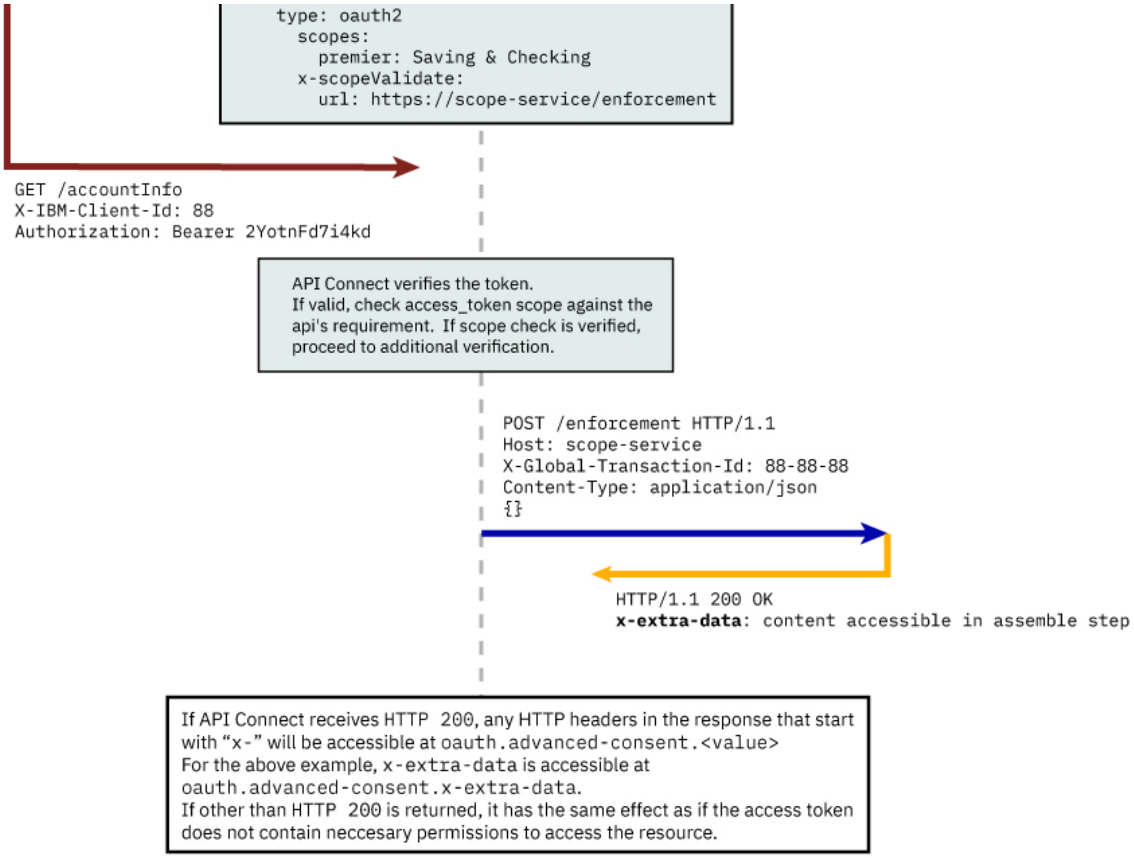
```
HTTP/1.1 200 OK
x-selected-scope: premier chk-888
api-authenticated-credential: cn=alice,o=spoon
```

```
GET /owner?app-name={AppName}&appid={AppId}&org={OrgName}&
orgId={OrgId}&catalog={Catalog}&catalogid={CatalogId}&
transid={TransactionId}&token_scope=premier%20chk-888&
api_scope=saving%20checking
Host: scope-service
```

```
HTTP/1.1 200 OK
x-selected-scope: premier chk-888 gold
```

```
HTTP/1.1 200 OK
{"access_token":"2YotnFd7i4kd",
"token_type":"bearer",
"scopes":"premier chk-888 gold"}
```

```
secure-banking.yaml

securityDefinitions:
  oauth-1:
```



```
type: oauth2
  scopes:
    premier: Saving & Checking
  x-scopeValidate:
    url: https://scope-service/enforcement
```

```
GET /accountInfo
X-IBM-Client-Id: 88
Authorization: Bearer 2YotnFd7i4kd
```

API Connect verifies the token.
If valid, check access_token scope against the api's requirement. If scope check is verified, proceed to additional verification.

```
POST /enforcement HTTP/1.1
Host: scope-service
X-Global-Transaction-Id: 88-88-88
Content-Type: application/json
{}
```

```
HTTP/1.1 200 OK
x-extra-data: content accessible in assemble step
```

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 neccesary 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 application API swagger file that describes the secured API.

```
GET /getaccount
HTTP/1.1
Host: server.example.com
X-IBM-Client-Id: 8888-8888-8888
Authorization: Bearer AAEkNjVkOWIyYjgtOWY5ZS00YWQwLWIyYzktZ
```

The following application API swagger 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 `AAEkNjVkOWIyYjgtOWY5ZS00YWQwLWIyYzktZ` is able to access the API because it contains one, or a combination of `scope-only` defined in secure-banking.yaml such as:

- `checking`
- `saving mutual`

- `checking saving mutual`

Advanced Scope Check
From version 5.0.7.2, 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 consumer API swagger 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 IBM® 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 the token does not contain necessary permission to access the resource.

```
     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.

Related information

- IETF RFC 6749

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tokens

If you are using OAuth authentication, you can enable refresh tokens.

Micro Gateway only V5.0.3+ OAuth 2.0 is only supported by the Micro Gateway from version 5.0.3 and onwards.

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 API Connect Token Endpoint are valid for 3600 seconds (1 hour), 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 AAEFYy1hbGxlhdS5nVX4x6iTL2sb3ymBivQb...
```

DataPower Gateway only After an access token expires, if the option is enabled in the OAuth provider API, the application can use refresh tokens. Each refresh token is valid for approximately 31 days after it is issued 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 Creating an OAuth provider API.

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 allowed by the API. For more information, see Protecting an API with OAuth security definition.

To request an access token, call the provider API's token endpoint operation with a POST call, including the following parameters:

- `client_id` in the header or query as is specified for the provider API or as a user name if using a confidential scheme.
- `client_secret` as a password if using a confidential scheme.
- `grant_type=refresh_token` in the header.
- `refresh_token=`*Refresh_Token* in the header, where *Refresh_Token* is your currently valid refresh token.

The response includes a new access token and a new refresh token.

- **OAuth token lifecycle management**
  API Connect provides two options to support OAuth token lifecycle management. Revocation is important to the token lifecycle.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

OAuth token lifecycle management

API Connect provides two options to support OAuth token lifecycle management. Revocation is important to the token lifecycle.

During OAuth processing, multiple types of token can be created. Certain type of tokens cannot be reused according to IETF 6749.

The API Connect OAuth token is self-contained, so any gateway with the known cryptographic material can issue and verify a token. Replay protection ensures that the same token cannot be reused.

To process request operations, API Connect must be aware of permissions that are issued. Monitoring issued permissions differs from determining the revocation of access rights for application or token. For information about how to enable the monitoring of issued permissions in the OAuth provider configuration, see Creating an OAuth provider API.

API Connect supports the following methods to manage token lifecycle that includes when to revoke access rights.

- DataPower gateway distributed cache support. For more information, see Managing tokens with the DataPower Gateway.
- An external service reached through the revocation URL. For more information, see OAuth revocation URL.

- **Managing tokens with the DataPower Gateway**
  API Connect can use the DataPower distributed cache to manage the token lifecycle that includes when to revoke access rights.
- **OAuth revocation URL**
  In IBM API Connect, you use an OAuth revocation URL to revoke or refresh specific access tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing tokens with the DataPower Gateway

API Connect can use the DataPower® distributed cache to manage the token lifecycle that includes when to revoke access rights.

Before you begin

To use the DataPower Gateway to manage tokens, the DataPower quota enforcement server must be enabled on the DataPower Gateway to use the distributed cache, see. Quota enforcement.

Note: This option is not supported for API Connect with IBM® Cloud public.

About this task

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.

The Allow user to view and revoke permissions and Allow application to revoke its token settings are set independently.

Procedure

Enable support for the DataPower distributed cache.

1. Click OAuth 2 > Tokens section.
2. Switch on the Enable revocation slider.
3. Select Use DataPower Gateway.

Optionally allow user to view and revoke permissions.

4. Select Allow users to View and revoke permissions.
   This option inserts 2 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 example. The API call to revoke a given application for a given user is shown in the second example.
   Note: View and revoke permissions should be limited to administrative applications because it allows applications subscribed to this OAuth provider API the ability to view and revoke permissions for other applications.
   View permissions example

   To list out 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: E2qM6mG2bX2uC1xT2iN1uU6bT5cV4dN7nW5kM5uP8vL3uF3cT7
            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" }
  ```

Optionally allow applications to self manage token lifecycle by revoking their own token.

5. Select Allow application to revoke its token.
   This option inserts 1 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 examples.

   Revoke `access_token` example

   - 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-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 a `refresh_token` example

   - 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

OAuth revocation URL

In IBM® API Connect, you use an OAuth revocation URL to revoke or refresh specific access tokens. If necessary, you can also revoke all tokens that are issued to a specific client ID or a resource owner.

Token revocation

An OAuth revocation URL provides a link to an external service that 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.

You can also use DataPower® gateway distributed cache support. For more information, see: Using Revocation with API Connect.

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 an access code flow, when exchanging the temporary authorization code for an access token, API Connect sends a GET request to the Revocation URL to determine if the temporary authorization code is valid, and (or) whether it has been used previously:

```
GET <revocationURL> HTTP/1.1
client-id: e515f992-cea3-4ff5-936e-39ec7fe80a40
resource-owner: alice
code: AAJgZEEoYF3el5xjZ_IUp_k6Q...
Cache-Control: no-transform
Host: token-mgmtsrv:10443
```

If the code is not in use, and therefore the request is valid, the OAuth revocation URL records the code and returns `HTTP 200`. If the OAuth revocation URL returns any response *other* than `HTTP 200`, the DataPower gateway service considers the request as failed and does not issue an access token. Possible reasons for failure include determining that the authorization code was previously used or could not be recorded by the revocation endpoint.

The maximum authorization code lifetime is 10 minutes.

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:

```
GET <revocationURL> HTTP/1.1
Accept: application/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>AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1...</token>
<token>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>
```

The revocation URL can respond with the access and refresh tokens that are to be revoked, or any resource owners for which all tokens are to be revoked.

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
X-Global-Transaction-ID: d37f0b165ce9273e132592f3
X-Transaction-ID: 402837825
Content-Length: 348
Host: token-endpoint:2775

<?xmlversion="1.0" encoding="UTF-8"?>
<token>
   <token_type>bearer</token_type>
   <access_token>AAENYy1hbGwtcmVmcmVzaOfNeQKX8ZeojsBY9v0FI7/OerQvzKHq...</access_token>
   <expires_in>3600</expires_in>
   <refresh_token>AAJnzJY3_EiHPDS6MVkcLV0ST9t5O_vq2I9VUrc_BEZG3qBgglITWOMysig-
1ORdc9DSjctGgcIM_bGNqOadTLQ-</refresh_token>
   <refresh_token_expires_in>2682000</refresh_token_expires_in>
   <scope>scope1</scope>
   <resource-owner>alice</resource-owner>
   <client_id>83d9cdcd-ba72-4d00-abae-005da8da5fb1</client_id>
</token>
```

The refresh_token and refresh_token_expires_in elements are required only if the refresh token is present. For the Access code grant type, a code parameter will be present.

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: AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1
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> HTTP/1.1
refresh-token: AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1
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: AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1
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: AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1
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: AAETb2F1dGgtcmV2b2tlLWN1c3RvbfZaRlVbnPSc1
client-id: 760d75a2-44b1-4485-8c6f-0d264fcf7398
resource-owner: alice
```

Revocation Service:

```
HTTP/1.1 200 OK
Content-Type: application/xml
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>
```

Note:

- In the previous example, there are no entries older than one month in the response (the maximum life of a refresh token).
- The **before** attribute uses the **xs:dateTime** syntax.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

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 must have created an API. For more information, see Creating API definitions.

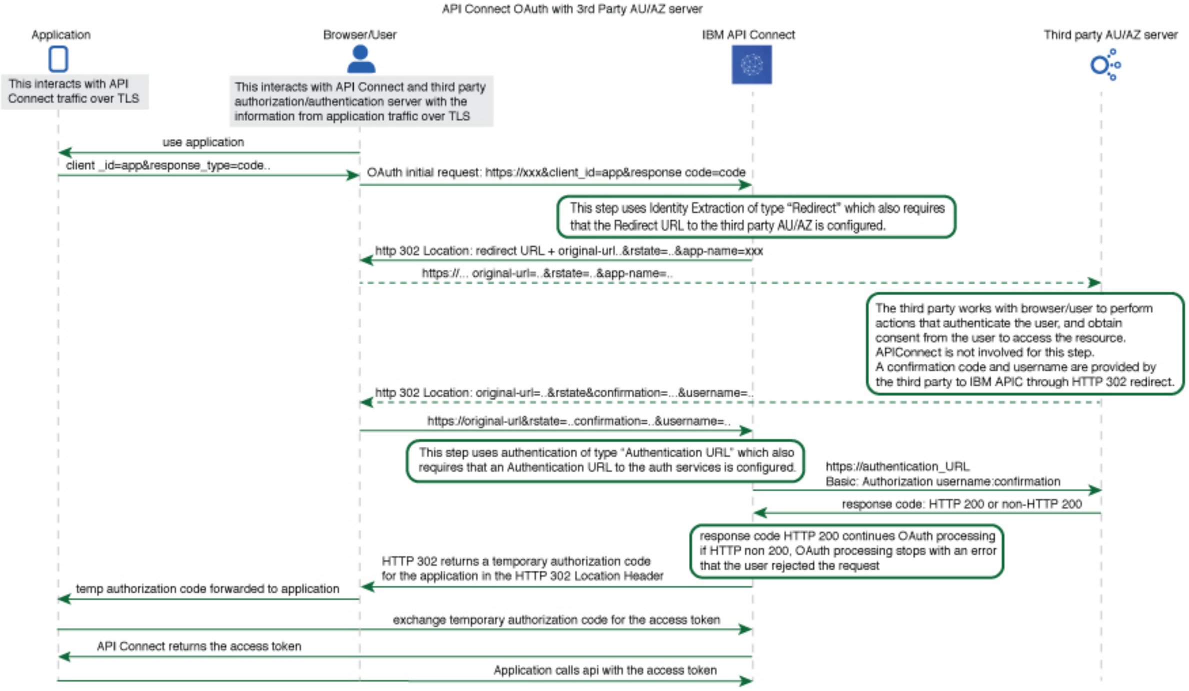
Additionally, you will need to either create or have created an OAuth security definition that uses Implicit Flow or Authorization Code Flow. For more information, see Protecting an API with OAuth security definition.

About this task

If you use methods for authentication that are not supported by IBM 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


API Connect OAuth with 3rd Party AU/AZ server
Application
Browser/User
IBM API Connect
Third party AU/AZ server
This interacts with API Connect traffic over TLS
This interacts with API Connect and third party authorization/authentication server with the information from application traffic over TLS
use application
client _id=app&response_type=code..
OAuth initial request: https://xxx&client_id=app&response code=code
This step uses Identity Extraction of type "Redirect" which also requires that the Redirect URL to the third party AU/AZ is configured.
http 302 Location: redirect URL + original-url..&rstate=..&app-name=xxx
https://... original-url=..&rstate=..&app-name=..
The third party works with browser/user to perform actions that authenticate the user, and obtain consent from the user to access the resource. APIConnect is not involved for this step. A confirmation code and username are provided by the third party to IBM APIC through HTTP 302 redirect.
http 302 Location: original-url=..&rstate&confirmation=...&username=..
https://original-url&rstate=..confirmation=..&username=..
This step uses authentication of type "Authentication URL" which also requires that an Authentication URL to the auth services is configured.
https://authentication_URL
Basic: Authorization username:confirmation
response code: HTTP 200 or non-HTTP 200
response code HTTP 200 continues OAuth processing if HTTP non 200, OAuth processing stops with an error that the user rejected the request
HTTP 302 returns a temporary authorization code for the application in the HTTP 302 Location Header
temp authorization code forwarded to application
exchange temporary authorization code for the access token
API Connect returns the access token
Application calls api with the access token

1. The application initiates a request to access an API protected with a third-party entity. IBM 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. IBM 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.
3. IBM 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 IBM 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.

Note: The API Manager UI also includes the ability to create and edit security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

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. 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.
   Except for **`original-url`**, all other parameters are provided only as information for the third-party entity to use or ignore. When a user is redirected to your page, the URL includes any of the following parameters that contain a value:

   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.

   V5.0.8 + catalogtitle
   : V5.0.8 + User-friendly display name for the catalog.

   g-transid
   : Same definition as transid. This parameter is provided only if its content is different from transid.

   org
   : The name of the developer organization that hosts the application.

   orgid
   : The id of the developer organization that hosts the application.

   V5.0.8 + orgtitle
   : V5.0.8 + 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 **`rstate`** is appended. The **`rstate`** is a hash code that is generated by API Connect for verification purposes. The URL is URL-encoded and must be decoded before further use, the **`rstate`** should be left unchanged.

   provider
   : The name of the API provider organization.

   providerid
   : The id of the API provider organization.

   V5.0.8 + providertitle
   : V5.0.8 + User-friendly display name for the provider organization.

   V5.0.8 + requested-scope
   : V5.0.8 + [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 gateway for the transaction that triggers 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&rstate=R_State&app-name=Application_Name`

with variables as described in the preceding list of query parameters. API Connect does not enforce a size limit on the length of the Redirect URL.

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=5af57a4
a-6db9-4141-ad08-
5709432af66e&rstate=5yXZSNocRPpJm9MZHR15MDc9hZhTiSRy10EhV28&username=spoon&confirmation=123456
78
```

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=5af57a4
a-6db9-4141-ad08-
5709432af66e&rstate=5yXZSNocRPpJm9MZHR15MDc9hZhTiSRy10EhV28&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. Click APIs.
   The APIs tab opens.
4. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
5. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
6. In your OAuth 2.0 provider API, 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 your provider API, see Creating an OAuth provider API, Steps 10.d and 10.e.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

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.
- V5.0.7 and earlier 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.
- If you apply more than one OAuth security definition to an API, they must all have the same client type setting, Public or Confidential.
- If you apply more than one OAuth security definition to an API, they must have compatible authentication settings. That is, the value of an authentication property must be the same for all applied OAuth security schemes that have that property.
- If you apply more than one OAuth security definition to an API, they must have compatible token refresh and revocation settings. That is, the value of a token refresh property or revocation property must be the same for all applied OAuth security schemes that have that property.

Note: The API Manager UI also includes the ability to apply security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

Procedure

To apply security definitions to an API, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. 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 API definitions.
5. Navigate to the Security section.
6. In the Security section, select the security definitions that you want to apply.
   Note: V5.0.7 and earlier 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.
7. To remove a security definition so that it is no longer applied to the API, clear the selection for that security definition.
8. Click the Save icon 💾 to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 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.

Note: The API Manager UI also includes the ability to apply security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

Procedure

To specify the security definition inheritance settings for an API operation, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. 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 API definitions and Defining Paths for a REST API.
5. Navigate to the Paths section.
6. In the Paths section, click the required operation to display its details.
7. In the Security subsection, select the security definitions that you want to apply, or select Use API security definitions to apply all security definitions.
   Note: V5.0.7 and earlier 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.
8. Click the Save icon 💾 to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

The API Connect implementation of CORS returns a Access-Control-Allow-Origin value that matches the requested Origin. This allows a broad range of origins. If you need tighter security you can disable CORS on the gateway and implement CORS as needed, using either a GatewayScript or XSLT policy in the assembly, custom policy, DataPower® extension, or on the backend.

Note: The API Manager UI also includes the ability to apply security definitions. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

Procedure

To enable CORS support for an API, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. To enable CORS support for an existing API, click the API that you want to work with. To create a new API before applying the scheme to it, see Creating API definitions.
5. Navigate to the Lifecycle section.
6. Ensure that the CORS slider is in the On position.
7. Click the Save icon 💾 to save your changes.
8. 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Securing your APIs with OpenID Connect

OpenID Connect (OIDC) is built on top of the OAuth 2.0 protocol and focuses on identity assertion. OIDC provides a flexible framework for identity providers to validate and assert user identities for Single Sign-On (SSO) to web, mobile, and API workloads.

About this task

OIDC uses the same grant types as OAuth (implicit, password, application and access code) but uses OIDC specific scopes, such as openid, with optional scopes to obtain the identity, such as email and profile. OIDC generates a JSON Web Token (JWT), rather than an opaque token with OAuth, that can optionally be signed and encrypted.

IBM® API Connect supports OIDC provider flow by building on top of the existing OAuth 2 provider capabilities. The additional OIDC functions are provided by using the API Connect API Assembly, which provides the flexibility to fully configure the protocol to meet your desired requirements.

A pre-supplied sample OAuth Provider API is provided for you to download and adapt, rather than creating your own from scratch. You must modify this API in accordance with your OIDC configuration.

Note: The following multivalued response types are supported:

- `code id_token`
- `id_token code`

Procedure

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. Import the pre-supplied sample OAuth Provider API as follows:
   a. Click Add > Import API from a file or URL.
   b. Click Or import from URL and in the URL field enter `https://raw.githubusercontent.com/ibm-apiconnect/openid/master/oidc_1.0.0.yaml`
   c. Click Import.
   The specified YAML file is imported, and the associated API definition, called OAuth 2 OIDC Provider, opens in the Design view.
5. Click the OAuth 2 OIDC Provider API and examine the API as follows:
   a. In the navigation pane, click Paths to display the paths that have been defined for the API.
   The token API is exposed on the paths **`/oauth2/authorize`** and **`/oauth2/token`**. The path **`/oauth2/introspect`** lets you obtain information about the access token.

   b. In the navigation pane, click OAuth 2.
   The default grant types are supported, but of particular importance are the available scopes. The scope **`openid`** triggers the OIDC flow. Add additional scopes for your applications here.

   c. Scroll down to the Authentication section and observe that the pre-defined authentication mechanism is Authentication URL. You must change the settings in this section in accordance with the authentication mechanism that you are using; for more information, see the Authentication section in Creating an OAuth provider API.

   d. Click the Assemble tab. You will notice several policies that control the generation of the JWT for OIDC flows.

You must customize the **set-variable** and **jwt-generate** policies in accordance with your OIDC configuration; for example, to provide custom claims and other JWT information. For more information, see Generate JWT (jwt-generate).

6. Modify the OAuth 2 OIDC Provider API in accordance with the guidance in step 5.
7. Click the Save icon 💾 to save your changes.
8. For any APIs that you want to secure with OIDC, complete the following steps:
   a. Open the Design view for the API.
   b. Navigate to the Security Definitions section.
   c. Click the Add Security Definition icon ⊕ and then click OAuth.
   d. Scroll down to your newly created OAuth security definition.
   e. Configure your OAuth definition as required, ensuring that you define a scope called `openid`, to match the scope that is defined in the pre-supplied sample OAuth Provider API.
   f. In the Security section, select the check box for your OAuth security definition, and ensure that the openid scope check box alongside it is also selected.
   g. Click the Save icon 💾 to save your changes.
9. Create a Product, add your APIs and the customized OAuth Provider API to the Product, and include them in a Plan in the Product; for details, see Creating a Product.
10. Stage the Product to your chosen Catalog; for details, see one or other of the following topics depending on whether you are working in IBM Cloud or in your own on-premises IBM API Connect deployment:
    - IBM Cloud: Staging a Product
    - On-premises: Staging a Product
11. Publish the Product; for details, see one or other of the following topics depending on whether you are working in IBM Cloud or in your own on-premises IBM API Connect deployment:
    - IBM Cloud: Publishing a new Product
    - On-premises: Publishing a new Product
12. Test the APIs by using an environment capable of sending HTTP requests and receiving HTTP responses. The following commands test the APIs by using the **curl** command in a command line interface. The commands here are for testing the OIDC Hybrid Flow; for testing other OIDC flows, you can use the same commands that you would use for the corresponding OAuth flows, but make sure that you include the **openid** scope.
    Note: To test the APIs, you must have an account on the Developer Portal. You must also have created an application that you have then used to subscribe to the Plan that contains the APIs that you want to test. For more information, see Registering an application and Signing up to use an API. You need to make a note of your Client ID and Client Secret so you can run the following commands.
    a. Log in and provide consent to the OAuth application. On successful completion, the OAuth application returns an authorization code and an ID token. The ID token is a JWT token that provides a set of claims.

```
curl -X GET \
  'Authorization_Endpoint_URL?
scope=openid%20App_Scopes&response_type=id_token%20code&client_id=Client_ID&redirect_uri=Redir
ect_URI' \
  -H 'authorization: Basic base64_encoded_username:password' \
  -H 'content-type: application/text' \
```

Replace *Authorization_Endpoint_URL*, *App_Scopes*, *Client_ID*, *Redirect_URI*, and *base64_encoded_username:password* with the appropriate values. *App_Scopes* is a space-separated list of any additional scopes that you defined when you configured the OAuth Provider API.
For example:

```
curl -X GET \
  'https://myhost.com:1234/myorg/mycatalog/oauth-end/oauth2/authorize?
scope=openid%20profile%20email&response_type=id_token%20code&client_id=51dfaed9-9d07-46fc-
83a6-52c09735a4a0&redirect_uri=https://example.com/redirect' \
  -H 'authorization: Basic c8Bpb156c3Bvd24=' \
  -H 'content-type: application/text' \
```

    b. Exchange the authorization code for an access token.

```
curl -X POST \
  Token_Endpoint_URL \
  -H 'content-type: application/x-www-form-urlencoded' \
  -d
'grant_type=authorization_code&redirect_uri=Redirect_URI&client_id=Client_ID&client_secret=Cli
ent_Secret&code=Authorization_Code
```

Replace *Token_Endpoint_URL*, *Redirect_URI*, *Client_ID*, *Client_Secret*, and *Authorization_Code* with the appropriate values.
For example:

```
curl -X POST \
  https://myhost.com:1234/myorg/mycatalog/oauth-end/oauth2/token \
  -H 'content-type: application/x-www-form-urlencoded' \
  -d
```

```
'grant_type=authorization_code&redirect_uri=https://example.com/redirect&client_id=51dfaed9-
9d07-46fc-83a6-
52c09735a4a0&client_secret=tW2cJ0cD0nY1qE6eH3yP8wQ2nT3eM5eV3cI1vU1qI6xN7fL7eS&code=AAL8k4uaJj3
PCi7L026rR1BheJkzWKzR7vWgWvo--DBqYRVckkwcL
```

c. Use the access token to invoke the API.

```
curl -X GET \
  'API_URL' \
  -H 'authorization: Bearer Access_Token' \
  -H 'x-ibm-client-id: Client_ID \
  -H 'x-ibm-client-secret: Client_Secret'
```

Replace *API_URL*, *Access_Token*, *Client_ID*, and *Client_Secret* with the appropriate values.
For example:

```
curl -X GET \
  'https://myhost.com:1234/myorg/mycatalog/current?zipcode=90210' \
  -H 'authorization: Bearer
AAEHZGVmYXVsdIX41OcqAfmg3xnrSMGmiqTLPhcQqFluuLdkWqxsaxudVMSBGBFY4KKdTa8NP7S3tJGi8K1vpDSam86j6W
1Feaa-l5VnuKOMETEQTNJNiwVXJl3eUNwDfw67pwpP0dg6e_VcUA1VnjnzDiAHVQyHKm8' \
  -H 'x-ibm-client-id: 51dfaed9-9d07-46fc-83a6-52c09735a4a0 \
  -H 'x-ibm-client-secret:
tW2cJ0cD0nY1qE6eH3yP8wQ2nT3eM5eV3cI1vU1qI6xN7fL7eS&code=AAL8k4uaJj3PCi7L026rR1BheJkzWKzR7vWgWv
o--DBqYRVckkwcL'
```

Related information

- Tutorial: Securing an API by using OAuth 2.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with Products in the API Designer

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. You use the API Designer to create, edit, and stage your Product, and the API Manager to manage the lifecycle of your Product.

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.
A diagram showing the heirarchy 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. V5.0.8 + 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 V5.0.8 + and cost. For example, you can have a "Demo Plan", which makes a single API available V5.0.8 + 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.

DataPower Gateway only V5.0.2 + 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.
- V5.0.3 + 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

IBM 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 Product**
  Create a 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 V5.0.8 +, and the subscription terms.
- **Defining a Plan**
  Define Plans to specify the limitations and subscription details of how developers can use your API Products. A Plan includes rate limit settings that can be applied to the Plan as a whole, or specified for each operation in an API. It can also include billing information that applies to the Product. Through Products and Plans, you have greater control over what APIs your developers have access to V5.0.8 +, and the subscription terms.
- **Staging a Product**
  Stage a 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. V5.0.5 + 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.
- **Creating a new version of your Product**
  You can have multiple versions of a Product. These versions can occupy any of the lifecycle stages, which facilitates development.

Related information

- Working with Products in the API Manager

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a Product

Create a 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 V5.0.8 +, and the subscription terms.

Before you begin

Define your APIs. For more information, see Creating API definitions.
Anybody using the API Designer can perform all actions within it. However, when staging or publishing a Product, permissions will be enforced by API Connect. For more information about user roles and permissions, see Administering user access.

Note: The API Manager user interface (UI), also includes the ability to create and edit Products. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.
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 Designer.

Procedure

To create a Product, complete the following steps:

1. To open the API Designer UI, enter the following command at the command line:

   ```
   apic edit
   ```

   The API Designer UI opens in your default browser.
   Note: If you are connected to a network, you are asked to sign in with IBM Cloud. Enter your IBMid credentials and click LOG IN. If this is your first time using API Designer, the Draft APIs information page opens, click Got it!, and continue with the following steps.
2. Select the Products tab in API Designer.
   The Products tab opens.
3. V5.0.4 + Click Add and then click New product.
   V5.0.3 and earlier Click Add and then click New Product.
4. V5.0.3 and earlier From the "Add a new product" window, complete the following steps:
   a. Specify a title, name, and version.
      Note: The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), and hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
   b. Click Add.
      The Design tab for the new Product opens.
5. V5.0.4 + From the "New product" window, complete the following steps:
   a. API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
   b. Specify a title, name, and version for the Product.
      Note: The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), and hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
   c. Click Create product.
      The Design tab for the new Product opens.
6. Optional: Enter description, contact, license, and terms of service information for the Product in the Info section.
7. In the Visibility section, specify the users that you want the Product to be visible to.
   You can choose Public, Authenticated users, or Custom.
   a. API Designer only If you select Custom, type the name of the organization or community that you want the Plans in the Product to be visible to in the Type to add field. For information about how to create and manage organizations or communities, see Administering Developer organizations.
   b. API Manager only If you select Custom, use the Type to add field to search for the developer organizations or communities that you want the Plans in the Product to be visible to.
      Note: To search for developer organizations or communities, the Product must be in the staged, published, or deprecated state. If the Catalog in which it is staged, published, or deprecated is not a development Catalog, you cannot make other changes to the Product while it is in one of these states. For more information, see The Product lifecycle.
8. Specify the users that can subscribe to the Product.
   You can choose Authenticated users or Custom.
   a. API Designer only If you select Custom, type the name of the organization or community that you want to be able to subscribe to the Plans in the Product in the Type to add field. For information about how to create and manage organizations or communities, see Administering Developer organizations
   b. API Manager only If you select Custom, use the Type to add field to search for the organizations or communities that you want to be able to subscribe to the Plans in the Product.
      Note: To search for developer organizations or communities, the Product must be in the staged, published, or deprecated state. If the Catalog in which it is staged, published, or deprecated is not a development Catalog, you cannot make other changes to the Product while it is in one of these states. For more information, see The Product lifecycle.
9. In the APIs section, specify the APIs that you want to include in the Product.
   a. Click the Add API icon ⊕.
   b. Select the APIs that you want to include, then click Apply.
      The selected APIs are listed.

   Note: To make an API available to application developers, you must include it in a Plan.
10. Add one or more Plans to the Product.
    a. Click the Add Plan icon ⊕.
    b. Expand the new Plan that has been created. If you have already added APIs to your Product, these are automatically included.
    c. Rename your Plan in the Title and Name fields, and optionally add a description.

    Note: A Default Plan is automatically created for you if you do not want to create your own, and any APIs that you selected in the previous step are automatically included in this Plan. You can clear the check box for any API to exclude it from the Plan, but at least one API must be included because a Product cannot be staged if it includes any Plans that do not include APIs. If you decide not to use the Default Plan, you must delete it.

11. Optional: V5.0.8 + Define the billing terms of the Plan.
    a. Select the billing model for the Plan. The monthly subscription is billed on the same day of each month.
    b. Select the currency and cost for the Plan.
    c. Select the number of days that customers can try the Plan before they are start incurring charges.
12. Verify that the APIs you require are included in the Plan.
    a. Expand the Plan to which you want to add APIs.
    b. Under APIs included, ensure that the check boxes of the APIs you require are selected. If there are APIs already selected, and you do not want them included in the Plan you are editing, clear their check boxes.

    Note: If you add the same API to more than one Plan, and the same Application (client ID) signs up for multiple Plans that contain the same API, the Gateway server cannot determine which Plan rate limit should be applied. If you anticipate that an API will be shared by an Application, a proxy API should be defined so that the correct Plan rate limit can be applied.
13. Optional: Select which operations from an API to include in the Plan.

    V5.0.0 ONLY V5.0.1 ONLY

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon •••.
    b. Select or clear the check boxes for the operations you want to include or exclude.

    V5.0.2 +

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon ⌄.
    b. Select or clear the check boxes for the operations you want to include or exclude.
14. Optional: Add a rate limit to your Plan.
    - V5.0.0 ONLY V5.0.1 ONLY Clear the Unlimited check box and then specify the rate limit you want to apply.
    - V5.0.2 + Click the Add new rate limit icon ⊕, and then specify the rate limit that you want to apply. You can set multiple rate limits, at second, minute, hour, day, and week time intervals.

      DataPower Gateway only To add burst limits to your Plan, so that you can prevent usage spikes that might damage infrastructure, click the Add new burst limit icon ⊕, and then specify the burst limit that you want to apply. You can set multiple burst limits, at second and minute intervals. To remove a rate or burst limit, click the Delete rate limit or Delete burst limit 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 Enforce 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 Enforce 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."

Enforce hard limit affects only the **rate limit**, as illustrated by the following scenarios:

Scenario A

Table 1. Enforce Hard Limit Enabled

Enforce 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. Enforce Hard Limit Disabled

Enforce Hard Limit	Burst Limit	Rate Limit

Enforce Hard Limit	Burst Limit	Rate Limit
OFF	100 calls/minute	1000 calls/day

- Same 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 Enforce 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.

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.
- V5.0.3+ 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

For information about setting specific rate limits for operations, see step 16.

15. Optional: Specify whether your Plan requires subscription approval. If you want subscriptions by developers to require approval through the API Manager user interface, select Require subscription approval; otherwise, ensure the check box is cleared.
16. Optional: Add a rate limit to an operation.
    V5.0.0 ONLY V5.0.1 ONLY
    a. Hover the cursor over the API that contains the operation, and click the Show operations icon ⋯.
    b. Hover the cursor over the operation that you want to apply a rate limit to. Click the Edit rate limit icon The Edit rate limit icon
    .
    c. Ensure the Unlimited check box is cleared, and then specify the rate limit you want to apply.
    If the Enforce hard limit check box is selected the Plan will stop applications from calling the operation after reaching the rate limit.
    V5.0.2+
    a. Hover the cursor over the API that contains the operation, and click the Show operations icon ⌄ .
    b. Hover the cursor over the operation that you want to apply a rate limit to, and click Override rate limit.
    c. Specify the rate limit that you want to apply.
    If the Enforce hard limit check box is selected, the Plan will stop applications from calling the operation after reaching the rate limit.
    d. Optional: To set additional rate limits, click the Add new rate limit icon ⊕ for the operation, and then specify the rate limit that you want to apply. You can set multiple rate limits, at second, minute, hour, day, and week time intervals.
17. Click the Save icon 💾 to save your changes.

Results

You have created a 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 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

V5.0.8 +

Defining a Plan

Define Plans to specify the limitations and subscription details of how developers can use your API Products. A Plan includes rate limit settings that can be applied to the Plan as a whole, or specified for each operation in an API. It can also include billing information that applies to the Product. Through Products and Plans, you have greater control over what APIs your developers have access to V5.0.8 +, and the subscription terms.

Before you begin

Define your APIs. For more information, see Creating API definitions. You must have a Product that is already created, or you must be creating one while you complete these steps. Plans are part of a Product.
Anybody using the API Designer can perform all actions within it. However, when staging or publishing a Product, permissions will be enforced by API Connect. For more information about user roles and permissions, see Administering user access.

Note: The API Manager user interface (UI), also includes the ability to create and edit Products. However, the preferred method for these tasks is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.
Each Product must have at least one Plan, if you want to make it available for customers to subscribe to. The Plan contains details about the rate limit and billing information when a customer subscribes to the Product. Defining multiple Plans for the same Product gives subscribers flexibility of selecting a Plan with terms that best meets their needs.

Procedure

To create a Plan, complete the following steps:

1. To open the API Designer UI, enter the following command at the command line:

   ```
   apic edit
   ```

   The API Designer UI opens in your default browser.
   Note: If you are connected to a network, you are asked to sign in with IBM Cloud. Enter your IBMid credentials and click LOG IN. If this is your first time using API Designer, the Draft APIs information page opens, click Got it!, and continue with the following steps.
2. Select the Products tab in API Designer.
   The Products tab opens.
3. If the Product that you want to work with is already there, select it, and continue with step 5.
   Your Product design is displayed.
4. Optional: See Creating a Product, if you have not yet created the Product.
5. Select Plans in the navigation.
   A Default Plan is automatically created for you if you do not want to create your own, and any APIs that you selected for this Product are automatically included in this Plan. This Plan has no rate limitations, and no billing information. You can clear the check box for any API to exclude it from the Plan, but at least one API must be included. A Product cannot be staged if it includes any Plans that do not include APIs. If you decide not to use the Default Plan, you must delete it.
6. Select the Delete icon beside the Default Plan to remove it from your Product.
7. Select the Add Plan icon ⊕ in the Plans section to add a new Plan.
   A Plan called *New Plan 1* is created and displayed.
8. Select the name of the Plan to display its details.
9. Replace the title, name, and version with your information for the new Plan.
   The title is the name of the Plan that the customer sees and subscribes to.
   Note: The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), and hyphen characters (-). A hyphen cannot be used as the first or last character in the name.
10. Optional: If you want to bill for the Plan, configure it for billing.
    a. Select your billing plan.
    b. Enter the monthly cost of the Plan.
    c. Select the unit of currency for the billing.
    d. Set the number of free trial days that subscribers can use when they subscribe to the Plan.
11. Optional: Add a rate limit to your Plan.

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.
Note: If you want to configure notifications that are triggered when your Plan approaches its rate limit in the Developer Portal, you must assign one of the following names to the rate limit in the Plan:

- rate-limit
- rate-limit-1
- rate-limit-2
- per-minute
- V5.0.0 ONLY V5.0.1 ONLY Clear the Unlimited check box and then specify the rate limit you want to apply.
- V5.0.2 + Click the Add new rate limit icon ⊕, and then specify the rate limit that you want to apply. You can set multiple rate limits, at second, minute, hour, day, and week time intervals.
  DataPower Gateway only To add burst limits to your Plan, so that you can prevent usage spikes that might damage infrastructure, click the Add new burst limit icon ⊕, and then specify the burst limit that you want to apply. You can set multiple burst limits, at second and minute intervals. To remove a rate or burst limit, click the Delete rate limit or Delete burst limit 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 Enforce 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 Enforce 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."

Enforce hard limit affects only the **rate limit**, as illustrated by the following scenarios:

Scenario A

Table 1. Enforce Hard Limit Enabled

Enforce 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. Enforce Hard Limit Disabled

Enforce Hard Limit	Burst Limit	Rate Limit
OFF	100 calls/minute	1000 calls/day

- Same 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 Enforce 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.

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.
- V5.0.3+ 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

For information about setting specific rate limits for operations, see step 14.

12. Optional: Specify whether your Plan requires subscription approval. If you want subscriptions by developers to require approval through the API Manager user interface, select Require subscription approval; otherwise, ensure the check box is cleared.
13. Verify that the APIs you require are included in the Plan.
    a. Under APIs included, ensure that the check boxes of the APIs you require are selected. If there are APIs already selected, and you do not want them included in the Plan you are editing, clear their check boxes.

    Note: If you add the same API to more than one Plan, and the same Application (client ID) signs up for multiple Plans that contain the same API, the Gateway server cannot determine which Plan rate limit should be applied. If you anticipate that an API will be shared by an Application, a proxy API should be defined so that the correct Plan rate limit can be applied.
14. Optional: Add a rate limit to an operation.

    V5.0.0 ONLY V5.0.1 ONLY

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon •••.
    b. Hover the cursor over the operation that you want to apply a rate limit to. Click the Edit rate limit icon The Edit rate limit icon .
    c. Ensure the Unlimited check box is cleared, and then specify the rate limit you want to apply.
       If the Enforce hard limit check box is selected the Plan will stop applications from calling the operation after reaching the rate limit.

    V5.0.2 +

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon ⌄ .
    b. Hover the cursor over the operation that you want to apply a rate limit to, and click Override rate limit.
    c. Specify the rate limit that you want to apply.
       If the Enforce hard limit check box is selected, the Plan will stop applications from calling the operation after reaching the rate limit.
    d. Optional: To set additional rate limits, click the Add new rate limit icon ⊕ for the operation, and then specify the rate limit that you want to apply. You can set multiple rate limits, at second, minute, hour, day, and week time intervals.
15. Optional: Select which operations from an API to include in the Plan.

    V5.0.0 ONLY V5.0.1 ONLY

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon •••.
    b. Select or clear the check boxes for the operations you want to include or exclude.

    V5.0.2 +

    a. Hover the cursor over the API that contains the operation, and click the Show operations icon ⌄ .
    b. Select or clear the check boxes for the operations you want to include or exclude.
16. Click the Save icon 💾 to save your changes.
17. Define any additional Plans that you need for your Product.

Results

You have created a Plan, and specified a set of APIs that can be made available to your developers.

What to do next

After you have defined all of your Plans, finish creating your Product and publish your Product to a Catalog. For more information, see Staging a 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

Staging a Product

Stage a 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. V5.0.5 + 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 user interface (UI). For more information, see Creating and configuring Catalogs.
V5.0.5 + Note: All references in this topic to a Catalog can also be applied to a 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 Stage permission for the target Catalog. For information on configuring Product management permissions for a Catalog, see Creating and configuring Catalogs.

Note: The API Manager UI also includes the ability to stage Products. However, the preferred method for this task is by using the API Designer UI, as described here. Any steps that are specific to a particular UI are marked with an icon.

About this task

Before a Product can be published, you must first stage that Product to a Catalog. A Catalog is a staging target, and behaves as a logical partition of the DataPower® Gateway or Micro Gateway, and the Developer Portal.
If you stage a Product to a Catalog, editing and then restaging that Product through the Products tab of API Designer or API Manager will effect changes to the staged version.

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 a LoopBack application through the API Designer. You can also publish LoopBack applications at the same time as staging your Product by using the following instructions.
If an API has an OpenAPI (Swagger 2.0) definition file, and is part of a Product that is being staged to a Catalog, validation of the OpenAPI (Swagger 2.0) definition file occurs during the staging process. The following validation occurs:

- Validation against the OpenAPI (Swagger 2.0) 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 IBM 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 IBM API Connect Gateway, then a "host" must be provided

V5.0.5 + Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the product that contains the API is staged. For more information, see V5.0.5 + Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.

Procedure

To stage a Product, complete the following steps:

1. Select the Products tab in API Designer.
   The Products tab opens.
2. Click the Product that you want to work with.
   If you have more than one version of the Product, ensure that you click the version that you want to work with.
3. Click the Publish icon: Publish icon.
4. V5.0.3 and earlier Complete either of the following steps:
   - If the Catalog to which you want to stage the Product is shown in the list:

        - API Designer only Select the Catalog that you require, and then select Stage only, followed by Select specific products and select the Product that you want to stage. Click Publish. Your Product is staged.
        - API Manager only Select the Catalog that you require. Your Product is staged.
    - API Designer only If the Catalog to which you want to stage the Product is not shown in the list, select Add and Manage Targets.
        - Select Add IBM Cloud target or Add a different target, enter the connection details for the target management server that you require, and click Sign in.
        - Select the organization and Catalog to which you want to stage the Product to, and click Next.
        - Optional. If you have a LoopBack application that you want to publish, select the App to publish to.
        - Click Save.
        - Now you have added the target Catalog, click the Publish icon Publish icon again, and select the Catalog.
        - Select Stage only, followed by Select specific products and select the Product that you want to stage. Click Publish.

    Your Product is staged.
5. API Designer only V5.0.4 + Click Add and Manage Targets and then proceed as follows:
    a. Select Add IBM Cloud target or Add a different target.
    b. If required, enter the connection details for the target management server, and click Sign in.
    c. Select the organization and Catalog to which you want to stage the Product.
    d. V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.
    e. Click Save.
    f. Optional: If you have a LoopBack application that you want to publish, select the App to publish to.
    g. Click Save.
    h. Now you have added the target Catalog, click the Publish icon Publish icon again, and select the Catalog.
    i. Select Stage only, followed by Select specific products and select the Product that you want to stage. Click Publish.

    Your Product is staged.
6. API Manager only V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.

    Your Product is staged to the Space that you selected.

Results

Your Product is staged to a Catalog. To view the state of the Product in the Catalog, open the API Manager UI, select the Dashboard section in the navigation pane, and click on 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: If the Product contains an API that includes a user-defined policy that has not been imported into this Catalog, the staging will fail. The user-defined policy must be available in the required Catalog for the Product to be staged successfully. For detailed instructions on how to import a user-defined policy into a Catalog, see Importing a user-defined policy into a Catalog.

What to do next

- Test an API. For more information, see Testing an API with the API Designer test tool.
- Publish your Product to a community 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a new version of your Product

You can have multiple versions of a Product. 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 Product.

About this task

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. Select the Products tab in API Designer.
   The Products tab opens.
2. Click the Product you want to create your new version from.
   The Product details page is displayed.
3. Click the More Actions icon ⋮, then click Save as a new version.
4. Enter your new version number, and click Save as a new version.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating and validating API and Product definitions by using the command line interface

The developer toolkit 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.

Important: The developer toolkit command line interface is distinct from the IBM® API Connect Command Line Interface. For more information on the IBM API Connect Command Line Interface, see The Command Line Interface.
The following topics explain how to use the command line interface to create an OpenAPI (Swagger 2.0) definition file, create a Product definition file, and validate a YAML or JSON definition.

- **Creating an OpenAPI (Swagger 2.0) definition file**
  APIs are defined in OpenAPI (Swagger 2.0) definition files, in YAML format. You can create a default OpenAPI (Swagger 2.0) definition file by using the **create** command and then modify it by using an editor of your choice.
- V5.0.5+ **Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files**
  If you deploy an API to an IBM API Connect Management server by using the developer toolkit command line, you can use the `$ref` field in your OpenAPI (Swagger 2.0) YAML and JSON API definition files to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file. When IBM 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.
- V5.0.7+ **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

(Swagger 2.0) definitions.

- **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 or you can find and use the schemas that describe valid APIs and Products to validate your own API and Product definitions.
- V5.0.2 + **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.

Related concepts

- Developer toolkit tutorials

Related reference

- Toolkit command summary

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating an OpenAPI (Swagger 2.0) definition file

APIs are defined in OpenAPI (Swagger 2.0) definition files, in YAML format. You can create a default OpenAPI (Swagger 2.0) 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 publish` command to publish the Product. You can also create a draft API in API Manager by using the **push** command.

You can create an API in the CLI by running `apic create --type api`, and add additional arguments on the command line.

Another option is to create an API interactively in the command line by running `apic create --type api` and following the prompts.

You can see further details and available options for the `apic create --type api` command by running:

```
apic create --type api --help
```

IBM® provides an extension to the OpenAPI (Swagger 2.0) specification; this extension is described in IBM extensions to the OpenAPI (Swagger 2.0) specification.

V5.0.2 + 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 --type 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.
Instead of supplying the `--template` option, you can set the following configuration variables:

- `template-default-api` - specifies the base file name of the .hbs template file.
- `template-path` - specifies the directory containing the template file.

For more information, see Creating and using API and Product definitions templates.
An API template file must have a .hbs filename extension. You can create a template from scratch, or start with the example (default) API template provided in API and Product definition template examples.

V5.0.7 +
From IBM API Connect Version 5.0.7.2, you can create multilingual API and Product documentation by using an `x-ibm-languages` extension directly in the OpenAPI (Swagger 2.0) definition. For more information, see Using x-ibm-languages to create multilingual API and Product documentation.

- **IBM extensions to the OpenAPI (Swagger 2.0) specification**
  You use the `x-ibm-configuration` object in your OpenAPI (Swagger 2.0) definition file to add extensions that are specific to IBM API Connect.

Related concepts

- Developer toolkit tutorials

Related tasks

- Creating a Product definition file

Related reference

- Toolkit command summary

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

IBM extensions to the OpenAPI (Swagger 2.0) specification

You use the **x-ibm-configuration** object in your OpenAPI (Swagger 2.0) 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
  assembly:
    execute:
      - assembly_component
   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: • **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. • **realized**: the API is in the implementation phase.	String
testable	Yes	Use the **testable** extension to specify whether the API can be tested using the test tool in the Developer Portal.	Boolean
enforced	Yes	Use the **enforced** extension to specify if the API Connect gateway is used to enforce the API. • **true** indicates that the API Connect gateway is used to enforce the API. • **false** indicates that the API Connect gateway is not used to enforce the API.	Boolean
cors	Yes	Use the **cors** extension to specify whether CORS access control is used for the API. The extension has an **enabled** field which is a Boolean.	Object (with a single Boolean field)

Extension	Required	Description	Type
assembly	No	Use the **assembly** extension to describe the application of policies and logic to an API. It contains an **execute** field that contains an array of policies that are applied in order. It can contain a **catch** field that contains an array of error cases to be caught. For information about use of the **execute** field, see execute. For information about use of the **catch** field, see catch.	Object
gateway	No	Use the **gateway** extension to specify which type of gateway you want to use. If you are using a DataPower® Gateway or Micro Gateway, it must be included and take one of the following values: • datapower-gateway • micro-gateway	String
properties	No	Use the **properties** extension to define properties for use in an API.	Object (properties)
catalogs	No	Use the **catalogs** extension to define Catalog-specific values for properties defined in the **properties** extension.	Object (catalogs)

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 occurr 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

execute

The execute field of an assembly has the following structure:

```
execute:
 - Policy_1
 - Policy_2
```

The following table describes the possible policies and logic constructs that can be included in an execute field.
Important: If you are using IBM® API Connect for IBM Cloud, you must apply only policies that can be run on the DataPower® Gateway.

Table 1. execute properties

Property	Required	Description	Data Type
DataPower Gateway activity-log[1]	No	Use the activity-log policy to log information that relates to the calling of API operations.	object (activity-log)
DataPower Gateway gatewayscript	No	Include a GatewayScript program.	object (gatewayscript)
DataPower Gateway Micro Gateway if	No	Use the if policy to execute a section of the assembly only when a condition is fulfilled.	object (if)
DataPower Gateway Micro Gateway invoke	No	Use the invoke policy to call an API. V5.0.8+ The DataPower gateway might replace the last invoke policy in your assembly with a proxy policy to improve performance. To disable this, see: API properties.	object (invoke)
Micro Gateway javascript	No	Include a JavaScript program.	object (javascript)
DataPower Gateway json-to-xml	No	Convert payload from JSON to XML.	object (json-to-xml)
V5.0.1+ DataPower Gateway V5.0.1+ jwt-generate	V5.0.1+ No	V5.0.1+ Generate a JSON Web Token (JWT).	V5.0.1+ object (jwt-generate)
DataPower Gateway V5.0.1+ jwt-validate	No	Validate a JSON Web Token (JWT).	object (jwt-validate)
V5.0.5+ DataPower Gateway V5.0.5+ ltpa-generate	V5.0.5+ No	V5.0.5+ Use the Generate LTPA Token security policy in IBM API Connect to generate a Lightweight Third Party Authentication (LTPA) token.	V5.0.5+ object ltpa-generate (Generate LTPA Token)
DataPower Gateway map	No	Use the map policy to transform variables.	object (map)
DataPower Gateway Micro Gateway 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 (operation-switch)
DataPower Gateway proxy	No	Proxy a service.	object (proxy)
DataPower Gateway 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 (redact)
DataPower Gateway Micro Gateway 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 (set-variable)
DataPower Gateway Micro Gateway throw	No	Use the throw policy to specify points at which an error should be thrown.	object (throw)
DataPower Gateway Micro Gateway user-defined-policy	No	You can apply your own user-defined policies to your APIs.	object (User-defined policies)

Property	Required	Description	Data Type
DataPower Gateway validate	No	Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.	object (validate)
V5.0.0+ DataPower Gateway validate	V5.0.0+ No	V5.0.0+ Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema.	V5.0.0+ object (validate)
	V5.0.3+ No	V5.0.3+ Use the Validate policy to validate the payload in an assembly flow against a JSON or an XML schema. Micro Gateway You can now also use the validate policy with the Micro Gateway to validate the payload in an assembly flow against a JSON schema.	
DataPower Gateway V5.0.2+ validate-usernametoken	No	Validate a WS-Security UsernameToken.	V5.0.2+ object (validate-usernametoken)
DataPower Gateway xml-to-json	No	Convert payload from XML to JSON.	object (xml-to-json)
DataPower Gateway xslt	No	Apply an XSLT transform to the payload.	object (xslt)

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']
```

1
Note: The Micro Gateway does not support the Activity Log policy. However, the Micro Gateway does collect the basic analytic statistics. The statistics that the Micro Gateway gathers are equivalent to what an Activity Log policy in the DataPower Gateway with `Content:activity` settings gathers with some exceptions:

- For the following fields, the Micro Gateway does not collect the information and sends empty payload: `requestHttpHeaders`, `responseHttpHeaders`, and `debug`.
- When the Micro Gateway starts with an APIMANAGER environment variable that specifies a valid Management server, the Micro Gateway automatically collects the basic analytic statistics. There is no mechanism to turn the collection function on or off at runtime.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

activity-log

Use the Activity Log policy 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 activity-log policy has the following format:

```
- activity-log:
  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 (Swagger 2.0) definition file.

You can also apply an activity-log policy by using the API Designer assembly editor to add a built-in policy to the API. For more information, see Activity Log (activity-log) in the built-in policies section.

For more information on activity-log output with examples, see ../com.ibm.apic.apionprem.doc/rapim_analytics_apieventrecordfields.html

The following table describes the policy properties:

Table 1. Activity Log policy properties

Property label	Required	Description	Data type
Title	Yes	A title for the policy is required, but a default value, **`activity-log`** 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: • **`none`**: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal. • **`activity`**: Logs invocation only (only the resource URI is recorded). • **`header`**: Logs activity and header. • **`payload`**: Logs activity, header, and payload (the original request, if any, and the final response). The default value is **`activity`**.	string
Error content	No	Indicates what content to log if an error occurs. Valid values: • **`none`**: Indicates that no logging occurs. Restriction: This option disables notifications for application developers who use your Developer Portal. • **`activity`**: Logs invocation only (only the resource URI is recorded). • **`header`**: Logs activity and header. • **`payload`**: Logs activity, header, and payload (the original request, if any, and the final response). The default value is **`payload`**.	string

Example 1

```
# use defaults

- activity-log:
  title: default activity logging
```

Example 2

```
- activity-log:
  title: no logging for successful calls
  content: none
  error-content: activity
```

Related concepts

- Creating an OpenAPI (Swagger 2.0) definition file

Related reference

- IBM extensions to the OpenAPI (Swagger 2.0) specification
- execute

Related information

- API event record fields
- API Analytics

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

gatewayscript

Use the gatewayscript policy to execute a specified DataPower® GatewayScript program.

The gatewayscript policy has the following structure:

```
- gatewayscript:
   title: Title
   desription: Description
   source: Script
```

The following table describes the policy properties:

Table 1. gatewayscript policy properties

Property label	Required	Description	Data type
Title	No	The title of the policy. The default value is `gatewayscript`.	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:
   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 (gatewayscript) in the built-in policies section.

Related concepts

- Variable references in API Connect

Related reference

- GatewayScript code examples
- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

if

Use the if construct when you want to execute a portion of your assembly only when a specific condition is fulfilled.

The if policy has the following format:

```
- if:
  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.

Table 1. if policy properties

Property	Required	Description	Data type
title	No	A title for the policy.	string
description	No	A policy description.	string
condition	Yes	A script that returns **true** or **false**. Use GatewayScript for a DataPower® Gateway implementation or JavaScript for a Micro Gateway implementation. For information about the context variables you can use and how to reference them in your script, see API Connect context variables.	string
execute	Yes	The policy assembly that you want to execute if the condition returns **true**. For more information, see execute.	string

Example

```
# carry out different redaction actions depending on the operation

- if:
  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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

invoke

Use the invoke policy to call an API.

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 has the following format:

```
- invoke:
  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
  stop-on-error:
    - stop_on_error_type
```

V5.0.8 + From IBM® API Connect V5.0.8.0 and beyond, 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 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.

Note that a proxy policy ignores the Stop on error property, and that a replacement with a proxy does not occur if you have configured any catches on the invoke policy. For more information about the proxy policy, see proxy. 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.

The following table describes the properties of the invoke policy.

Table 1. Invoke policy properties

Property	Required	Description	Data type
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: - Keep - GET - POST - PUT - DELETE - PATCH - HEAD - OPTIONS The default value is **`GET`**. However, if set to **`Keep`**, or 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 **`60`**.	integer
compression	No	Specifies whether data is to be compressed by using gzip before it is uploaded. The default value is **`false`**.	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 *message*. Use this property to specify an alternate location to store the invoke response. This variable can then be referenced in other actions, such as map. Note: If you want the response to be saved in *message*, leave the output property blank, do **not** supply the value `message`.	string
DataPower Gateway only cache-key	No	Specifies the unique identifier of the document cache entry.	string
DataPower Gateway only cache-response	No	The cache response type. Valid values: - protocol: The cache behavior is defined by the Cache-Control headers on the request and response. - no-cache: Specifies that there is no caching. However, if the document is already in the cache, the document is retrieved from the cache. - time-to-live: Specifies that the response stays in the cache for the specified time. The default value is **`protocol`**.	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 **`false`**.	boolean
DataPower Gateway only 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 **`time-to-live`**. Enter a value in the range 5 - 31708800. The default value is **`900`**.	integer

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 `catch` 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.	Boolean

Example

```
- invoke:
  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
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Micro Gateway only

javascript

The javascript policy has the following structure:

```
- javascript:
   title: Title
   desription: Description
   source: Script
```

The following table describes the policy properties:

Table 1. javascript policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `javascript`.	string
Description	No	A description of the policy.	string
Source	Yes	The JavaScript source code to execute.	string

Example

The following is an example of a simple javascript policy:

```
- javascript:
   title: Example_JavaScript
   source: console.debug('Hello World!');
   description: A simple JavaScript policy.
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

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.

The json-to-xml policy has the following structure:

```
- json-to-xml:
   title: Title
   description: Description
```

The following table describes the policy properties:

Table 1. Policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `json-to-xml`.	string
Description	No	A description of the policy.	string

Example

The following is an example of a json-to-xml policy:

```
- json-to-xml:
   title: JSON to XML transform
   description: Transforms JSON message body to XML format
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.1+

jwt-generate

Use the Generate JWT security policy in IBM® API Connect to generate a JSON Web Token (JWT).

The jwt-generate policy has the following structure:

```
- jwt-generate:
  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
```

The following table describes the policy properties:

Table 1. Generate JWT policy properties

Property label	Property name	Required	Description	Data type
Title	title	No	The title of the policy. The default value is `jwt-generate`.	string
Description	description	No	A description of the policy.	string

Property label	Property name	Required	Description	Data type
JSON Web Token (JWT)	jwt	No	Runtime variable in which to place the JWT that is generated. The default value is: **`generated.jwt`**. However, if not set, the JWT that is generated is written to the Authorization Header as a Bearer token.	string
JWT ID Claim	jti-claim	No	Indicates whether a JWT ID (jti) claim should be added to the JWT. If selected, the property is set to **`true`**, and a UUID is generated and set as the JTI claim value.	boolean
Issuer Claim	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: **`iss.claim`**	string
Subject Claim	sub-claim	No	Runtime variable from which the Subject (sub) claim string can be retrieved.	string
Audience Claim	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
Validity Period	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 **`3600`**.	integer
Private Claims	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	jws-jwk	No	Runtime variable that contains the JWK that is used to sign the JWT.[1]	string
Cryptographic Algorithm	jws-alg	No	The cryptographic algorithm to use. Valid values are: • HS256 • HS384 • HS512 • RS256 • RS384 • RS512 • ES256 • ES384 • ES512 • V5.0.8 + PS256 • V5.0.8 + PS384 • V5.0.8 + PS512 Note: The following algorithms are not available in the drop-down list and must be added to the OpenAPI source manually: • PS256 • PS384 • PS512 For example: **`- jwt-generate:`** **`    title: jwt-generate`** **`    iss-claim: iss.claim`** **`    exp-claim: 3600`** **`    version: 1.0.0`** **`    jws-alg: PS256`** For more information on specifying the OpenAPI source for the Generate JWT policy, see jwt-generate.	string
Sign Crypto Object	jws-crypto	No	The cryptographic object to use to sign the JWT.[1]	string
Encryption Algorithm	jwe-enc	No	The encryption algorithm to use. Valid values are: • A128CBC-HS256 • A192CBC-HS384 • A256CBC-HS512	string

Property label	Property name	Required	Description	Data type
Encrypt JWK variable name	jwe-jwk	No	Runtime variable that contains the JWK to use to encrypt the JWT.	string
Key Encryption Algorithm	jwe-alg	No	The key encryption algorithm to use. Valid values are: • RSA1_5 • RSA-OAEP • RSA-OAEP-256 • dir • A128KW • A192KW • A256KW	string
Encrypt Crypto Object	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:
    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 (jwt-generate) in the built-in policies section.

[1] 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.1+

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.

The jwt-validate policy has the following structure:

```
- jwt-validate:
  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
```

The following table describes the policy properties:

Table 1. Validate JWT policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is `jwt-validate`.	string
Description	description	No	A description of the policy.	string
JSON Web Token (JWT)	jwt	Yes	Context or runtime variable that contains the JWT to be validated. The default value is: `request.headers.authorization`. However, if this property is not set, the policy looks for the JWT in the `request.headers.authorization` location by default. Note: The format of the authorization header must be: `"Authorization: Bearer jwt-token"` where *jwt-token* is the encoded JWT.	string
Output Claims	output-claims	Yes	Runtime variable to which the full set of claims that are contained in the JWT is assigned. The default value is: `decoded.claims`.	string
Issuer Claim	iss-claim	No	The Pearl Compatible Regular Expression (PCRE) to use to validate the Issuer (iss) claim.	string
Audience Claim	aud-claim	No	The PCRE to use to validate the Audience (aud) claim.	string
Decrypt Crypto Object	jwe-crypto	No	The cryptographic object (a shared key or certificate) to use to decode the claim.[1]	string
Decrypt Crypto JWK variable name	jwe-jwk	No	Runtime variable that contains the JWK to use to decrypt the JWT.[1]	string
Verify Crypto Object	jws-crypto	No	The cryptographic object (a shared key or certificate) to use to verify the signature.[2]	string
Verify Crypto JWK variable name	jws-jwk	No	Runtime variable that contains the JWK to use to verify the signature.[2]	string

Example

The following is an example of a jwt-validate policy:

```
- jwt-validate:
    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
```

```
      - jwt-validate:
          title: validate_jwt
          jwt: jwt
          output-claims: decoded.claims
          version: 1.0.0
          iss-claim: apic
          jwe-jwk: hs256-enc-key
          jws-jwk: hs256-key
```

For more information about how to use a jwt-validate security policy, see Validate JWT (jwt-validate) in the built-in policies section.

[1] 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.

[2] A JWK 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.5 +

ltpa-generate (Generate LTPA Token)

Use the Generate LTPA Token security policy in IBM® API Connect to generate a Lightweight Third Party Authentication (LTPA) token.

Restriction:

- The ltpa-generate policy is deprecated and is not supported in API Connect releases later than Version 5.
- The ltpa-generate policy can be used only with the DataPower® Gateway.

The ltpa-generate policy has the following structure:

```
- ltpa-generate:
  title: title
  description: description
  tokenVersion: version_of_the_LTPA_token
  tokenOutput: output_location_for_the_token
  tokenExpiry: token_expiry_period
  key: name_of_the_LTPA_key
  authenticatedUserName: runtime_variable_containing_the_authenticated_user_name
```

The following table describes the policy properties:

Table 1. Generate LTPA Token policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is `ltpa-generate`.	string
Description	description	No	A description of the policy.	string
LTPA Key	key	Yes	The name of the LTPA key that you want to use to generate the LTPA token. API Designer only Enter the name of the LTPA key by using one of the following syntax options: • *`my-ltpa-key`* - entering the name with no version number means that at run time the policy selects version 1.0.0 of the LTPA key. • *`my-ltpa-key:2.0.0`* - entering the name with a version number means that at run time that specific version of the key is used. • *`my-ltpa-key:latest`* - entering the name with `latest` means that at run time the policy selects the latest version of the LTPA key to use. API Manager only Select the LTPA key from the drop-down menu. Each LTPA key has a non-version specific option, for example my-ltpa-key, latest version. Select this option if you want the policy at run time to select the latest version of the LTPA key to use. Otherwise, select a specific version. Note: The automatic version selection feature relies on the LTPA key being configured with a version number that conforms to the version.release.modification version numbering scheme.	LTPAKey
Authenticated User Name	authenticatedUserName	Yes	The runtime variable that contains the authenticated user name. The LTPA token is generated with this property as the user. For example, if the API is configured with a basic authentication security definition, then the authenticated user name can be specified as `$(client.app.id)`. If the API is configured with an OAuth security definition, then the authenticated user name can be specified as `$(oauth.resource-owner)`. Alternatively, before the Generate LTPA Token policy, you can configure a set-variable policy to set a runtime variable with a particular user name, and then specify this runtime variable as the authenticated user name.	string

Property label	Property name	Required	Description	Data type
Token Version	tokenVersion	Yes	The version of the LTPA token. Select from the following values: • WebSphereVersion1 • WebSphereVersion1-FIPS • WebSphereVersion2 • WebSphere70Version2 The default value is **`WebSphereVersion2`**.	string
Token Output	tokenOutput	Yes	Define where in the output source the policy should place the generated LTPA token. Select from the following options: • In Cookie Header • In WSSec Header[1] The default option is In Cookie Header. Note: In WSSec Header should be selected only if the message has an XML or SOAP media type.	enum
Token Expiry	tokenExpiry	Yes	The length of time (in seconds) that is added to the current date and time, in which the LTPA key is considered valid. The default value is **`600`**.	integer

Example

```
- ltpa-generate:
    title: ltpa-generate
    tokenVersion: WebSphereVersion2
    tokenOutput: in-cookie-header
    tokenExpiry: 600
    key: 'my-first-ltpa-key:1.0.0'
```

Errors

The following error can be thrown while the policy is being executed:

- **`LTPAGenerateError`** - an error that captures all the errors that occur during the execution of the policy. Upon failure, the detailed error message is assigned to the runtime variable ltpa-generate.error-message, so it can be retrieved via catch.

If a catch is not configured, in the case of a failure the Generate LTPA Token policy returns an HTTP code 500 failure. The detailed error message can be found in the system log.
Tip: If there is an error, make the following checks:

- Verify that the IBM DataPower firmware is at Version 7.5.1 or later.
- Check that the password that is set in the LTPA key is correct.
- If In WSSec Header is selected for the Token Output property, verify that the payload of the message contains an XML or SOAP media type.

For more information about how to use an ltpa-generate security policy, see Generate LTPA Token (ltpa-generate) in the built-in policies section.

Related information

- LTPA
- LTPA versions and token formats
- Single sign-on for authentication using LTPA cookies

[1] If the In WSSec Header option is selected, the following conditions apply:

- If the input is XML, the policy creates a SOAP envelope, and places the LTPA token in the SOAP security header, and places the input XML in the SOAP body.
- If the input is SOAP, but without a SOAP security header, the policy creates a SOAP security header and places the LTPA token in this header. The rest of the SOAP message is untouched.
- If the input is SOAP and there is already a binary security token in the SOAP security header, the policy overwrites the existing token with the newly generated LTPA token. The rest of the SOAP message is untouched.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

map

Use the map policy to transform your assembly flow and specify relationships between variables.

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

   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:
            includeEmptyXMLElements: boolean
            namespaceInheritance: boolean
            inlineNamespaces: boolean
```

Table 1.

Property	Required	Description	Data type	Belongs to
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*	A reference to the definition of the type of the variable.	String	inputs or outputs
type	Yes*	The type of the variable.	String	inputs or outputs

Property	Required	Description	Data type	Belongs to
content	No	The content type of the variable: `application/xml` or `application/json`. If None 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**	Specifies by name the output variable that is to be set to a value by the action.	String	actions
create	Yes**	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	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.	String	actions
foreach	No	Specifies by name an array for which further actions should be performed for each element.	String	actions
V5.0.4 + options	No	Advanced XML options to control empty elements, inherit namespaces, and define namespaces inline.	Boolean	outputs
V5.0.8 + messagesInputData (From API Connect Version 5.0.8.7)	V5.0.8 + No	V5.0.8 + This property defines the severity level for log messages that relate to input data. Specify one of the following values: • `error` • `warn` • `info`	V5.0.8 + String	V5.0.8 + options

* There must be one of $ref or type in the description of a variable.

** There must be one of set or create in an actions field.

Example

```
      - map:
          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: true
              namespaceInheritance: true
              inlineNamespaces: true
```

operation-switch

Use the operation-switch construct when you want to execute alternative policy assemblies, conditional on the operation that is being called.

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 (Swagger 2.0) document.

The operation-switch policy has the following format:

```
- operation-switch:
 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.

Table 1. operation-switch policy properties

Property	Required	Description	Data type	
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: • **GET** • **POST** • **PUT** • **DELETE** • **HEAD** • **PATCH** • **OPTIONS**	string	operations
path	No	A relative path to an individual endpoint. For example, **/account_status**.	string	operations
operationID	No	The operation is defined in OpenAPI (Swagger 2.0). The policy operation refers to the OpenAPI (Swagger 2.0) 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 execute.	string	case

Example

Example that defines match operations with **verb/path** combinations:

```
# carry out different redaction actions depending on the operation

- operation-switch:
  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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

proxy

Apply the proxy policy to proxy another API within your operation, particularly if you need to call a large payload.

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:
 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
```

The table describes the properties of the proxy policy.

Table 1. Proxy policy properties

Property	Required	Description	Data type
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

Property	Required	Description	Data type
verb	No	The operation method type. Valid values: • Keep • GET • POST • PUT • DELETE • PATCH • HEAD • OPTIONS The default value is **`Keep`**.	string
http-version	No	The HTTP version. The default value is `1.1`.	string
timeout	No	The timeout value in seconds. The default value is `60`.	integer
compression	No	Specifies whether data is to be compressed by using gzip before it is uploaded. The default value is **`false`**.	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 **`context.message`**. The assembly developers can specify an additional location to store the proxy result with the output property.	string
DataPower Gateway only cache-key	No	Specifies the unique identifier of the document cache entry.	string
DataPower Gateway only cache-response	No	The cache response type. Valid values: • protocol: The cache behavior is defined by the Cache-Control headers on the request and response. • no-cache: Specifies that there is no caching. However, if the document is already in the cache, the document is retrieved from the cache. • time-to-live: Specifies that the response stays in the cache for the specified time. The default value is **`protocol`**.	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 **`false`**.	boolean
DataPower Gateway only 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 **`time-to-live`**. Enter a value in the range 5 - 31708800. The default value is **`900`**.	integer

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

redact

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.

The redaction policy has the following format:

```
- redact:
  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.

The following table describes the policy properties:

Table 1. redact policy properties

Property	Required	Description	Data type
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: • **remove**: Completely removes the specified fields. • **redact**: Redacts (obfuscates with "*"s) the fields to block out the data. The default value is **redact**. Note: If a numerical value is being redacted, the redacted value is depicted as ********** and the type is changed to **string**.	string
from	No	Determines where the redaction is to be applied. Valid values: • **all**: Apply the redaction to the request body, the response body, and the activity logs. • **request**: Apply the redaction to the request body only. • **response**: Apply the redaction to the response body only. • **logs**: Apply the redaction to the activity logs only. You can supply one or more values. The default value is **all**.	Boolean
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 Constructing XPath expressions to redact fields.	string

Example

```
# Specify separate remove and redact actions

- redact:
  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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

set-variable

Use the set-variable policy to set a runtime variable to a string value, or to add or clear a runtime variable.

The set-variable policy has the following format:

```
- set-variable:
  title: title
  description: description
  actions:
    - action_type: variable_name
      value: value
```

Table 1. set-variable policy properties

Property	Required	Description	Data type
title	No	A title for the policy.	string
description	No	A policy description.	string
action_type	Yes	Defines what action to apply on a runtime variable. Valid values: • `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. • `add`: Indicates that you want to add a runtime variable. Can be used to set new headers or to append a new entry of the same header name. • `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.	string
variable_name	Yes	Specifies the name of the variable that you want to set to a string value, or that you want to add or clear.	string
value	Yes*	Allocates this value to the specified variable. Can be a literal value, or another variable. * variable_value is required only when `set` or `add` is specified as the action.	string

Example 1

```
# clear a variable

set-variable:
  title: clear_region
  actions:
    - clear: message.headers.region
```

Example 2

```
# set a variable to the value of an API Gateway context variable

set-variable:
  title: set content type
  actions:
    - set: message.headers.contenttype
      value: $(message.headers.content-type)
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

The switch policy has the following format:

```
   - switch:
     title: switch
     description: 'Description'
     case:
      - condition: Script_1
        execute:
         Assembly_Section_1
      - condition: "((request.verb==='GET')&&(api.operation.path==='/path-1'))||
((request.verb==='POST')&&(api.operation.path==='/path-2'))"
        execute:
         Assembly_Section_2
      - otherwise:
         Assembly_Section_2
```

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 switch policies. For more information, see execute.

To use a condition field to switch 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.

Table 1. switch policy properties

Property	Required	Description	Data type
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 **true** or **false**. Use GatewayScript for a DataPower® Gateway implementation or JavaScript for a Micro Gateway implementation. For information about the context variables you can use and how to reference them in your script, see API Connect context variables.	string
execute	Yes (one per condition)	The policy assembly that you want to execute if the condition returns **true**. For more information, see execute.	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 execute.	string

Example

```
   - switch:
     title: switch
     case:
      - condition: 4 == 5
        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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

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.

Table 1. operation-switch policy properties

Property	Required	Description	Data type
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:
          title: throw
          name: '404'
          message: Not found
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

User-defined policies

You can apply your own user-defined policies to your APIs.

The policy configuration that you add to the OpenAPI (Swagger 2.0) definition file must conform to the schema that you defined in the YAML file that describes the policy, and will have the following general format:

```
policy_name:
  property1: value1
  property2: value2
      .
      .
      .
```

For details on user-defined policies, and how to describe and implement them, see Authoring policies.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

validate

Use the validate policy to validate the payload in an assembly flow against a schema.

Restriction:

- The schema that represents the XML can reference only one XML namespace.
- The schema cannot reference polymorphic XML elements.
- The OpenAPI `discriminator` field is **not** supported by the validate policy.
- The validate policy can be used only with the DataPower® Gateway, not with the Micro Gateway.

The validate policy has the following format:

```
- validate:
  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 (Swagger 2.0) definition file.

The following table describes the policy properties:

Table 1. validate policy properties

Property	Required	Description	Data type
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: • `request`: 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 Configuring an operation. • `response`: 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 Configuring an operation. • The name of a schema definition, in the following format: `#/definitions/schema_name` The schema must be defined in the `definitions:` section of your OpenAPI (Swagger 2.0) file.	string

You can also apply an validate policy by using the API Designer assembly editor to add a built-in policy to the API. For more information, see Validate (validate) in the built-in policies section.

Example 1

```
validate:
  title: validate the response
  definition: #/definitions/RouteOutput
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.2 +

validate-usernametoken

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.

The validate-usernametoken policy has the following format:

```
- validate-usernametoken:
    title: title
    description: description
    auth-type: Authentication URL_or_LDAP Registry
    auth-url: authentication_url_to_use
    tls-profile: tls_profile_to_use
    ldap-registry: name_of_the_ldap_user_registry
    ldap-search-attribute: name_of_the_ldap_user_password_attribute
```

The following table describes the policy properties:

Table 1. Validate Username Token policy properties

Property label	Property name	Required	Description	Data type
Title	title	Yes	The title of the policy. The default value is `validate-usernametoken`.	string
Description	description	No	A description of the policy.	string
Authentication type	auth-type	Yes	The authentication type to use to validate the UsernameToken. Valid values: • `Authentication URL`: Select this value to validate the user credentials against an authentication URL. • `LDAP registry`: Select this value to validate the user credentials against an LDAP user registry. The default value is: `Authentication URL`.	string
Authentication URL	auth-url	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 `Authentication URL`.	string
TLS profile	tls-profile	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 `Authentication URL`.	string
LDAP registry name	ldap-registry	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 `LDAP registry`.	string
LDAP search attribute[1]	ldap-search-attribute	Yes	The name of the LDAP user password attribute. Note: This property is required only if Authentication type is set to `LDAP registry`.	string

Examples

The following example shows an LDAP user registry authentication:

```
- validate-usernametoken:
    title: "validate-usernametoken"
    auth-type: "LDAP Registry"
    ldap-registry: "wstest"
    ldap-search-attribute: "userPassword"
```

The following example shows an Authentication URL definition:

```
- validate-usernametoken:
    title: "validate-usernametoken"
    auth-type: "Authentication URL"
    auth-url: "https://www.google.com"
    tls-profile: "default-ssl-profile"
```

For more information about how to use a validate-usernametoken security policy, see Validate Username Token (validate-usernametoken) in the built-in policies section.

[1] 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

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).

The xml-to-json policy has the following structure:

```
- xml-to-json:
   title: Title
   description: Description
```

The following table describes the policy properties:

Table 1. Policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `xml-to-json`.	string
Description	No	A description of the policy.	string

Example

The following is an example of a xml-to-json policy:

```
- xml-to-json:
   title: XML to JSON transform
   description: Transforms XML message body to JSON format
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

xslt

Use the xslt policy to apply an XSLT transform to the payload of the API definition.

The xslt policy has the following structure:

```
- xslt:
   title: Title
   description: Description
   input: Input_True_False
   source: Transform
```

The following table describes the policy properties:

Table 1. xslt policy properties

Property label	Required	Description	Data type
Title	Yes	The title of the policy. The default value is `xslt`.	string
Description	No	A description of the policy.	string

Property label	Required	Description	Data type
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
Source	Yes	The XSLT transform source to execute.	string

You can also apply an xslt policy by using the API Designer assembly editor to add a built-in policy to the API. For more information, see XSLT (xslt) in the built-in policies section.

For examples of the OpenAPI (Swagger 2.0) 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 in the user-defined policies section.

Related concepts

- Variable references in API Connect

Related reference

- XSLT (xslt)
- API Connect context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

catch

Use the catch extension to catch errors that occurr 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
`errors`	Yes	The errors for which the catch will activate. For a list of errors, see Error cases supported by assembly catches.	Array (String)
`execute`	Yes	The section of the assembly that will execute when the catch is activated.	Array (Object)
`default`	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 `execute`	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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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
property	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
value	No	The default value that the property takes. It can be empty.	String
description	No	The description of the property. It can be empty.	String
encoded	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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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
`catalog_name`	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
`properties`	Yes	This field contains any properties that are to be set.	Object
`property_name`	Yes	The field name is the property to be set and must match a property defined in the `properties` extension. It contains the value of the property for the catalog to which the field belongs. If, in the `properties` extension, it has been specified as encoded, this will be encoded when saved or staged by API Connect.	String

The following is an example of a `catalogs` extension:

```
  catalogs:
    Sandbox:
      properties:
        ID: 5678
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files

If you deploy an API to an IBM® API Connect Management server by using the developer toolkit command line, you can use the **`$ref`** field in your OpenAPI (Swagger 2.0) YAML and JSON API definition files to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file. When IBM API Connect processes the source API definition file, the **`$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:

- Push a draft API to an IBM API Connect Management server by using the **apic drafts:push** command. For more information, see Working with Drafts.
- Stage or publish an API to an IBM 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 (Swagger 2.0) file.

Example

A source YAML file contains the following OpenAPI (Swagger 2.0) 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 (Swagger 2.0) 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 IBM API Connect processes the source YAML file, the **`$ref`** field is replaced with the target code fragment, yielding the following OpenAPI (Swagger 2.0) 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 --type 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.
You can create a Product in the CLI by running **`apic create --type product`** and adding additional arguments on the command line..
Another option is to create a Product interactively in the command line by running **`apic create --type product`** and following the prompts.

You can see further details and available options for the **`apic create --type product`** command by entering the command:

```
apic create --type product --help
```

V5.0.2 + You can also create a Product from a custom Handlebars template by using the following command:

```
apic create --type 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.
V5.0.2 + Instead of supplying the `--template` option, you can set the following configuration variables:

- `template-default-product` - specifies the base file name of the .hbs template file.
- `template-path` - specifies the directory containing the template file.

V5.0.2 + For more information, see Creating and using API and Product definitions templates.
A Product definition file contains the following sections:

- A 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`
   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.

V5.0.7 + From IBM API Connect Version 5.0.7.2, you can create multilingual API and Product documentation by using an `x-ibm-languages` extension directly in the OpenAPI (Swagger 2.0) definition. For more information, see Using x-ibm-languages to create multilingual API and Product documentation.

Related concepts

- Working with Products in the API Designer

Related information

- Changing the availability of a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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:
     name: Short_Name
     title: Product_Title
     version: Product_Version
     description: Product_Description
   ```

   where:
   - *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.
   - *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.
   - *Product_Description* is a short description of the policy. Any string can be used.
   - *version* is the version number of the policy.
     Tip: The **version.release.modification** version numbering scheme is recommended, for example 1.0.0.
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:
  name: Short_Name
  title: Product_Title
  version: Product_Version
  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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

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
         orgs: View_Orgainzations
         tags:
            - View_Tag_1
            - View_Tag_2
   ```

   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.
   - *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.
   - 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:`.
3. Under visibility, specify who can subscribe to your Product's Plans; use the following syntax:

   ```
      subscribe:
         enabled: Subscribe_Toggle
         type: Subscribe_Audience
         orgs: Subscribe_Organizations
         tags:
            - Subscribe_Tag_1
            - Subscribe_Tag_2
   ```

   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.
- *Subscribe_Organisations* 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`.
- 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.

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
      orgs:
         - Organization_1
         - Organization_2
      tags:
         - View_Tag_1
         - View_Tag_2
   subscribe:
      enabled: Subscribe_Toggle
      type: Subscribe_Audience
      orgs:
         - Organization_1
         - Organization_2
      tags:
         - Subscribe_Tag_1
         - Subscribe_Tag_2
```

where all variables are as described in previously in this topic. The indentation must be as in the example.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Referencing the APIs for your Product

Detail the file paths for the APIs you want to include in your Product.

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; use the following syntax:

```
apis:
   - $ref 'file:API_1_File_Path'
   - $ref 'file:API_2_File_Path'
```

where the *API_File_Path* variables are the file paths for the YAML files containing OpenAPI (Swagger 2.0) representations of your APIs.

Alternatively, you can reference your APIs by name and version; to do so, use the following syntax:

```
apis:
   - $ref API_1_Name:API_1_Version
   - $ref API_2_Name:API_2_Version
```

where

- the *API_Name* variables are the case-sensitive names of your APIs.
- the *API_Version* variables are the versions of your APIs.

For more information about creating OpenAPI (Swagger 2.0) definitions, see Creating an OpenAPI (Swagger 2.0) 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.
DataPower Gateway only V5.0.2 + 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:

```
      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.
   - V5.0.3+ 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.
   To reference by file path, use the following syntax:

```
      apis:
         - $ref: 'file:API_File_Path'
```

   where *API_File_Path* is the file path of the YAML file for an API that you have already included in your Product. The API must have been added to the Product before it can be included in a Plan, as is described in Referencing the APIs for your Product.

   To reference by name and version, use the following syntax:

```
apis:
   - $ref API_1_Name:API_1_Version
   - $ref API_2_Name:API_2_Version
```

   where
   - the *API_Name* variables are the case-sensitive names of your APIs.
   - the *API_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:
         - $ref: 'file:File_Path'
            operations:
               - path: 'Operation_1_Path'
                 verb: 'Operation_1_Verb'
               - path: 'Operation_2_Path'
                 verb: 'Operation_2_Verb'
```

   where
   - the *Operation_Path* variables are the paths of operations that is to be included. The dash is required for each new operation.
   - the *Operation_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:
         - $ref: 'file:API_File_Path'
            - path: 'Operation_Path'
              verb: 'Operation_Verb
              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 examples.

If you have referenced APIs using their names and versions, and enforced a rate limit for your Plan and not specified individual operations, your Plans section should have the following form:

```
plans:
   Plan_Name:
      description: Plan_Description
      approval: Approval_Toggle
      rate-limits:
         Name:
            value: Limit
            hard-limit: Limit_Toggle
      apis:
         - $ref: 'API_Name:API_Version'
         - $ref: API_Name:API_Version
```

If you have referenced your APIs using their file path, and enforced rate limits at the Plan level and a separate one for one of two operations, your Plans section should have the following form:

```
plans:
   Plan_Name:
      description: Plan_Description
      approval: Approval_Toggle
      rate-limits:
         Name:
            value: API_Limit
            hard-limit: API_Limit_Toggle
      burst-limits:
         Name:
            value: Burst_limit
      apis:
         - $ref: 'file:API_File_Path'
           operations:
               - path: Operation_1_Path
                 rate-limits:
                    Name:
                       value: Operation_Limit
                       hard-limit: Operation_Limit_Toggle
              - path: Operation_2_Path
```

In both examples, variables are as in the previous steps and the indentation must be as presented.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

An example YAML representation of a Product

Products can be represented by using a YAML file in a similar fashion to how APIs can be represented by using OpenAPI (Swagger 2.0).

The following code describes a complete Product, with sample values.

V5.0.0 ONLY V5.0.1 ONLY

```
product: 1.0.0

info:
  name: example_product
  title: Example Product
  version: 1.0.0
  description: This is an example Product.
  contact:
    name: Eva Smith
    url: 'http://example.com/contact'
    email: esmith@example.com
  license:
    name: Example License
    url: 'http://example.com/license'
  termsOfService: This service is an example only

visibility:
   view:
      enabled: true
      type: public
   subscribe:
      enabled: true
      type: authenticated

apis:
   - $ref: "file:///example_api.yaml"

plans:
  Example-Plan:
    description: This is an example Plan.
    approval: true
    rate-limit:
      value: 40/5second
      hard-limit: true
    apis:
      - $ref: "file:///example_api.yaml"
        operations:
        - path: '/example_operation_1'
          verb: 'GET'
          rate-limit:
            value: 100
            hard-limit: true

        - path: '/example_operation_2'
           verb: 'GET'
```

V5.0.2 +

```
product: 1.0.0

info:
  name: example_product
  title: Example Product
  version: 1.0.0
  description: This is an example Product.
  contact:
    name: Eva Smith
    url: 'http://example.com/contact'
    email: esmith@example.com
  license:
    name: Example License
    url: 'http://example.com/license'
  termsOfService: This service is an example only

visibility:
   view:
      enabled: true
      type: public
   subscribe:
      enabled: true
      type: authenticated

apis:
   - $ref: "file:///example_api.yaml"

plans:
  Example-Plan:
    description: This is an example Plan.
    approval: true
    rate-limits:
```

```
   per-second:
    value: 40/5second
    hard-limit: true
   per-minute:
    value: 500/1minute
    hard-limit: false
  burst-limits:
   burst-limit-1:
    value: 10000/1day
  apis:
   - $ref: "file:///example_api.yaml"
    operations:
    - path: '/example_operation_1'
     verb: 'GET'
     rate-limits:
      unlimited:
       value: unlimited

    - path: '/example_operation_2'
      verb: 'GET'
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

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 (Swagger 2.0) definitions.

From IBM® API Connect Version 5.0.7.2, 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 (Swagger 2.0) 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 (Swagger 2.0) 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 (Swagger 2.0) YAML file with French as the default language, see the Examples section.

Examples

An example API OpenAPI (Swagger 2.0) 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 (Swagger 2.0) 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 (Swagger 2.0) 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 or you can find and use the schemas that describe valid APIs and Products to validate your own API and Product definitions.

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.

About this task

Procedure

Use one of the following methods to validate your API definition:

- To perform validation 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 (Swagger 2.0) section of the API and none of its extensions.

  V5.0.5+ Note: If the OpenAPI (Swagger 2.0) file that defines your API uses a `$ref` field to reference a fragment of OpenAPI (Swagger 2.0) code that is defined in a separate file, the `$ref` field is replaced with the contents of the target file before the API definition is validated by the **apic validate** command. For more information, see V5.0.5+ Using $ref to reuse code fragments in your OpenAPI (Swagger 2.0) files.
- To locate the JSON schemas that you can use to validate your API or Product, complete the following steps:
  1. Navigate to the *npm_install_folder*/apiconnect/node_modules/apiconnect-validate/schemas folder, where *npm_install_folder* is your npm global installation folder; for example:

     ```
     /usr/local/lib/node_modules
     ```

Note: If you are using a search to find the folder, be aware that there is a different node_modules folder inside *npm_install_folder*/apim.toolkit.

2. Select the schema that you want to use.
   - To validate a Product, use the product-schema.json file.
   - To validate the main body of an API, use the swagger-v2.0-schema.json file.
   - To validate the IBM API Connect extension of an API, use the x-ibm-configuration.json file.
3. Validate your API definition by using your preferred method.

Related concepts

- Creating an OpenAPI (Swagger 2.0) definition file

Related tasks

- Creating a Product definition file

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

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 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 --type [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. Alternatively, you can set the **template-path** configuration variable to the location of the directory containing the template.

   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.

   In addition to the **--template** option, you can set default values for working with templates by using these configuration variables:
   - **template-default-product** - Base file name of the product template (.hbs) file.
   - **template-default-api** - Base file name of the API template (.hbs) file.
   - **template-path** - List of directories in which to search for templates. It may specify one or more directories (absolute paths). Separate multiple directories by a space and enclose in quotes. Note you can use this configuration variable along with the **--template** option.

   As with all configuration variables, you can set these values locally (for a single application project) or globally (for all projects). For more information, see Toolkit command summary.

Related reference

- Template variables for API and Product definitions
- API and Product definition template examples

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Developer toolkit tutorials

Tutorials for using the developer toolkit. Ensure you select the correct tutorials for the version of IBM® API Connect that you are using.

Use the following links to select the correct tutorials for your version of IBM API Connect:

- V5.0.7+ **Developer toolkit tutorials for V5.0.7 and later**
  Tutorials for using the developer toolkit in IBM API Connect Version 5.0.7 and later.
- V5.0.6 and earlier **Developer toolkit tutorials for V5.0.6 and earlier**
  Tutorials for using the developer toolkit in IBM API Connect Version 5.0.6 and earlier.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

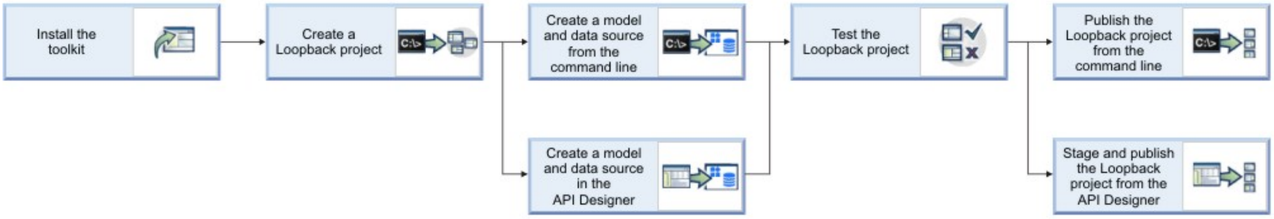
Developer toolkit tutorials for V5.0.7 and later

Tutorials for using the developer toolkit in IBM® API Connect Version 5.0.7 and later.

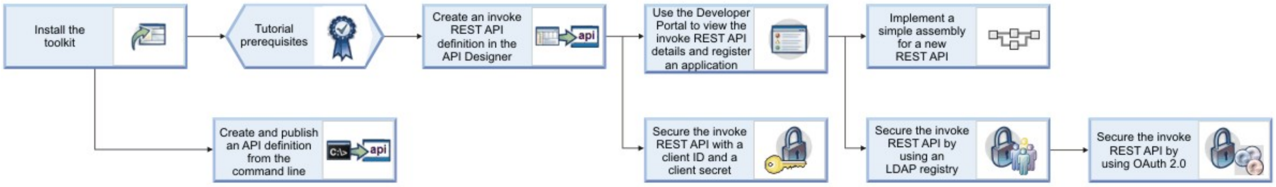
Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
With the developer toolkit tutorials you can either create API definitions that invoke an existing API implementation, or you can create your own API implementations with LoopBack®.

The following diagrams show the sequential flow through the IBM API Connect Developer toolkit tutorials. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.

Working with LoopBack projects


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

Working with API definitions that call an existing endpoint


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

The following sections contain the tutorials that take you through each of these scenarios:

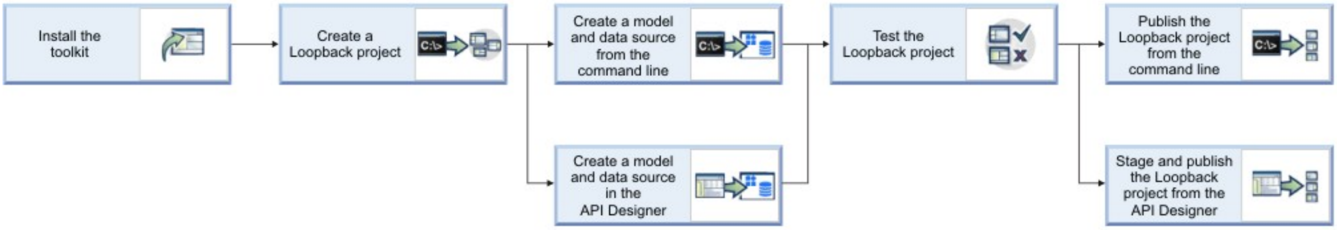
- **Tutorials for working with LoopBack projects**
  Tutorials for using LoopBack functions in the developer toolkit in IBM API Connect Version 5.0.7 and later.
- **Tutorials for working with API definitions that call an existing endpoint**
  Tutorials for creating API definitions to expose and secure BankA APIs in IBM API Connect Version 5.0.7 and later.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials for working with LoopBack projects

Tutorials for using LoopBack® functions in the developer toolkit in IBM® API Connect Version 5.0.7 and later.

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

- **Tutorial: Creating a LoopBack project from the command line**
  This tutorial shows you how to create a new LoopBack project by using the command line in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Creating a model and a data source from the command line**
  This tutorial shows you how to add a LoopBack model and data source to a project by using the `apic` command-line tool in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Creating a model and a data source in the API Designer**
  This tutorial shows you how to add a new model and data source to a LoopBack project by using the API Designer in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Installing LoopBack connectors**
  This tutorial shows you how to install a LoopBack data source connector manually by using the command line in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Testing a LoopBack project**
  This tutorial shows you how to run a LoopBack project locally for testing by using either the command line or the API Designer Explore tool in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Publishing a project from the command line**
  This tutorial shows you how to publish a LoopBack project from the command line in IBM API Connect Version 5.0.7 and later. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.
- **Tutorial: Staging and publishing a project from the API Designer**
  This tutorial shows you how to stage and publish a project using the API Designer in IBM API Connect Version 5.0.7 and later. *Staging* a project copies all the files to the target, but does not run the project application code. *Publishing* a project copies all the project files to the target and runs the project application code. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating a LoopBack project from the command line

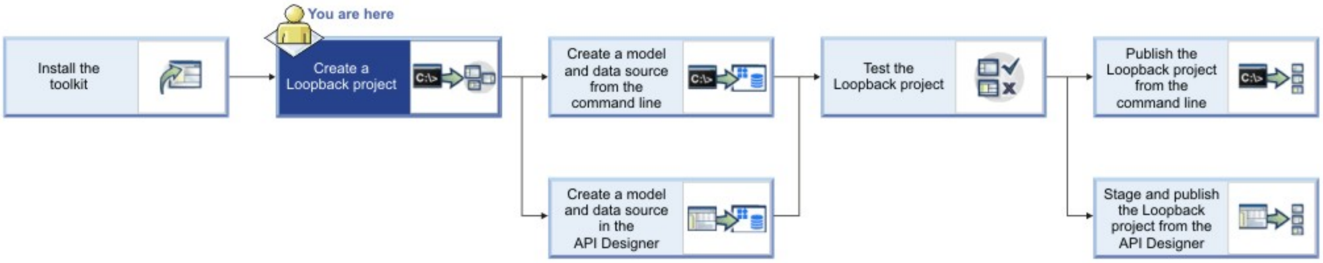
This tutorial shows you how to create a new LoopBack® project by using the command line in IBM® API Connect Version 5.0.7 and later.

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
You can accomplish the same thing using API Designer; for more information, see Creating new projects with API Designer. For more information on LoopBack, see LoopBack documentation.

Before you begin

Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

Note: To complete this tutorial, you must have internet access, because LoopBack installs dependencies from the public npm repository. Alternatively, you can configure a private npm repository.

Procedure

Complete the following steps:

1. From the command-line interface, enter the following command, which is used to create and manage LoopBack applications:

   ```
   apic loopback
   ```

   You will now create a project called "acme-bank".

2. At the prompt, enter `acme-bank` as the project name. Then, press Enter.

   ```
   ? What's the name of your application? acme-bank
   ```

   Note: In general, a project name can contain any characters except blank space (" "), forward slash ("/"), ampersand ("&"), at ("@"), plus ("+"), percent ("%"), and colon (":") .
3. Enter the name of the directory in which to create the project. You can press Enter to use a directory with the same name as the project, or type a new name and press Enter.

   ```
   ? Enter name of the directory to contain the project: acme-bank
   ```

4. Select the version of LoopBack to use. Choose the current production version, 3.x.

   ```
   ? Which version of LoopBack would you like to use? 3.x (current)
   ```

5. Specify the kind of application that you want to create by using the arrow keys to select empty-server:

   ```
   ? What kind of application do you have in mind? (Use arrow keys)
   ❯ empty-server (An empty LoopBack API, without any configured models or datasources)
     hello-world (A project containing a basic working example, including a memory datab
   ase)
     notes (A project containing a basic working example, including a memory database)
   ```

   Then, press Enter to create an empty Loop Back API.

The tool displays a number of messages as it creates the project directory and adds a number of directories and files to it. It also runs **`npm install`** to install all the project dependencies, as specified in **`package.json`**. This process creates a **`node_modules`** directory and might take some time.

An empty LoopBack project contains the following directories:

- **`server`**: contains server model and data source definitions, and other server code.
- **`definitions`**: contains YAML definition files.
- **`node_modules`**: created by node.js

For more information on the contents of a LoopBack project, see Project layout reference (LoopBack documentation).

What you did in this tutorial

In this tutorial, you created a new LoopBack project called "acme-bank".

What to do next

Create a model and a data source by following either of these tutorials:

- Tutorial: Creating a model and a data source from the command line.
- Tutorial: Creating a model and a data source in the API Designer.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating a model and a data source from the command line

This tutorial shows you how to add a LoopBack® model and data source to a project by using the **apic** command-line tool in IBM® API Connect Version 5.0.7 and later.

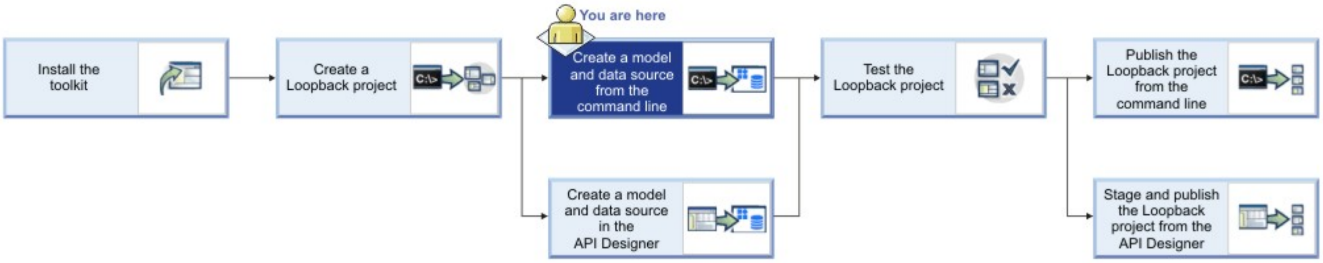
Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also create a LoopBack project (the "acme-bank" project) as described in Tutorial: Creating a LoopBack project from the command line and make sure the current working directory is the project root directory:

```
cd acme-bank
```

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

About this tutorial

You're going to start defining the acme-bank API that contains models for bank branches, accounts, and so on. In this tutorial, you will complete the following activities:

- Add a new data source to your LoopBack project.
- Add a model to your LoopBack project.

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**.

Adding a data source to a project

A LoopBack project requires at least one data source. To access the data source, the appropriate LoopBack connector must be installed. By default, an empty LoopBack project does not have any data sources defined. For now, you will add the in-memory data source, which is suitable for development and testing. Complete the following steps to add the data source to the project:

1. Change directories to the acme-bank project and enter the following command:

   ```
   apic create --type datasource
   ```

2. At the prompt, enter `bankDS` as the name of the data source:

   ```
   ? Enter the data-source name: bankDS
   ```

   Note: In general, you can use any alphanumeric character, dashes, and underscores in a data source name.
   The tool prompts you to select the connector to use for the data source:

   ```
   ? Select the connector for myds: (Use arrow keys)
   ❯ In-memory db (supported by StrongLoop)
     IBM DB2 (supported by StrongLoop)
     IBM DashDB (supported by StrongLoop)
     IBM MQ Light (supported by StrongLoop)
     IBM Cloudant DB (supported by StrongLoop)
     IBM DB2 for z/OS (supported by StrongLoop)
     MongoDB (supported by StrongLoop)
   (Move up and down to reveal more choices)
   ```

3. Press Enter to choose the local in-memory data source. This data source, suitable for development and testing, is built in to LoopBack. When working on your API development project, you would choose the connector for your back-end data store.
4. When prompted for the key to use for client persistence, press Enter without typing a value:

   ```
   ? window.localStorage key to use for persistence (browser only):
   ```

   For now, this will be a server-only data source, so this setting is not relevant.
5. When prompted for the path to a file to use for server persistence, press Enter without typing a value:

   ```
   ? Full path to file for persistence (server only):
   ```

   For now, leave this empty since you won't need to persist data across server restarts.

The tool updates the app's OpenAPI (Swagger 2.0) definition file and the **`server/datasources.json`** file with settings for the new data source. For more information, see Connecting models to data sources (LoopBack documentation).

Note:
The in-memory data source is built in to LoopBack and is suitable only for development and initial testing. When you are ready to connect your models to a real data source such as database server, follow the same procedure, but choose the connector for your back-end data store. The tool will prompt you for additional settings and will automatically install the appropriate connector package from npm.

The Oracle, DB2®, and SQLLite connectors have additional prerequisites; for more information, see Tutorial: Installing LoopBack connectors.

Adding a model to a project

The next step is to add a model to the project. Models represent back-end data sources such as databases or other back-end services (for example, REST or SOAP). Every LoopBack application has a set of default models, which you can extend to suit your application's requirements. You can also define custom models. Models are stored in JSON format and determine properties and other characteristics of the API. For more information on models, see Defining models. Follow these steps to add a model to your LoopBack project:

1. Ensure the current directory is the LoopBack project directory, **`acme-bank`**. From the command line, enter the following command:

   ```
   apic create --type model
   ```

2. At the prompt, enter `branch` as the name of the model:

   ```
   ? Enter the model name: branch
   ```

   Note: In general, you can use any alphanumeric characters plus dashes and underscores in a model name.
   The tool prompts you to select the data source to use from a list that includes the in-memory data source that you just added to the project:

   ```
   ? Select the data-source to attach item to: (Use arrow keys)
     (no data-source)
   ❯ bankDS (memory)
   ```

3. Use the arrow key to select bankDS (memory), and then press Enter.
   The tool prompts you to select the model's base class from a list that includes the LoopBack built-in models and any other models defined in the project.

   ```
   ? Select model's base class (Use arrow keys)
     Model
   ❯ PersistedModel
     ACL
   ```

```
  AccessToken
  Application
  Change
  Checkpoint
(Move up and down to reveal more choices)
```

If you were defining a user model, you would generally select the User as the base model; otherwise in most cases, you would choose PersistedModel as the base class for a custom model.

4. For this tutorial, use the arrow keys to select PersistedModel, and then press Enter.
5. When prompted as to whether you want to expose the model's REST API, press Enter to choose the default (yes):

   ```
   ? Expose branch via the REST API? (Y/n)
   ```

   Tip: If the model is exposed over REST, then all the standard create, read, update, and delete operations are available via REST endpoints; see PersistedModel REST API for more information.
   Since you chose to expose the model over REST, the tool prompts you for the plural form of the model name.

   ```
   ? Custom plural form (used to build REST URL):
   ```

6. Press Enter to use the default standard English rules for pluralization (in this case, "branches").
7. When prompted as to whether you want to create a server-only model, or a common model that can be used in both server or client LoopBack API, press Enter to keep the default (common):

   ```
   ? Common model or server only? (Use arrow keys)
   ```

8. When prompted to add properties to the model, enter `type` for the first property:

   ```
   Let's add some branch properties now.

   Enter an empty property name when done.
   ? Property name: type
   ```

9. When prompted to select the data type of the property, press Enter to select the string type:

   ```
   ? Property type: (Use arrow keys)
   ❯ string
     number
     boolean
     object
     array
     date
     buffer
   ```

10. To indicate that the property is required, enter `y`:

    ```
    ? Required? (y/N) y
    ```

    Then, press Enter.

11. When prompted for a default value, press Enter for no default value:

    ```
    ? Default value [leave blank for none]:
    ```

12. When prompted to add another property, add a property called `phone`:

    ```
    Let's add another branch property.
    Enter an empty property name when done.
    ? Property name: phone
    ```

13. Define the property type as **string** and set the property to **not required**.
14. When prompted to add another property, just press Enter to finish adding the model and update the Swagger definition.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Added a data source to your project using the command-line tool.
- Added a model to your project using the command-line tool.

What to do next

Test your LoopBack project.

Tutorial: Creating a model and a data source in the API Designer

This tutorial shows you how to add a new model and data source to a LoopBack® project by using the API Designer in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also do the following:

1. Create a LoopBack project. For more information, see Creating new projects with API Designer.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

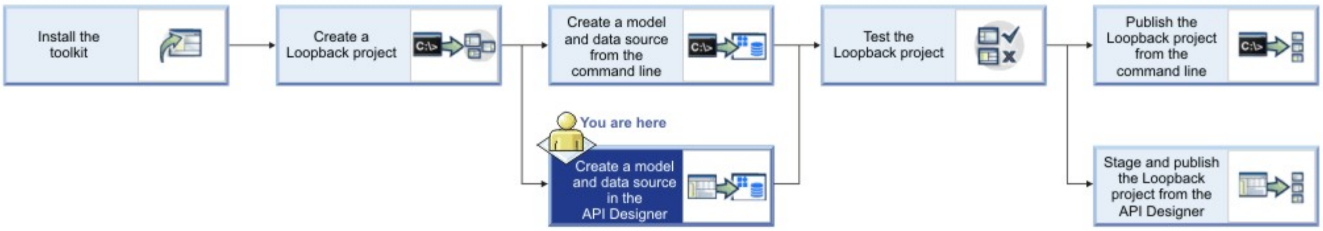
   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
You are here
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

About this tutorial

This tutorial builds on Tutorial: Creating a LoopBack project from the command line. In this tutorial, you will complete the following activities:

- Create a model with the API Designer.
- Create a data source with the API Designer.

Adding a data source

A LoopBack project requires at least one data source. To access the data source, the appropriate LoopBack connector must be installed. By default, an empty LoopBack project does not have any data sources defined. For now, you will add the in-memory data source, which is suitable for development and testing. Complete the following steps to add the data source to the project:

Complete the following steps:

1. Click Data Sources .
2. Click Add. The New LoopBack Data Source window opens.
3. Enter `bankDS` in the Name text field.
   Note: You can use any alphanumeric characters, dashes, and underscores in a data source name.
4. Click **New**.
5. By default, the Connector setting shows In-memory db and the other settings are blank. Keep the default settings for now, and API Designer automatically saves the new data source.
   Note: The In-memory data source is built in to LoopBack and is suitable only for development and initial testing. When you are ready to connect your models to a real data source such as database server, change the Connector setting accordingly then install the data source connector by following the instructions in Tutorial: Installing LoopBack connectors. Then enter the connector settings (host name, port, database name, user name, password) as appropriate for your Connector type, and click the Save icon 💾. Then, API Designer automatically tests the connection to the data source. If the test is successful, it displays the message Success - Data source connection test succeeded.
6. You can test the data source connection by clicking 🏃 . The message Success - Data source connection test succeeded is displayed.
7. Click All Data Sources. The data source will appear in the list of data sources. API Designer updates the **`server/datasources.json`** file with settings for the new data source.

Adding a model

The next step is to add a model to the project. Models represent back-end data sources such as databases or other back-end services (for example, REST or SOAP). Every LoopBack application has a set of default models, which you can extend to suit your application's requirements. You can also define custom models. Models are stored in JSON format and determine properties and other characteristics of the API. For more information on models, see Defining models. Follow these steps to add a model to your LoopBack project:

1. Click Models .
2. Click Add. The New LoopBack Model window opens.
3. Enter `branch` in the Name text field, then click New.
   Note: You can use any alphanumeric characters, dashes, and underscores in a model name.
4. In the Data Source field, select bankDS.
5. In the Properties section, click the Add property icon ⊕ .
6. In the Property Name text field, enter `type`.
7. For Type, select string.
8. Keep the default values for the other settings:
   - Required: Whether the property must have a value.
   - Is Array: Whether the property is a JavaScript array with elements of the specified type.
   - ID: Whether the property is a unique identifier. For more information on LoopBack ID properties, see LoopBack documentation.
   - Index: Whether the property represents a column (field) that is a database index.
   - Description: Text description of the property.

   For more information, see Model definition JSON file reference.
9. Click the Add property icon ⊕ again to add another property.
10. In the Property Name text field, enter `phone`.
11. For Type, select string.
12. Select Required to make the property required. This means that it must have a value when you add or update a model instance.
13. Select ID to indicate that the property must be unique. Leave the other settings with their default values.
14. Click the Save icon 💾 to save your changes.
15. Click All Models to finish editing the model.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a new LoopBack model.
- Created a new LoopBack data source.

What to do next

Test your LoopBack project.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Installing LoopBack connectors

This tutorial shows you how to install a LoopBack® data source connector manually by using the command line in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
To install a data source connector, you must be able to run an `npm install` command, which requires a connection to the internet. Alternatively, you can configure a private npm repository.

The Oracle, DB2®, and SQLLite connectors require C compiler tools to build and install binary extensions. The exact requirements depend on your operating system. Once you have installed compiler tools, you can install the connector following the instructions in the following procedure.

Windows
Note: You may need to restart your computer at various points during installation.
Install:

- Microsoft .NET Framework 4.0
- Visual Studio. Use **Visual Studio Community** unless you want purchase Visual Studio Enterprise. Run the installer, check "Visual C++" under "Programming Languages", and accept the default installation location.
  Note: The Visual Studio installation can take a long time.
- Windows SDK. If the installation fails, look for **C++ 2010 x64&x86 Redistributable** in your installed programs and uninstall it.
- Python 2.7.10.
- Version 3 of npm. Enter the following command:

  ```
  npm install -g npm
  ```

  Then ensure the `npm` command uses the correct version:

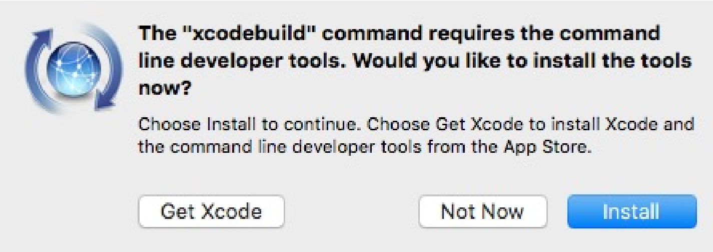
  ```
  npm -v
  ```

  If the version shown is not 3.*x.x*, then edit your system PATH to ensure that `C:\Users\username\AppData\Roaming\npm` supersedes any other entries.

- For 64-bit builds of Node.js and native modules you also need the Windows 7 64-bit SDK. If the installation fails, try uninstalling any C++ 2010 x64&x86 redistributable that you have installed first. If you get errors that the 64-bit compilers are not installed you may also need the compiler update for the Windows SDK 7.1.

Note: Do not use Cygwin (Windows bash shell emulator). Use the Windows command shell instead.
Mac OS X
You must have the command-line developer tools (or full installation of Xcode) and Python.

If you don't already have Xcode or the command-line tools installed, installation will prompt you as follows:


The "xcodebuild" command requires the command line developer tools. Would you like to install the tools now?

Choose Install to continue. Choose Get Xcode to install Xcode and the command line developer tools from the App Store.

Click Install to install the command-line developer tools only. Alternatively, you can click Get Xcode to install the full Xcode product, but doing this can take a long time.

Python is also required. Most versions of OSX come with Python by default. If for some reason you don't have it, download and install Python.

Linux

Many Linux® systems come with the necessary tools. The specific requirements are:

- Python version 2.7. **NOTE**: version 3.*x* is not supported.
- `make`
- A C/C++ compiler toolchain, like GCC. **NOTE**: g++ version 4.2 or later is required.

On Debian and Debian-derived distributions (Ubuntu, Mint, and so on), use the command:

```
apt-get install build-essential
```

About this task

Before you can use a LoopBack data source to access data in a backend system such as a database, you must install the data source connector. The In-memory and email connectors are built-in to LoopBack, so you don't need to install them. The command-line tool and API Designer will automatically prompt you to install connectors as needed, but you can also do it manually.

Install a new data source connector manually by following these steps:

Procedure

1. Open a new shell window.
2. Install the connector package by entering this command in the project root directory:

   ```
   npm install --save connector-package
   ```

   where *connector-package* is the name of the npm package for the LoopBack connector, as shown in the table.

   Table 1. LoopBack Connectors

Data Source	Connector Package
Database connectors	
Cloudant®	loopback-connector-cloudant
DashDB	loopback-connector-dashdb
DB2	loopback-connector-db2
DB2 for zOS	loopback-connector-db2z
Microsoft SQL Server	loopback-connector-mssql
MongoDB	loopback-connector-mongodb
MQ Light	loopback-connector-mqlight
MySQL	loopback-connector-mysql
Oracle	loopback-connector-oracle
PostgreSQL	loopback-connector-postgresql
Other connectors	
SOAP web services	loopback-connector-soap
REST web services	loopback-connector-rest

3. For Oracle, you must modify /etc/environment, reboot the system, then enter the following commands:

   ```
   $ echo "/home/strongbot/orapp2/node_modules/instantclient" | sudo tee -a
   /etc/ld.so.conf.d/loopback_oracle.conf
   $ sudo ldconfig
   ```

Results

You successfully installed a new data source connector for LoopBack.

Installing the Oracle connector

About this task

The LoopBack Oracle connector might require some additional installation and troubeshooting steps. Either follow the manual installation and troubleshooting steps or install the LoopBack CLI that provides an additional command to install and troubleshoot the Oracle connector:

- Follow the instructions in Installing the Oracle connector for your platform.
- If you prefer (or if you still have issues with the connector), install the LoopBack CLI by using the comand:

  ```
  npm install -g loopback-cli
  ```

  Then enter the following command to install and troubleshoot the Oracle connector:

  ```
  lb oracle
  ```

  This command determines if the connector is ready to use. If it is, the tool will print "Oracle connector is ready" and exit. Otherwise, it will prompt you to install Oracle Instant Client, `loopback-connector-oracle`, and the `oracledb` module. For more information, see Oracle installer command.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

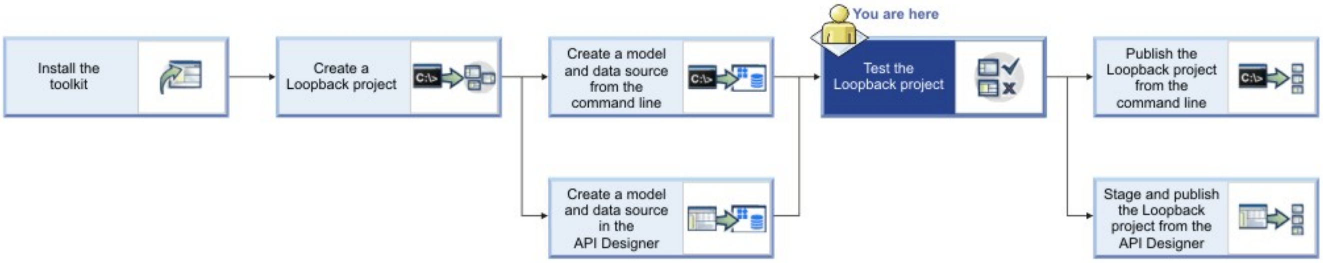
Tutorial: Testing a LoopBack project

This tutorial shows you how to run a LoopBack® project locally for testing by using either the command line or the API Designer Explore tool in IBM® API Connect Version 5.0.7 and later.

Before you begin

Important: From IBM API Connect Version 5.0.8.7, the `apic start` command referred to on this page is no longer supported by the API Connect toolkit. This tutorial therefore applies only to IBM API Connect versions earlier than Version 5.0.8.7.
Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

Also ensure your current working directory is the project top-level directory. Enter the following command:

```
cd acme-bank
```

About this tutorial

This tutorial builds on Tutorial: Creating a model and a data source in the API Designer or Tutorial: Creating a model and a data source from the command line (you only need to complete one of these). In this tutorial, you will complete the following activities:

- Test a LoopBack project by using the command line
- Test a LoopBack project by using the API Designer Explore tool

Note: You can test a LoopBack project either by using the command line or the API Designer, depending on your preference. The procedures accomplish the same result, which is to run the LoopBack project (Node.js application) and the Micro Gateway on your local system.

Test a LoopBack project by using the command line

Complete the following steps:

1. Start the local Micro Gateway (using the default TLS certificate) with the following command:

   ```
   TLS_SERVER_CONFIG=node_modules/microgateway/config/defaultTLS.json apic start
   ```

   This runs the LoopBack project (API) and the Micro Gateway locally. You will see the message:

   ```
   Service acme-bank (id 1) started on port 4001
   Service acme-bank-gw (id 2) started on port 4002
   ```

   Note: If you previously ran other projects, you may see different port numbers.
2. To confirm that the project is running locally, open `http://localhost:4001` in your browser. For the default LoopBack (empty-server or hello-world) project, you'll see something like this:

   ```
   {"started":"2016-03-07T22:24:55.322Z","uptime":35.839}
   ```

3. You can then test the API endpoints by using `curl`. First, create a model instance by entering the following command:

   ```
   curl --request POST \
     --url 'https://localhost:4002/api/branches' \
     --header 'accept: application/json' \
     --header 'content-type: application/json' \
     --header 'x-ibm-client-id: default' \
     --header 'x-ibm-client-secret: SECRET' \
     --data '{"type":"ATM location", "phone": "555-1212"}' \
   ```

   The console displays the data that is added to the model, ("type":"ATM location","phone":"555-1212"}. Note the `id` property that LoopBack adds automatically.
   Note: If you receive the following error from curl: `curl: (60) SSL certificate problem: Invalid certificate chain`, and you are accessing a local server, you can turn off the certificate verification by using the `-k` or `--insecure` option. If you are accessing an external server, the non-secure options are not recommended.
4. To display all of the model instances in the "acme-bank" project, enter the following command:

   ```
   curl --request GET \
     --url 'https://localhost:4002/api/branches' \
     --header 'accept: application/json' \
     --header 'content-type: application/json' \
     --header 'x-ibm-client-id: default' \
     --header 'x-ibm-client-secret: SECRET' \
   ```

   The console displays the JSON data in the model.
   Note: If you receive the following error from curl: `curl: (60) SSL certificate problem: Invalid certificate chain`, and you are accessing a local server, you can turn off the certificate verification by using the `-k` or `--insecure` option. If you are accessing an external server, the non-secure options are not recommended.

Test a LoopBack project by using the API Designer Explore tool

To test your API endpoints using the API Designer Explore tool, complete the following steps:

1. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
Note: If you need to run the editor on a different port, use the following command:
Linux Mac OS X

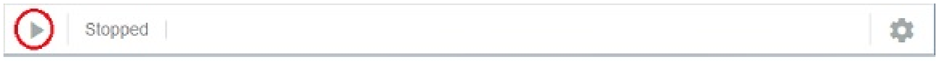
```
PORT=port_number apic edit
```

Windows

```
set PORT=port_number && apic edit
```

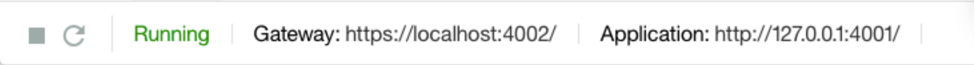
where *port_number* is the port number to use.

2. Start the local test servers by completing the following steps:
   a. In the test console at the bottom of the screen, click the Start the servers icon:



Stopped

   b. Wait until the `Running` message is displayed:



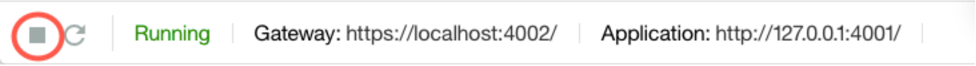
Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/

Depending on your project configuration and whether other processes are running, different port numbers might be displayed.

3. Click http://127.0.0.1:*port_number* to display the API root endpoint. For the default LoopBack (empty or hello-world) project, you'll see something like this:

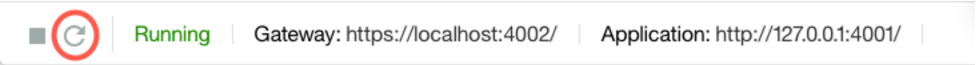
```
{"started":"2016-03-07T22:24:55.322Z","uptime":35.839}
```

Note: To stop your project, click the Stop the servers icon:



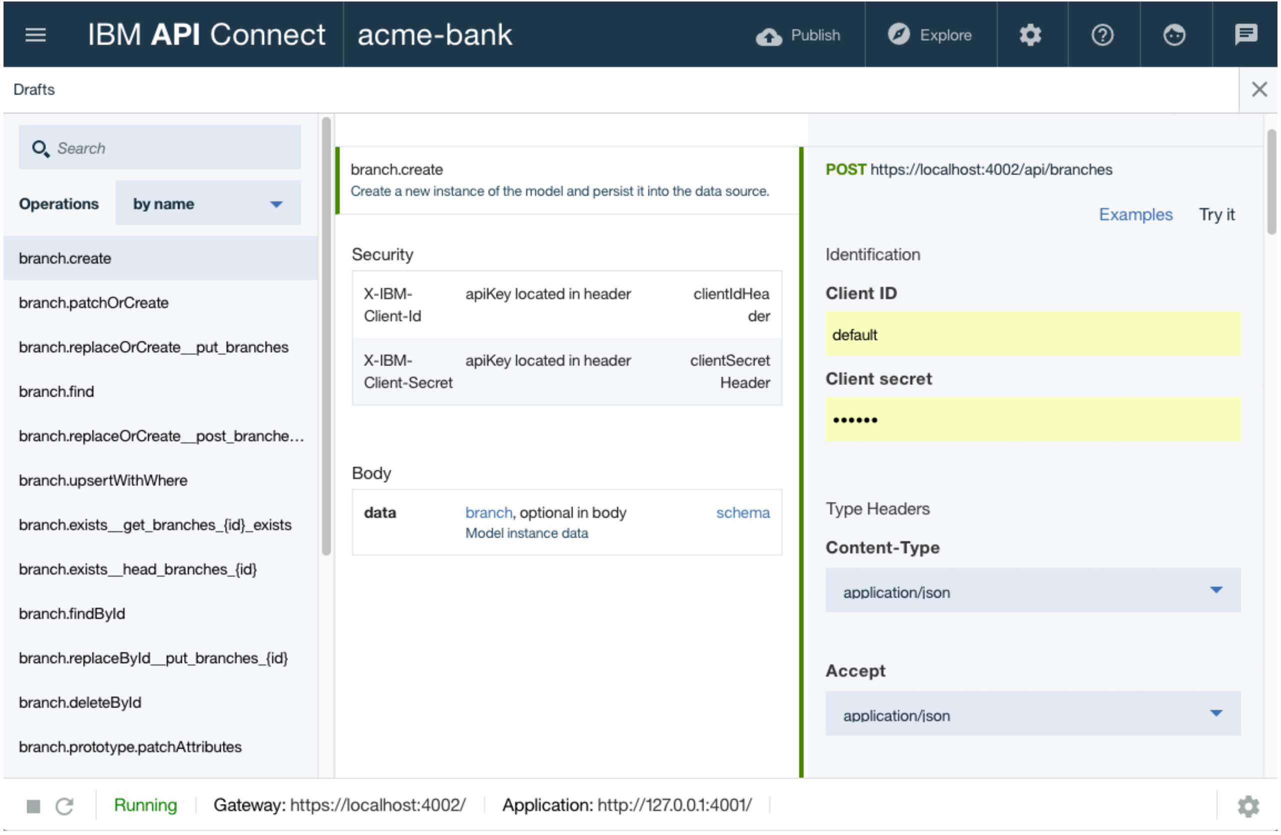
Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/

To restart it, click the Restart the servers icon:



Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/

4. Click Explore. You will see the API Explore tool. The side bar shows all the REST operations for the LoopBack models in the API. Models that are based on `PersistedModel` by default have a standard set of create, read, update, and delete operations.
5. Click the operation branch.create in the left pane to display the endpoint.



branch.create
branch.patchOrCreate
branch.replaceOrCreate__put_branches
branch.find
branch.replaceOrCreate__post_branche…
branch.upsertWithWhere
branch.exists__get_branches_{id}_exists
branch.exists__head_branches_{id}
branch.findById
branch.replaceById__put_branches_{id}
branch.deleteById
branch.prototype.patchAttributes

branch.create
Create a new instance of the model and persist it into the data source.

Security

X-IBM-Client-Id	apiKey located in header	clientIdHeader
X-IBM-Client-Secret	apiKey located in header	clientSecretHeader

Body

data	branch, optional in body Model instance data	schema

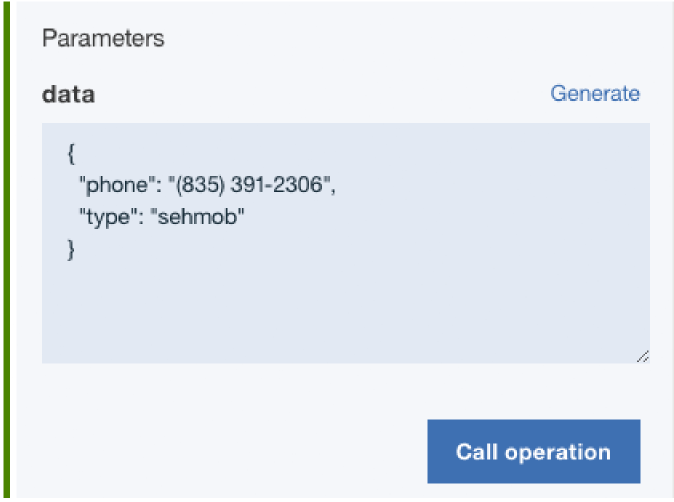
POST https://localhost:4002/api/branches

Identification
Client ID: default
Client secret: ••••••
Type Headers
Content-Type: application/json
Accept: application/json

Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/

The center pane displays summary information about the endpoint, including its parameters, security, model instance data, and response codes. The right pane provides template code to call the endpoint using the `curl` command, with a drop down menu to choose languages such as Ruby, Python, Java™, and Node.

6. To test the REST endpoints in the API Explore tool, on the right pane click Try it, and then scroll down to Parameters and click Generate to generate some dummy data. By default, the generated data includes the properties defined for the endpoint, for example:



```
{
  "phone": "(835) 391-2306",
  "type": "sehmob"
}
```

7. Click Call operation to call the endpoint with that data. You should see the request and response parameters, along with the JSON instance data that you entered.
   Note: If you see an error message due to an untrusted certificate for `localhost`, click the link provided in the error message in API Explore to accept the certificate, then proceed to call the operations in your web browser. The exact procedure depends on the web browser you are using.
   If you load the REST endpoints directly in your browser, you will see the message: `{"name":"PreFlowError","message":"unable to process the request"}`. You must use API Explore to test REST endpoints in your browser because it includes the requisite headers and other request parameters.

8. Test the REST endpoints by using the `curl` command shown, for example:

```
curl --request POST \
  --url https://localhost:4002/api/branches \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --header 'x-ibm-client-id: default' \
  --header 'x-ibm-client-secret: SECRET' \
  --data '{"type":"ATM location","phone":"408-123-4567"}' -k
```

9. Paste the command into a console window being sure to include the `-k` at the end of the command (as shown in the example) to avoid certificate errors. If you wish, edit the JSON data to make it more meaningful. When you enter the command, the console will show the data entered, for example `{"type":"ATM location","phone":"408-123-4567"}`.
10. To confirm that the operation added a model instance, click branch.find then click Call operation with no filters to display all branch instances. An example result (with two model instances) would blook like this:

```
[
{
     "type": "standalone",
     "phone": "831-555-1212"
   },
   {
     "type": "ATM location",
     "phone": "408-123-4567"
   }
 ]
```

You can experiment with other operations if you wish, to get a feeling for the standard REST endpoints of a LoopBack `PersistedModel`.

V5.0.6 + Note: You can also run the API Explore tool directly from the command line. Ensure that your local test servers are running, then run the command **apic explore**. The API Explore tool opens, and shows the operations, definitions, and documentation for all of the APIs that are contained in your project directory. You can specify a single API to explore by specifying the name of the API in the command. The `-e` or `--external` options open the Explore tool on 0.0.0.0 instead of the default 127.0.0.1. The external option binds the server to all IP addresses on the machine, and makes the tool accessible on the wider network.

What you did in this tutorial

In this tutorial, you completed at least one of the following activities:

- Tested a LoopBack project from the command line.
- Tested a LoopBack project from the API Designer Explore tool.

What to do next

Publish your project by following either of these tutorials:

- Tutorial: Publishing a project from the command line
- Tutorial: Staging and publishing a project from the API Designer

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

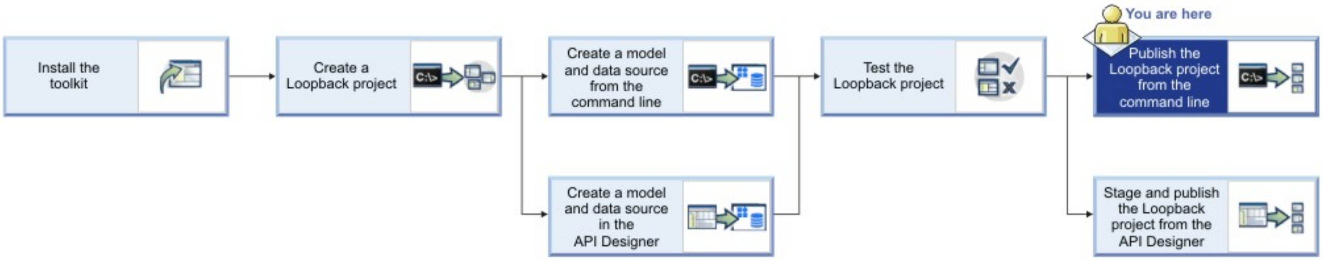
Tutorial: Publishing a project from the command line

This tutorial shows you how to publish a LoopBack® project from the command line in IBM® API Connect Version 5.0.7 and later. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.



Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
You are here
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

Also ensure your current working directory is the project top-level directory. Enter the following command:

```
cd acme-bank
```

To stage or publish to API Connect collective, you must install and configure an API Connect collective. You must add a Collective and a Collective controller by using the Cloud Manager. See Installing an API Connect collective.

Note: IBM API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. However, for this tutorial you will continue to publish to an API Connect collective.
You must also fulfill the following additional prerequisites to stage or publish project:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.
2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.

About this tutorial

In this tutorial, you will publish a LoopBack project to an API Connect collective from the command line.

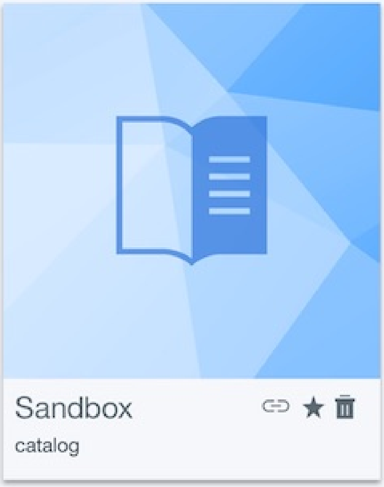
Display general command-line help information by entering the following command: **`apic --help`** or **`apic -h`**. Display help information for a specific **`apic`** command by entering the following command: **`apic`** ***`command_name`*** **`--help`** or **`apic`** ***`command_name`*** **`-h`**.

Publishing a project to API Connect collective from the command line

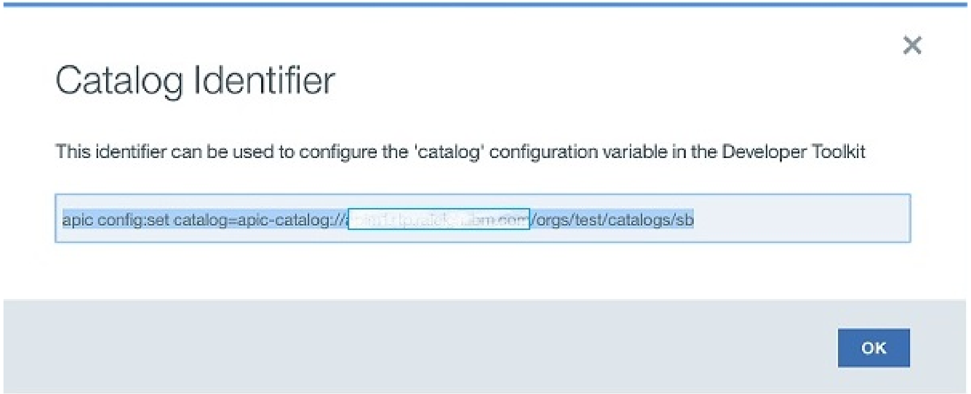
Be sure you have added the API Connect collective and a Collective controller to the Cloud Manager as described in Installing an API Connect collective.

Note: If your application has binary modules (for example, if it uses the DB2®, SQLLite3, or Oracle connectors), then you must create the application on the same type of platform as that of the target API Connect collective.

1. Obtain the identifier for the development Catalog, to which you will publish your project, by completing the following steps in the API Manager user interface:
   a. Log in to the API Manager user interface.
   b. On the Dashboard, by default, you see the development Catalog, called Sandbox.



   c. Click the Show catalog identifier icon: ⊂⊃.
   d. Copy the Catalog identifier from the Catalog Identifier dialog.



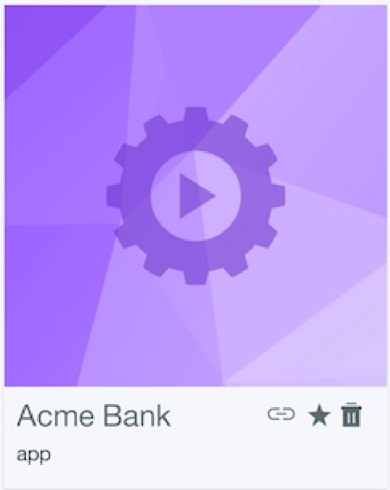
Catalog Identifier

This identifier can be used to configure the 'catalog' configuration variable in the Developer Toolkit

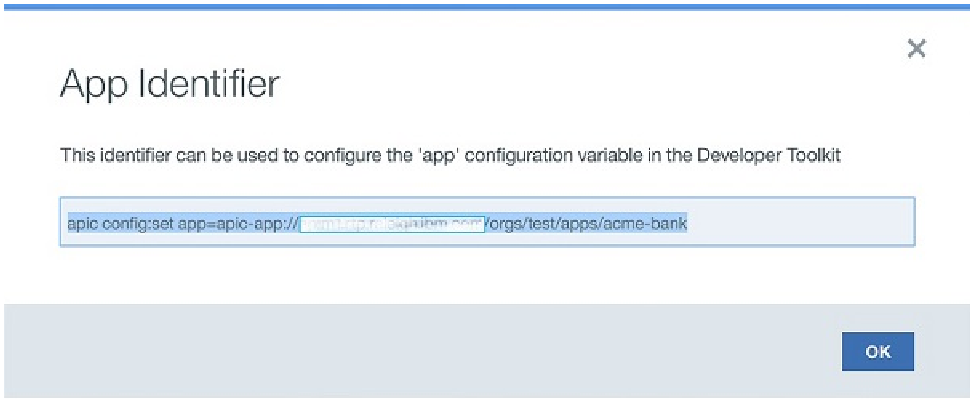
```
apic config:set catalog=apic-catalog://[redacted]/orgs/test/catalogs/sb
```

Keep this string handy, because you use it in the following procedure.

   e. Click ⊕ Add > App.
   f. In the Add App dialog, enter `Acme Bank` for Display Name, and keep the automatically-generated default value of Name. Select the collective you created previously, then click Add.
   g. The API Manager dashboard displays the App tile.



   h. Click the Show app identifier icon ⊂⊃ and copy the App identifier from the App Identifier window.


App Identifier

This identifier can be used to configure the 'app' configuration variable in the Developer Toolkit

```
apic config:set app=apic-app://[redacted]/orgs/test/apps/acme-bank
```

Keep this string handy, since you use it in the procedure below.

2. Complete the following steps in a command console on the machine where you installed the toolkit:
   a. If you haven't already done so, log in to API Manager by entering the following command:

      ```
      apic login -s API_manager_hostname -u username -p password
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager, *username* is your API Manager user name and *password* is your API Manager password.
   b. Paste the command strings you copied in steps 1.d and 1.h into your command line console. For example:

      ```
      apic config:set catalog=apic-catalog://API_manager_hostname/orgs/climbon/catalogs/sb
      apic config:set app=apic-app://API_manager_hostname/orgs/climbon/apps/acme-bank
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager.
   c. Ensure your current working directory is the project root directory (`acme-bank`). Publish the Product by entering the following command:

      ```
      apic publish -s API_manager_hostname definitions/acme-bank-product.yaml
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager. The console displays messages that confirm the Product has been published, for example:

      ```
      Staged definitions/acme-bank-product.yaml to climbon:sb [acme-bank1:1.0.0]
      Published definitions/acme-bank-product.yaml to climbon:sb [acme-bank1:1.0.0]
      ```

   d. Publish the application by entering the following command:

      ```
      apic apps:publish
      ```

      The console displays messages such as this:

      ```
      ...preparing project
      ...building package for deploy
      ...uploading packages to 161.51.151.222:9443, scale: 1
      Upload successful: acme-bank-5707fc46e4b07dfbe0999be6-1460148297009-package.tgz
      Upload successful: acme-bank-5707fc46e4b07dfbe0999be6-1460148297009.deploy.xml
      Upload successful: apic.acme-bank-5707fc46e4b07dfbe0999be6-1460148297009.scalingPolicy.xml

      Upload to liberty server completed succesfully.
      Applications may take a few minutes to update and start.
      Collectives admin center: https://161.51.151.222:9443/adminCenter.
      ```

The project is now published to your API Connect collective.

What you did in this tutorial

In this tutorial, you published a project to API Connect collective by using the command-line tools.

What to do next

If you want to use an application with a DataPower® Gateway, configure your collective and gateway and then modify your assembly. For more information, see Configuring your DataPower Gateway and API Connect collective controller to communicate and Modifying the assembly to call an application endpoint hosted on a collective.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Staging and publishing a project from the API Designer

This tutorial shows you how to stage and publish a project using the API Designer in IBM® API Connect Version 5.0.7 and later. *Staging* a project copies all the files to the target, but does not run the project application code. *Publishing* a project copies all the project files to the target and runs the project application code. You publish a LoopBack® project to make its APIs available to application developers through the Developer Portal.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

To stage or publish to API Connect collective, you must install and configure an API Connect collective. You must add a Collective and a Collective controller by using the Cloud Manager. See Installing an API Connect collective.

Note: IBM API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. However, for this tutorial you will continue to publish to an API Connect collective.
You must also fulfill the following additional prerequisites to stage or publish project:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.
2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.

You must also do the following:

1. Create a LoopBack project. For more information, see Tutorial: Creating a LoopBack project from the command line.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

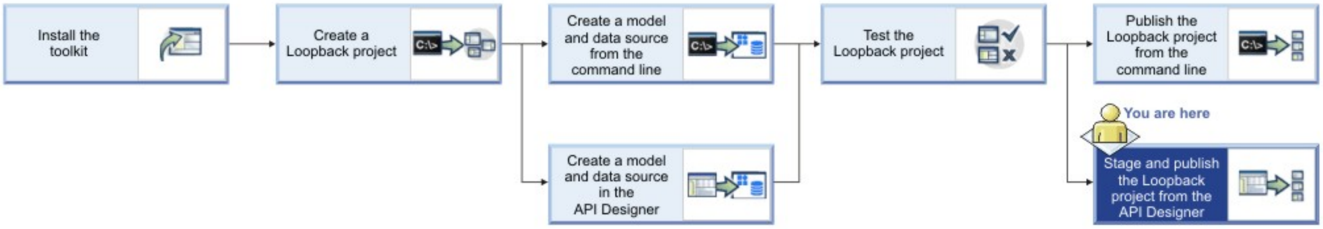
   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
You are here
Stage and publish the Loopback project from the API Designer

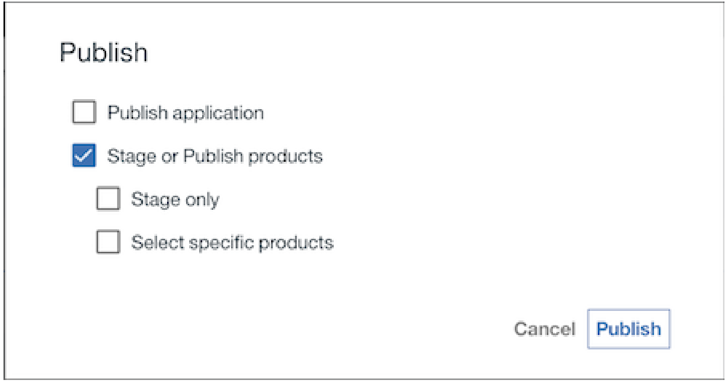
About this tutorial

In this tutorial, you will stage and publish a LoopBack project to an API Connect collective from the API Designer.

Publishing a project to an API Connect collective

Note: If your app has binary modules (for example, if it uses the DB2®, SQLLite3, or Oracle connectors), then you must create the app on the same type of platform as that of the target API Connect collective.

1. In the API Designer, click Publish then click Add and Manage Targets.
2. Click Add a different target. You'll see the Sign in to IBM API Connect dialog.
3. Enter your API Connect host address, Username, and Password. Click Sign in. You'll see the Select an organization and catalog dialog.
4. Select the desired organization, and select the Sandbox Catalog.
5. Click Next. You'll see the Select an App dialog. Click the desired app, and then click Save.
6. In the API Designer, click Publish again.
7. In the drop-down selection, under Other Orgs, click the target you just created. You'll see the Publish dialog:

   Publish

   ☐ Publish application

   ☑ Stage or Publish products

   ☐ Stage only

   ☐ Select specific products

8. Select Publish application and Stage or Publish products, then select Select specific products. Select acme-bank, and then click Publish.

While the project is being published, the console displays messages like the following output:

```
Staged /Users/john/acme-bank/definitions/acme-bank-product.yaml to climbon:sb [acme-bank:1.0.0]
Published /Users/john/acme-bank/definitions/acme-bank-product.yaml to climbon:sb [acme-bank:1.0.0]
Successfully published products
...building package for deploy
Creating keys...this may take some time
...uploading packages to 169.53.159.240:9443, scale: 1
Upload successful: acme-bank-57057c0ee4b011906e320bda-1459977334690-package.tgz
Upload successful: acme-bank-57057c0ee4b011906e320bda-1459977334690.deploy.xml
Upload successful: apic.acme-bank-57057c0ee4b011906e320bda-1459977334690.scalingPolicy.xml

Upload to liberty server completed succesfully.
Applications may take a few minutes to update and start.
```

Once the project is published, the API Designer displays a Success message.

If you want to confirm that the application was published in your API Connect collective, you can connect to the Liberty Admin Center by clicking Manage this app from within your App in the API Manager.

What you did in this tutorial

In this tutorial, you published a project to an API Connect collective by using the API Designer.

What to do next

If you want to use an application with a DataPower® Gateway, configure your collective and gateway and then modify your assembly. For more information, see V5.0.2 + Configuring your DataPower Gateway and API Connect collective controller to communicate and V5.0.2 + Modifying the assembly to call an application endpoint hosted on a collective.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials for working with API definitions that call an existing endpoint

Tutorials for creating API definitions to expose and secure BankA APIs in IBM® API Connect Version 5.0.7 and later.

Introduction

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
BankA has an existing set of REST-based services that it wants to expose through APIs, to help it grow within the mobile and device application market. The BankA business team knows that an increased mobile and device application presence will promote their brand image and result in increased customer numbers.

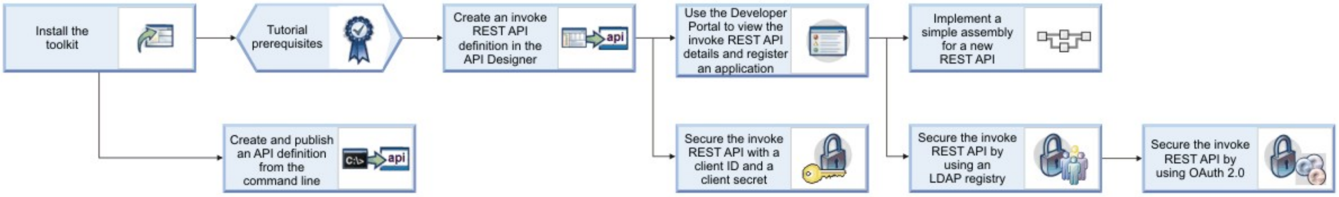
In the following tutorials, you develop the BankA API management solution. The initial solution includes the documentation of a BankA branch information API and the implementation of a pure proxy to access the branch information REST service. This simple proxy allows BankA to monitor the use of the service and set rate limits on the API. You will also document, create, and implement a new operation by using an existing RESTful service.

After you define the API operations, you create and publish a Plan to socialize the API operations.

Learning objectives

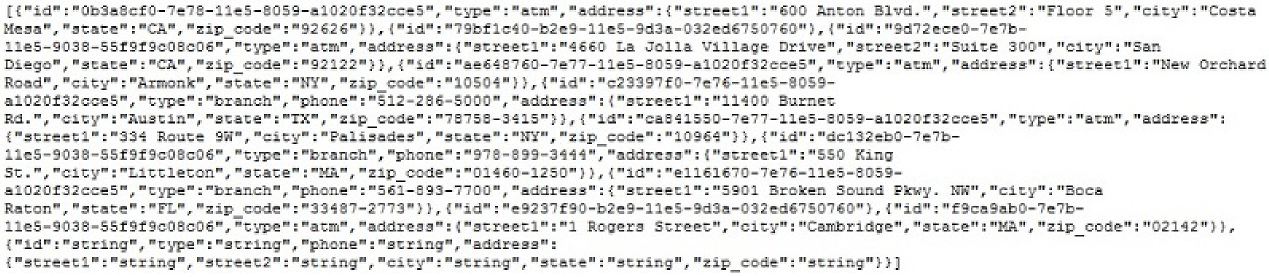
During these tutorials you will learn how to create, define, and test an API. You will also learn how to create and test an assembly API, and how to use the Developer Portal. Security options are also demonstrated.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

Prerequisites

1. Install the developer toolkit if you have not already done so. For more information, see Installing the toolkit.
2. Open your browser.
3. To ensure that the JSON formatted response of the BankA branch information API is operational, enter the URL **`https://apictutorials.mybluemix.net/branches`** and verify that the API provides a response that is similar to the following example:



```
[{"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":"79bf1c40-b2e9-11e5-9d3a-032ed6750760"},{"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":"e9237f90-b2e9-11e5-9d3a-032ed6750760"},{"id":"f9ca9ab0-7e7b-
11e5-9038-55f9f9c08c06","type":"atm","address":{"street1":"1 Rogers Street","city":"Cambridge","state":"MA","zip_code":"02142"}},
{"id":"string","type":"string","phone":"string","address":
{"street1":"string","street2":"string","city":"string","state":"string","zip_code":"string"}}]
```

To complete the tutorial tasks that involve publishing your API, you must fulfill the following additional prerequisites:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.

2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.
3. If you have internet connectivity, to run the API Designer you must have a IBM Cloud account. To create a IBM Cloud account, use the IBM Cloud registration page.

You publish an API to make it available to application developers through the Developer Portal.

- **Tutorial: Creating an invoke REST API definition**
  This tutorial shows you how to define and implement a REST API definition that proxies an existing service in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Securing an API with a client ID and client secret**
  This tutorial shows you how to secure an API so that a calling application must supply a client ID and a client secret in IBM API Connect Version 5.0.7 and later. This option is similar to requiring a user ID and password to be supplied.
- **Tutorial: Securing APIs by using an LDAP user registry**
  This tutorial shows you how to secure an API with an LDAP registry so that LDAP credentials must be supplied when the API is called, in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Securing an API by using OAuth 2.0**
  This tutorial shows you how to secure an API by using OAuth 2.0 so that an application can access the API on a user's behalf, in IBM API Connect Version 5.0.7 and later.
- **Tutorial: Creating and publishing an API definition from the command line**
  This tutorial shows you how to use the developer toolkit to create an API definition, include it in a Product, and publish the Product to a Catalog in API Manager by using IBM API Connect Version 5.0.7 and later.

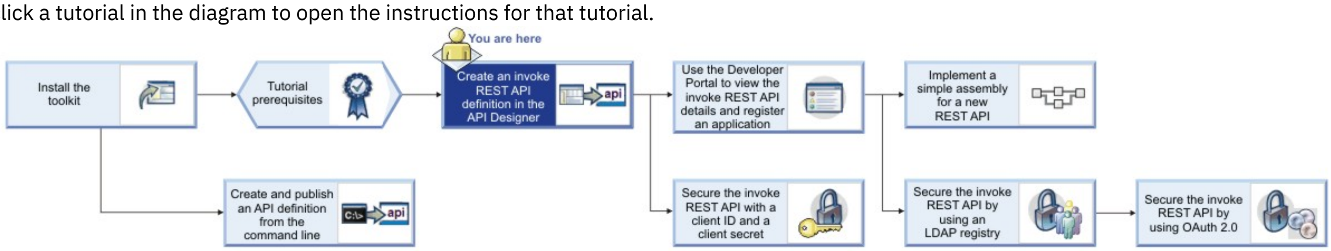
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating an invoke REST API definition

This tutorial shows you how to define and implement a REST API definition that proxies an existing service in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

In this tutorial you will complete the following lessons:

1. Creating a REST API definition
2. Testing the REST API
3. Creating a Product and a Plan for the REST API
4. Publishing your Product

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. Create a folder to hold your API and Product definitions, and change to that folder in a command window.
2. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

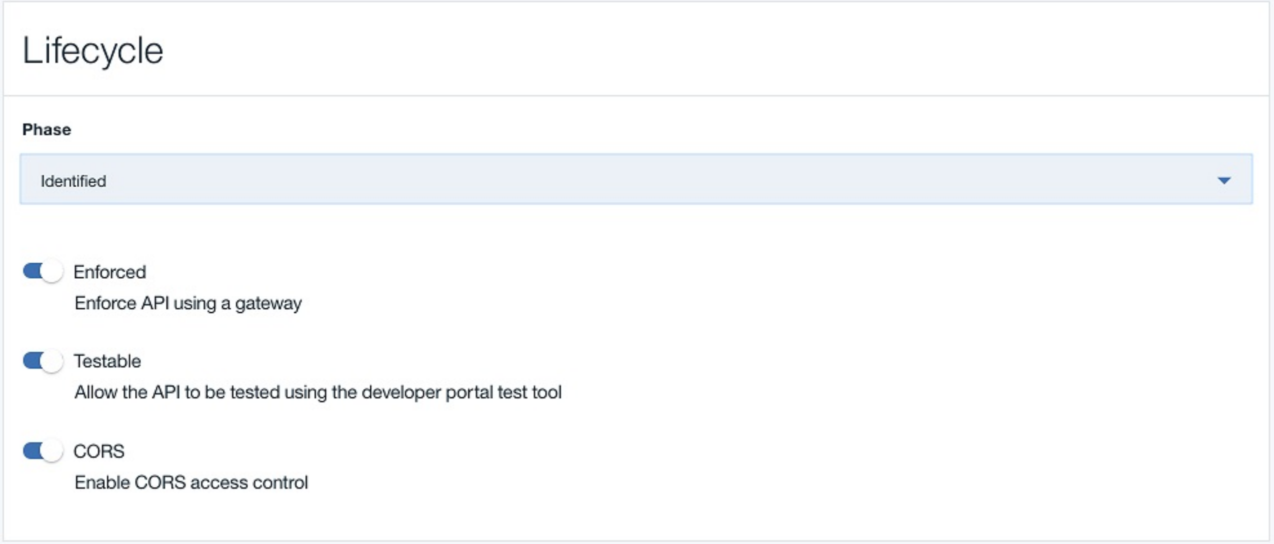
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
3. Log in to APIConnect Designer with the appropriate method.
4. Click Add > New API.
5. Enter the appropriate information to create a REST API definition.
   a. In the Title field, enter `Branches`.
   b. The Name and Base Path fields autopopulate with the terms `branches` and `/branches` respectively.
   c. Leave the Version field at `1.0.0`.
   d. Leave the default Additional properties as they are.
6. You do not add a product at this time, click Create API.
7. If the API Editor help screen appears, click the sentence Learn more about composing APIs, or click Got it! to access the main screen immediately.
8. In the side bar, click Lifecycle to display the Lifecycle panel. Ensure that the Enforced, Testable, and CORS toggles are set to the On position as shown in the following screen capture:



Lifecycle

Phase

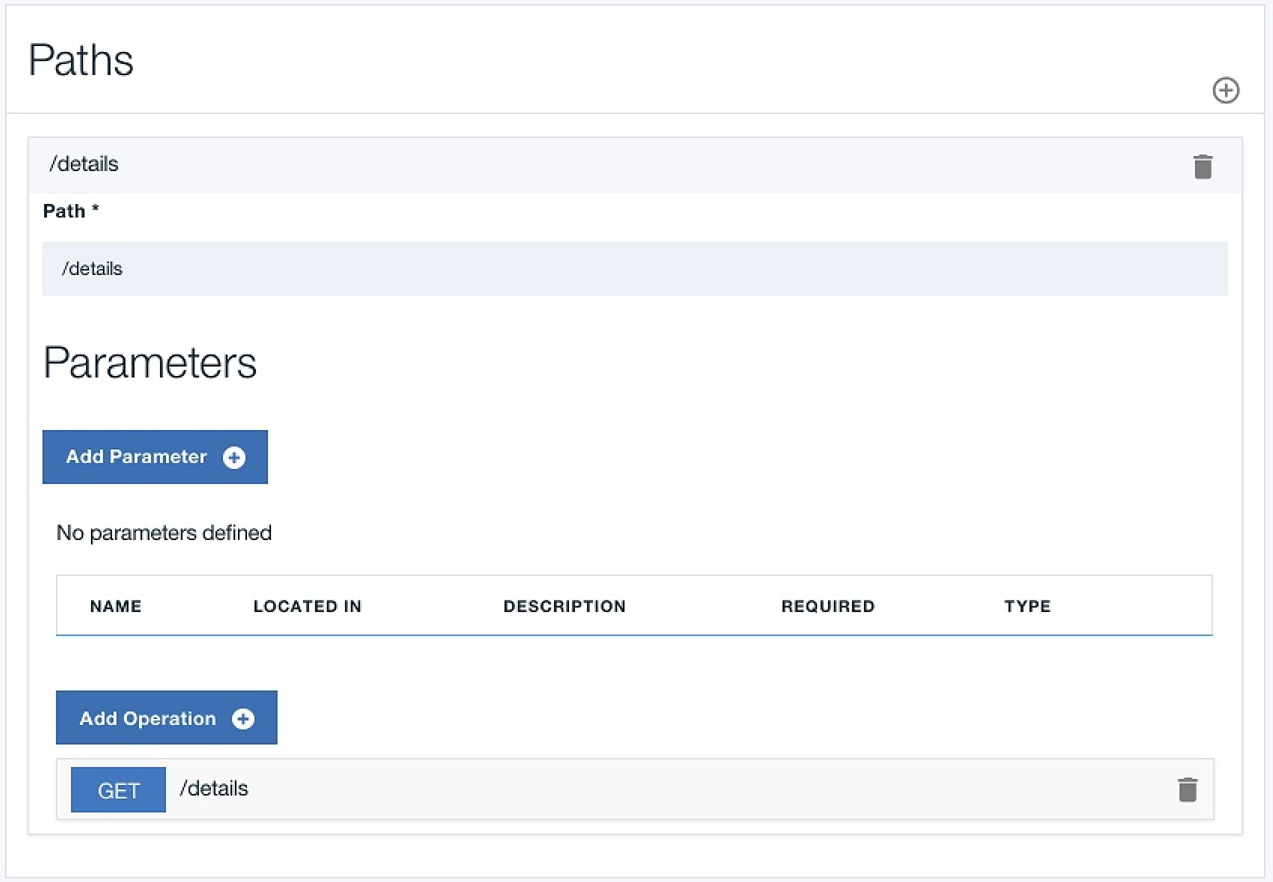
Identified

Enforced
Enforce API using a gateway

Testable
Allow the API to be tested using the developer portal test tool

CORS
Enable CORS access control

9. In the side bar, click Security Definitions to display the Security Definitions panel. Notice that `clientIdHeader` security definition already exists, and in the Security section you see that `Option 1` is active with `clientIdHeader (API Key)`.
10. In the side bar, select Paths to display the Paths panel. Create a new path by clicking the Add Path icon ⊕.
11. In the Path field, replace the default path segment with `/details`. When an operation is called, this path segment is appended to the URL of your API.



Paths

/details

Path *

/details

Parameters

No parameters defined

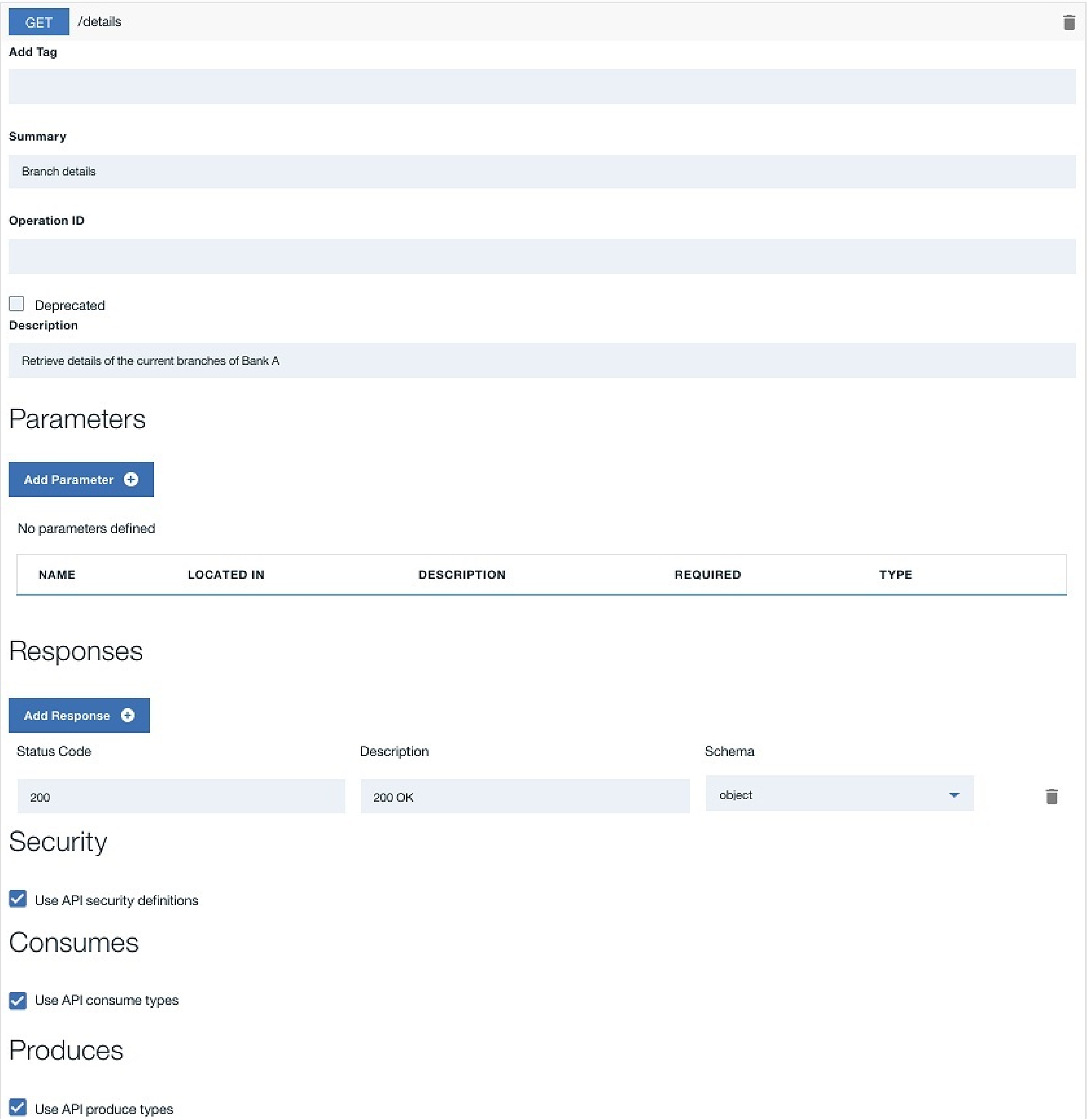
NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

/details

12. By default, a single GET operation is already in your Path. Click the GET button to expand the setting dialog.
13. For the operation, provide a summary and a description as in the following table.

Table 1. Operation definition values

Field	Value
Summary	`Branch details`
Description	`Retrieve details of the current branches of BankA`



Add Tag

Summary: Branch details

Operation ID

☐ Deprecated

Description: Retrieve details of the current branches of Bank A

Parameters

No parameters defined

NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

Responses

Status Code	Description	Schema
200	200 OK	object

Security

☑ Use API security definitions

Consumes

☑ Use API consume types

Produces

☑ Use API produce types

14. In the side bar, click Definitions to display the Definitions panel. Add a Definition by clicking the Add Definition icon ⊕.
15. Expand your new definition by clicking new-definition-1. For the Name field, enter `address`, and a Description of `The format of the address object`.
16. Using the same Definitions panel, configure the Properties definition according to the following table. Edit the default property and then create new properties by clicking Add Property and editing the default values.

Table 2. 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

This is an OpenAPI (Swagger 2.0) 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.

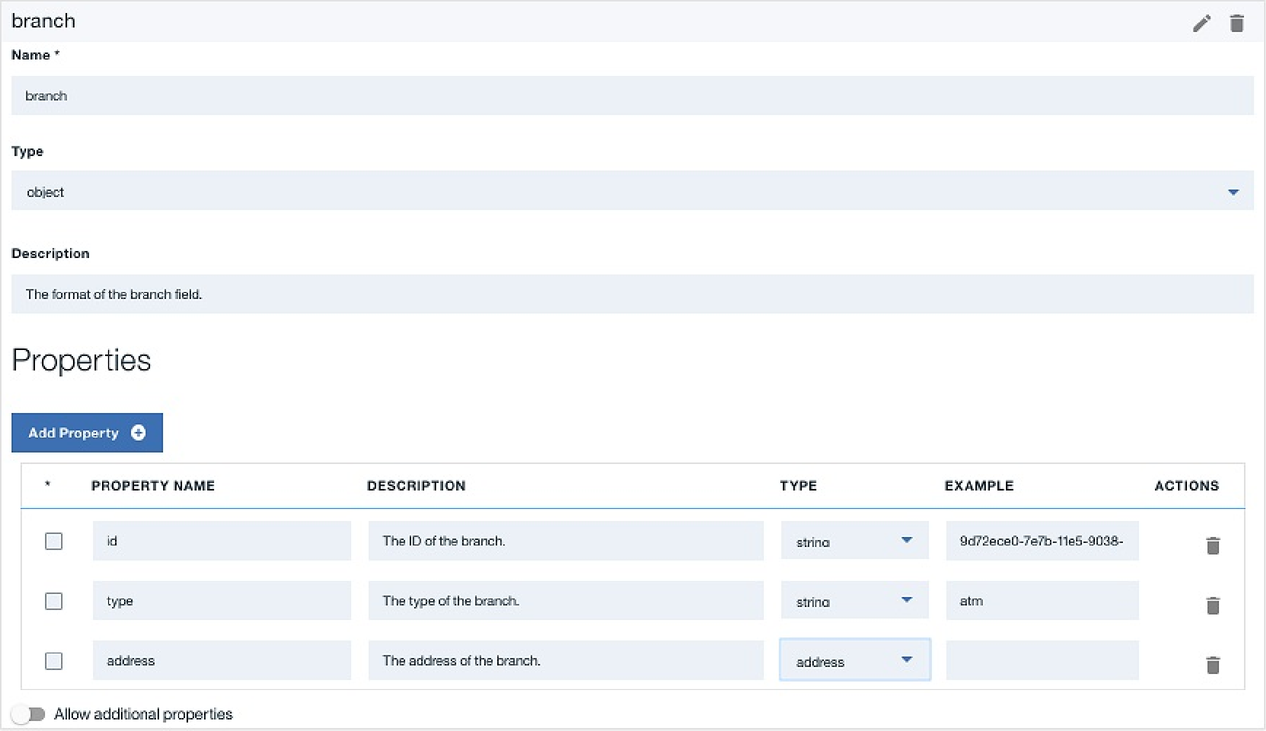
The Required The Required icon column indicates whether a property is required for success if a rest-validate policy uses the definition to perform validation. In this tutorial, no validation is performed and so none of your properties need to be marked as required.

17. Create a second definition by clicking the Add icon ⊕ in the Definitions panel.
18. Name the definition `branch` and, in the Description field, enter `The format of the branch field`.
19. 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 Property.

Table 3. 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`

Notice that for the address property, the type of the property references another definition within your API and the example is left blank. In this manner, you can create complex data structures.



branch

Name *

branch

Type

object

Description

The format of the branch field.

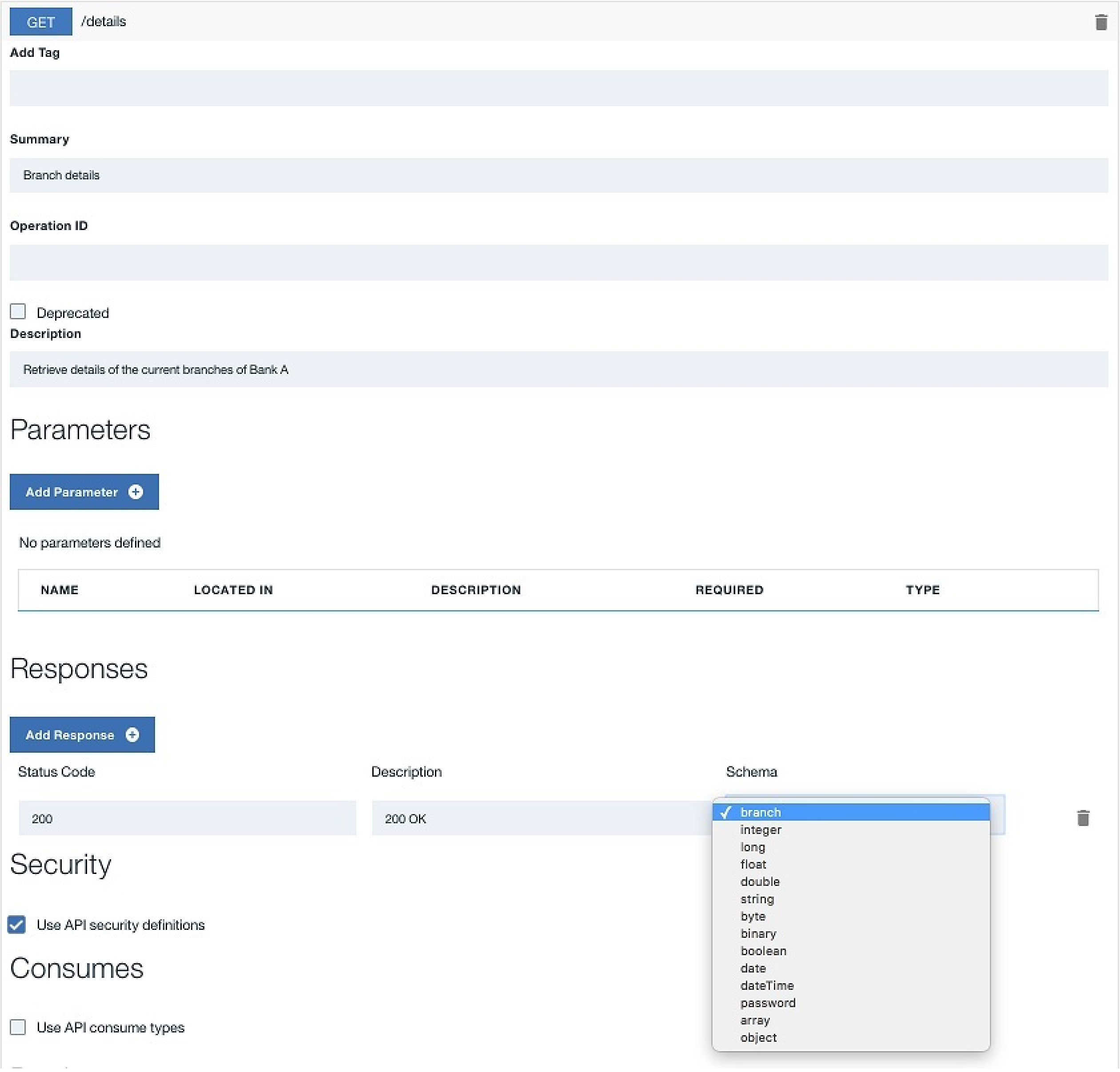
Properties

Add Property ⊕

*	PROPERTY NAME	DESCRIPTION	TYPE	EXAMPLE	ACTIONS
□	id	The ID of the branch.	string	9d72ece0-7e7b-11e5-9038-	🗑
□	type	The type of the branch.	string	atm	🗑
□	address	The address of the branch.	address		🗑

Allow additional properties

20. In the side bar, select Paths to display the Paths panel. For the /details Path, click GET to expand the available settings. Include the branch definition in the GET operation Status Code 200 response by clicking the Schema field and selecting branch from the drop-down list.



GET /details

Add Tag

Summary: Branch details

Operation ID

☐ Deprecated

Description: Retrieve details of the current branches of Bank A

Parameters

No parameters defined

NAME	LOCATED IN	DESCRIPTION	REQUIRED	TYPE

Responses

Status Code	Description	Schema
200	200 OK	✓ branch

Security

☑ Use API security definitions

Consumes

☐ Use API consume types

21. In the submenu navigation bar, click the Assemble tab to open the assemble view.
22. Access the invoke policy property sheet by clicking the invoke label.
23. Populate the Title, Description, and URL fields according to the following table. When called, your API now invokes the existing Branches API and uses its response. In this tutorial, no transformations are applied to the response of this API and so the entirety of the response is returned to the caller. You can see this response at https://apictutorials.mybluemix.net/branches.

Table 4. invoke fields

Field	Value
Title	`Branches Invoke`
Description	`Invoke an API to retrieve the status of all branches in the BankA system`
URL	`https://apictutorials.mybluemix.net/branches`

Leave the remaining fields with their default values.

24. Click the Save icon 💾 to save your changes.
25. Click the Source tab to view the OpenAPI (Swagger 2.0) definition of your API. All the configuration you have performed is included in this definition, either as part of the standard OpenAPI (Swagger 2.0) schema, or as part of the **`x-ibm-configuration`** extension.

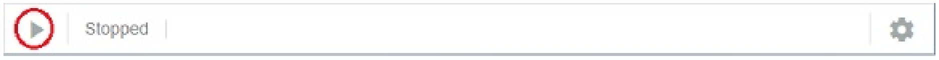
Your REST API is defined. This example helped you to configure the REST API invocation through the Assembly tool. No coding was required. The definitions help developers who are creating applications and integrating with the BankA Branches REST API for the first time.

Testing the REST API

Test your REST API to ensure that it is defined and implemented correctly.

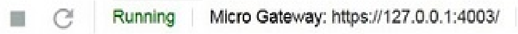
To test the REST API, complete the following steps:

1. Start the local test servers by completing the following steps:
   a. In the test console at the bottom of the screen, click the Start the servers icon:



Stopped

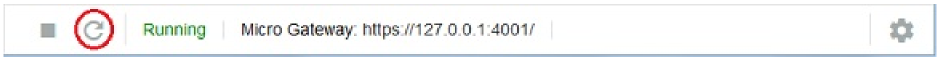
   b. Wait until the `Running` message is displayed:



Running | Micro Gateway: https://127.0.0.1:4003/

   Depending on your project configuration and whether other processes are running, a different port number might be displayed.

   Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking the Restart the servers icon



Running | Micro Gateway: https://127.0.0.1:4001/

   .
2. Click the Assemble tab.
3. Click the Test icon ▶. The test tool opens, overlaying the palette.
4. In the Operation section, use the drop-down menu to select the get /details operation.
5. At the bottom of the section, click Invoke. The operation is called by the test tool. The response of your API is shown in the test tool.

In other tutorials you get a chance to test the API by using the Developer Portal and API Manager test tools, both of which run online.

Creating a Product and a Plan for the REST API

Create a Product and a Plan so that you can later stage or publish your APIs.

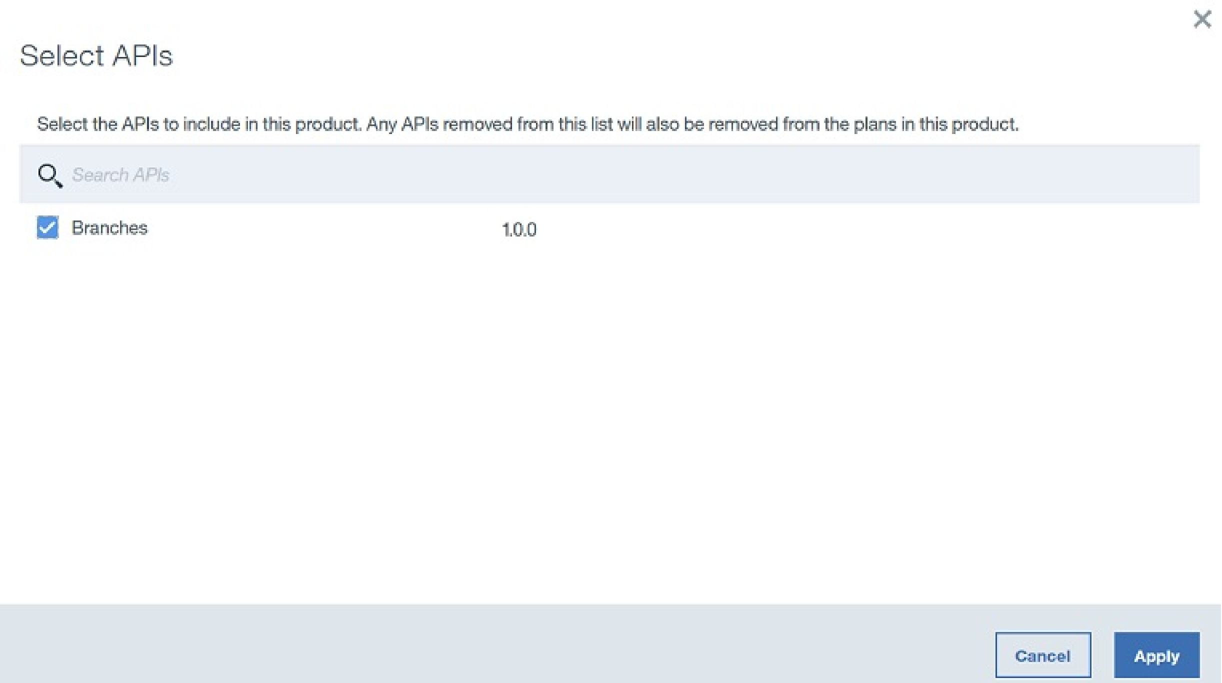
To create a Product, complete the following steps:

1. Click All APIs and then click the Products tab.
2. Click Add and then click New Product. The "New Product" window opens.
3. Complete the fields as shown in the following table and then click Create product.

Table 5. Product fields

Field Name	Value
Title	`Banking Services`
Name	`banking-services`
Version	`1.0.0`

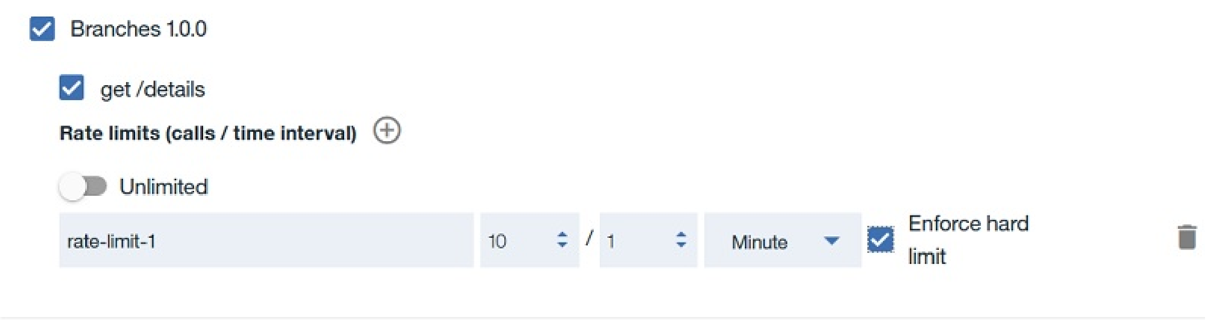
4. In the Visibility section you can control who the Product is visible to and who can subscribe to its Plans. The Product visibility is set to "Public" and so anybody will be able to see the Product when it is published to the Developer Portal. When published, the Plans can be subscribed to by "Authenticated users", which refers to users who have accounts in the Developer Portal.
5. In the APIs section, click the Add API icon ⊕. The Select APIs window opens.
6. Select the Branches API.



Select APIs

Select the APIs to include in this product. Any APIs removed from this list will also be removed from the plans in this product.

7. Click Apply.
8. Expand the Plan titled Default that has been automatically created. Because no APIs have been excluded, the only API in the Product is included in the Plan.
9. Enter `Basic` for the Title field and the Name field.
10. Add a rate limit to your /details operation by completing the following steps:
    a. Expand the Branches API.
    b. Click Override rate limit beside the get /details operation.
    c. Use the controls to set the rate limit as 10 requests per 1 minute against rate-limit-1, and select Enforce hard limit.



11. Click the Save icon 💾 to save your changes.

You have created the Banking Services Product with the Basic Plan within it, you added the Branches API to the Basic Plan, added a rate limit to the **`rate-limit-1`** operation, and staged your Banking Services Product to your development environment.
Note: These steps are not necessary to test your APIs offline, but a Product is needed when making your APIs externally available.

Publishing your Product

Publish your Product and the API it contains to make them externally available for later tutorials.

1. In the API Designer, click Publish and then click Add and Manage Targets.
2. Click Add a different target.
3. In the API Connect host address field, enter the address of your Management server, for example, `example.host.com`.
4. Provide a user name and password for an API Manager account on your server and then click Sign in.
5. In the Organization field, select the provider organization that you want to publish with.
6. Select Sandbox from the list of Catalogs. If you have a large number of Catalogs, use the Search field to refine the list of Catalogs.
7. Click Save.
8. Click Publish and then click your newly created target.
9. Select Select specific products and then select your Banking Services Product.
10. Click Publish. Your Product is now available through your gateway server and visible in both API Manager and the Developer Portal,

Your Product and the API it contains are published to your specified target.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a REST API definition.
- Tested a REST API.
- Created a Product that contains a Plan.
- Published a Product to a Catalog.

What to do next

- Discover and use your API in the Developer Portal.
- Secure your API with a client ID and secret.
- Secure your API by using an LDAP registry.
- Secure your API by using OAuth.

Related concepts

- Working with Products in the API Designer

Related tasks

- Creating API definitions
- Testing an API with the API Designer test tool

Related information

- IBM API Connect overview
- Creating APIs by using the API Designer
- Using the Developer Portal

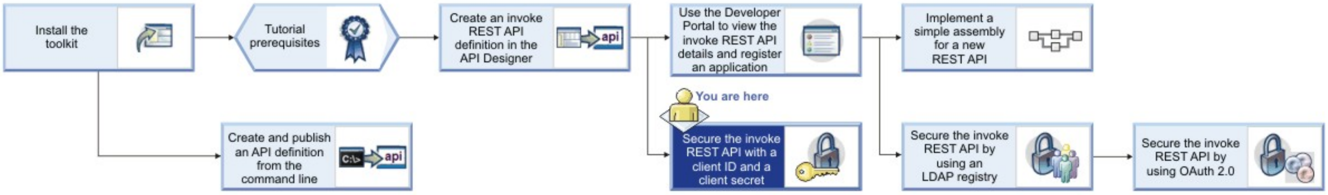
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Securing an API with a client ID and client secret

This tutorial shows you how to secure an API so that a calling application must supply a client ID and a client secret in IBM® API Connect Version 5.0.7 and later. This option is similar to requiring a user ID and password to be supplied.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.



Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
You are here
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

You will modify the security settings for the Branches API, which you created in the tutorial Tutorial: Creating an invoke REST API definition, so that a calling application must supply a client ID and a client secret, then you will attempt to call the Branches API with and without the client ID and client secret, to verify that the client ID and client secret are required.

You will complete the following lessons:

1. Setting the identification mechanism of an API
2. Calling an API by using a client ID and client secret

Setting the identification mechanism of an API

To modify the security settings for the Branches API so that a calling application must supply a client ID and client secret, complete the following steps:

1. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

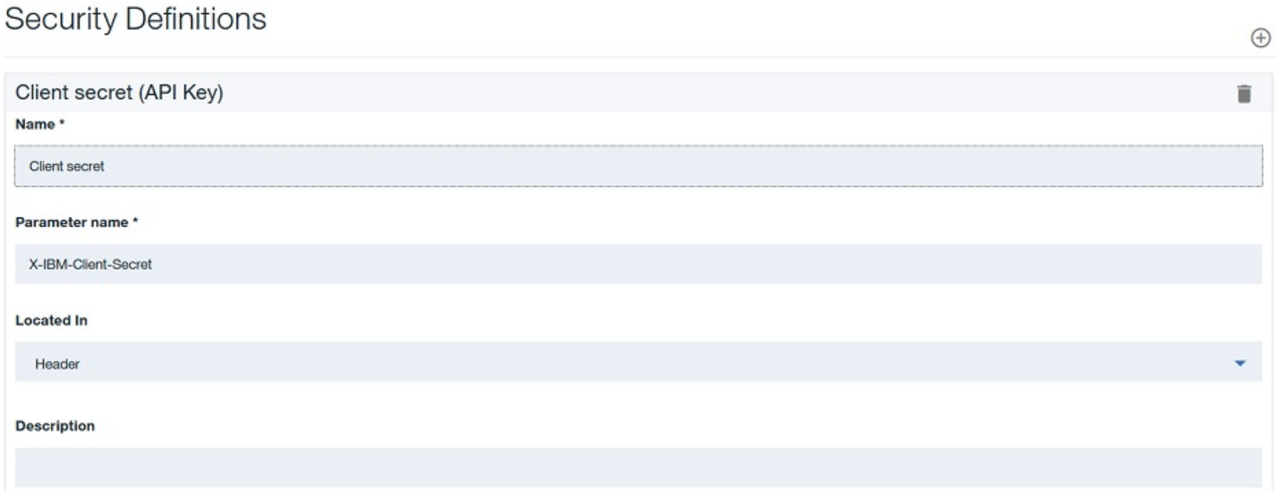
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
2. Click APIs, then click the Branches REST API that you created in the tutorial Tutorial: Creating an invoke REST API definition.
3. Navigate to the Security Definitions section.
4. Note that, by default, a Client ID security definition already exists for your API.
5. Click the Add Security Definition icon ⊕ in the Security Definitions section, then select API Key. A new API Key security definition displays in the Security Definitions section.
6. Change the value of the Name field to `Client secret`.
7. Leave the value of the Parameter Name field as X-IBM-Client-Secret. You have defined a new security scheme.



Security Definitions

Client secret (API Key)

Name *

Client secret

Parameter name *

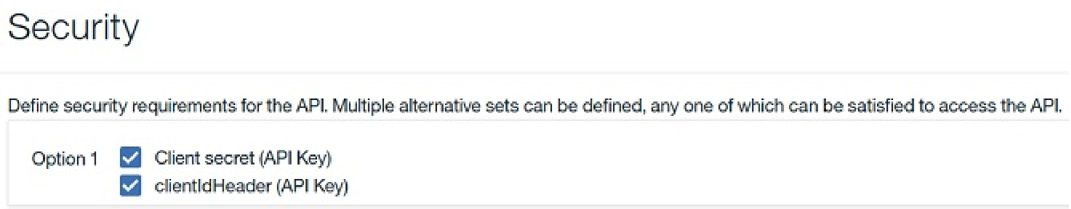
X-IBM-Client-Secret

Located In

Header

Description

8. Navigate to the Security section.
9. In the Security section, select Client secret (API Key), in addition to Client ID (API Key) which should already be selected by default.



Security

Define security requirements for the API. Multiple alternative sets can be defined, any one of which can be satisfied to access the API.

Option 1 ☑ Client secret (API Key)
☑ clientIdHeader (API Key)

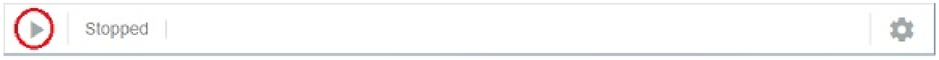
10. Click the Save icon 💾 to save your changes.

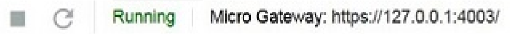
You have modified the operation so that a calling application must supply a client ID and client secret.

Calling an API by using a client ID and client secret

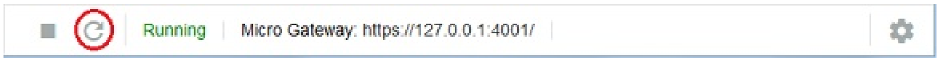
Now that you have determined the client ID and client secret for the Baggage Tracker application, you can supply them when calling the BankA API. For the purposes of this tutorial, you call the Branches API by testing it in the API Designer Explorer.

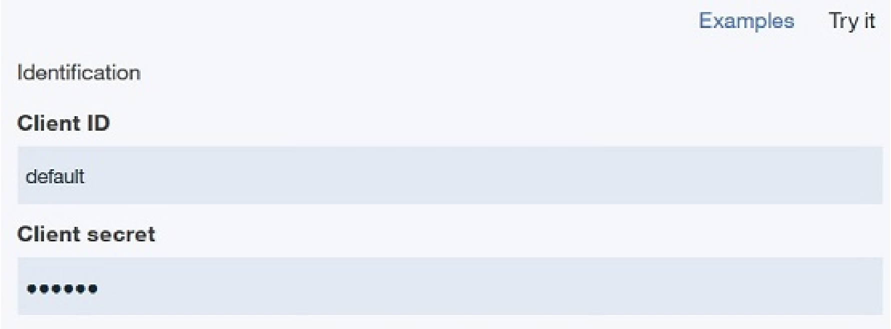
To call the Branches API by using a client ID and client secret, complete the following steps:

1. Start the local test servers by completing the following steps:
   a. In the test console at the bottom of the screen, click the Start the servers icon:
   

   b. Wait until the **`Running`** message is displayed:
   

   Depending on your project configuration and whether other processes are running, a different port number might be displayed.

   Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking the Restart the servers icon
   

   .
2. Click Explore, then click Try it.
3. Scroll down to the Identification section in the pane on the right.
   The Client ID field contains the value `default`, and the Client secret field contains the value `SECRET` in redacted form; these are the default values that are used for testing in the API Designer Explorer.
   
   Examples Try it

   Identification

   Client ID

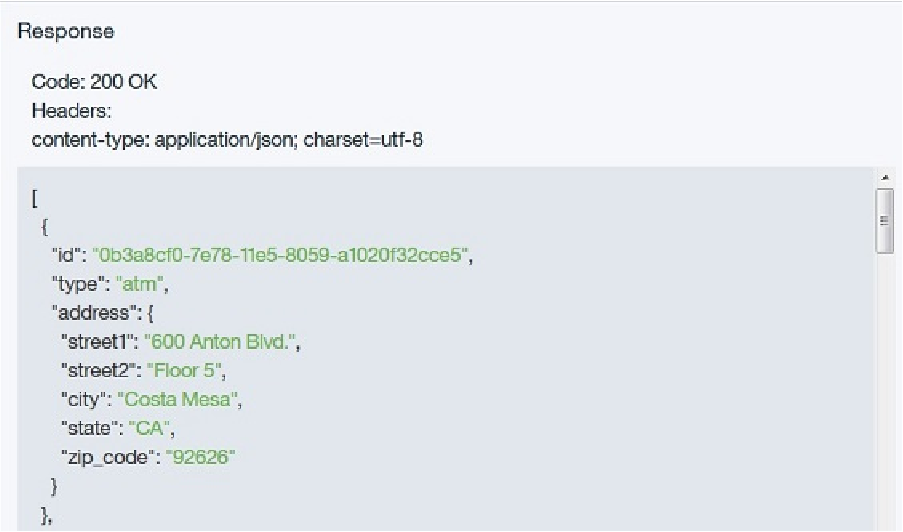
   default

   Client secret

   ••••••

   When the API is published and becomes available to application developers through the Developer Portal, the API will be called by using application specific client ID and client secret values; for more information, see Adding an application.

4. Remove the client ID and client secret values and click Call operation to test the API. The call fails.
5. Restore the client ID and client secret value by entering `default` in the Client ID field and `SECRET` in the Client secret field, and click Call operation to test the API. The Branches response is returned correctly:


Response

Code: 200 OK
Headers:
content-type: application/json; charset=utf-8

```
[
  {
    "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
    "type": "atm",
    "address": {
      "street1": "600 Anton Blvd.",
      "street2": "Floor 5",
      "city": "Costa Mesa",
      "state": "CA",
      "zip_code": "92626"
    }
  },
```

What you did in this tutorial

In this tutorial, you completed the following activities:

- Set the identification mechanism of an API.
- Called an API by using a client ID and client secret.

Related concepts

- Working with Products

Related information

- Creating APIs by using the API Designer
- Tutorial: Creating an invoke REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

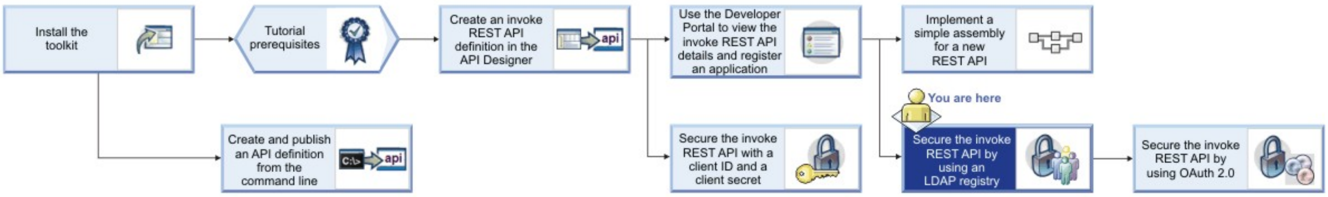
Tutorial: Securing APIs by using an LDAP user registry

This tutorial shows you how to secure an API with an LDAP registry so that LDAP credentials must be supplied when the API is called, in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial assumes that you have an available working LDAP registry in which you have a registered user ID, and to which you know the connection details.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
You are here
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

You will complete the following lessons:

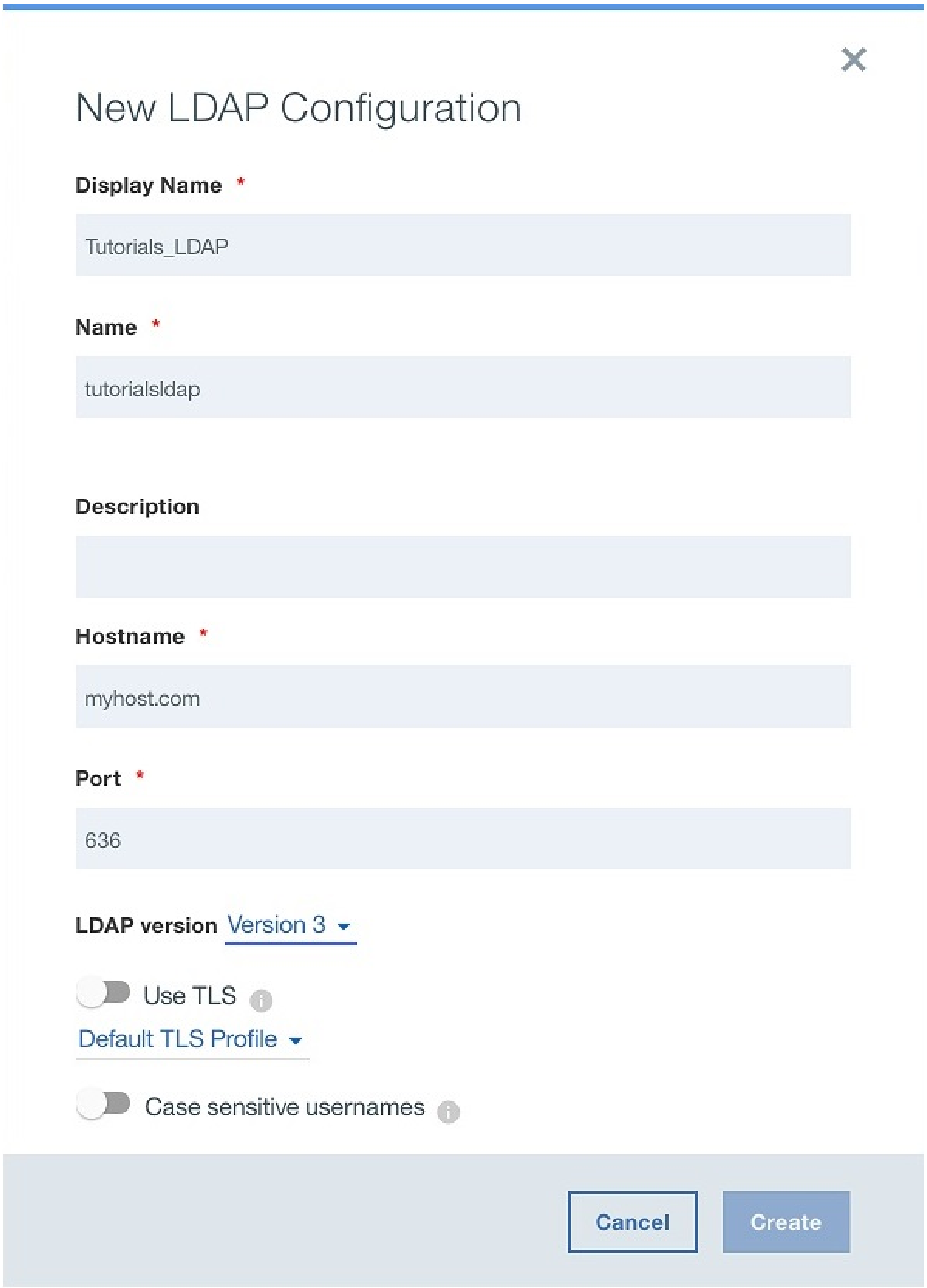
1. Creating an LDAP user registry definition in API Manager
2. Securing an API with an LDAP registry
3. Republishing a Product
4. Resubscribing to the Basic Plan
5. Testing a secured API in the Developer Portal

Creating an LDAP user registry definition in API Manager

To secure APIs with an LDAP registry, you must first define the connection details for the LDAP registry by creating an LDAP user registry definition in API Manager.

If an LDAP user registry definition has already been created, obtain the name of that LDAP user registry definition and proceed to Securing an API with an LDAP registry, otherwise complete the following steps.

1. Sign in to the API Manager user interface.
2. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
3. In the navigation pane, select Admin and select the Security tab.
4. Click User Registries.
5. Click Add > LDAP Registry. The New LDAP Configuration window opens.
6. In the New LDAP Configuration window, complete the following steps, as applicable for your LDAP registry:
   a. Enter `Tutorials_LDAP` in the Display Name field and `tutorialsldap` in the Name field and, optionally, enter a description.
   b. Enter the host name and port.
   c. Click the Version drop-down list and select the registry version.
   d. To protect user credentials, move the Use TLS slider to ON.
      Note:
      - If you want to use this option, TLS must be enabled on your LDAP server.
      - If the Use TLS slider is not set to ON, user credentials are not protected in transit to the LDAP user directory.
   e. Select Anonymous bind or Authenticated bind.
   f. If required, enter, in the Admin DN and Password text fields, the distinguished name and password of a user that has access to connect to the server and read all user and group data; for example, a built-in administrator account.
   g. In the Base DN text field, enter the Base DN information. If you are unsure of the correct value, click Test Bind & Get Base DN to display a drop-down menu of Base DNs available for the configured LDAP server.
   h. Enter the Prefix and Suffix information.
   i. When you are done, click Test configuration. If the test is successful, a confirmation message is displayed. If the test is not successful, an error message is displayed. Recheck your settings and run the test again.
   j. Click Create to create the registry. The following screen capture shows a sample configuration; you must ensure that you enter the details applicable to your LDAP registry:


New LDAP Configuration

Display Name *

Tutorials_LDAP

Name *

tutorialsldap

Description

Hostname *

myhost.com

Port *

636

LDAP version Version 3

Use TLS

Default TLS Profile

Case sensitive usernames

Note: You can view or edit the details of your LDAP registry configuration by completing the following steps:

1. In the navigation pane, select Admin and select the Security tab.
2. Click User Registries.
3. Click the manage icon ⋮ for the user registry that you want to edit, then click Edit user registry. The details of the registry are displayed.
4. Modify the configuration details as required, then click Save to save your changes.

Securing an API with an LDAP registry

You will use the API Designer user interface to secure the Branches API with the Tutorials_LDAP registry that you defined previously. The Branches API was created and configured in the tutorial Tutorial: Creating an invoke REST API definition.

1. In a command window, change to the project folder that you created in the tutorial Tutorial: Creating an invoke REST API definition.
2. Change directories to your LoopBack® project and enter the following command:

```
apic edit
```

After a brief pause, the console displays this message:

```
Express server listening on http://127.0.0.1:9000
```

API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
Note: If you need to run the editor on a different port, use the following command:
Linux Mac OS X

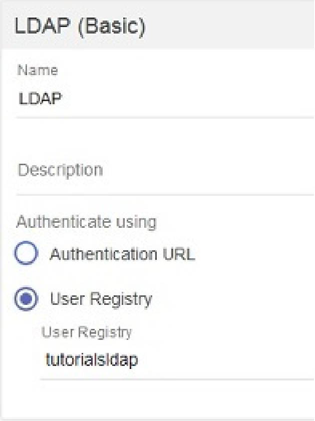
```
PORT=port_number apic edit
```

Windows

```
set PORT=port_number && apic edit
```

where *port_number* is the port number to use.

3. Click APIs, then click the Branches REST API that you created in the tutorial Tutorial: Creating an invoke REST API definition.
4. Navigate to the Security Definitions section.
5. In the Security Definitions section click the Add Security Definition icon ⊕.
6. Select Basic. A new Basic security definition displays in the Security Definitions section.
7. Change the value of Name to `LDAP`.
8. Select User Registry and, in the User Registry field, enter `tutorialsldap` to match the name that you gave to the LDAP registry you created in Creating an LDAP user registry definition in API Manager, or if an LDAP user registry definition has already been created enter the corresponding name.



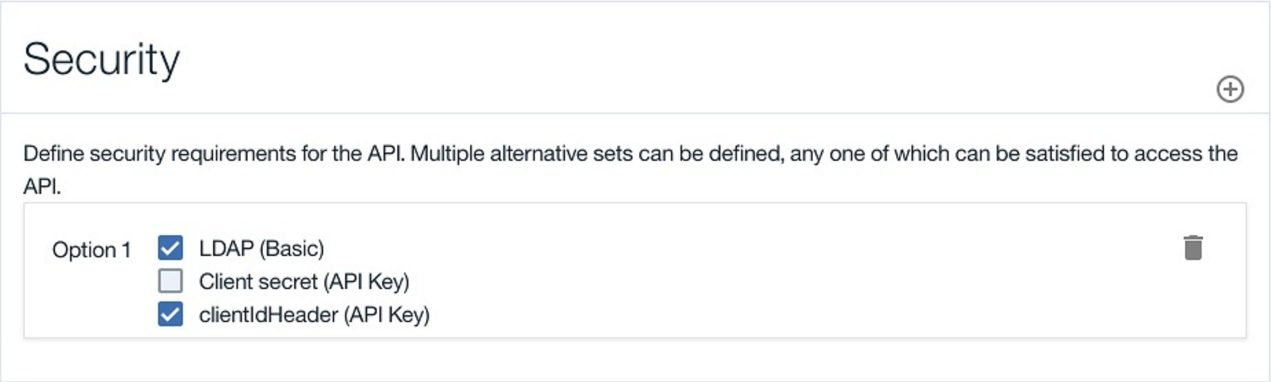
LDAP (Basic)

Name
LDAP

Description

Authenticate using
○ Authentication URL
● User Registry
User Registry
tutorialsldap

9. If you have previously completed the tutorial Tutorial: Securing an API with a client ID and client secret, clear the selection for the Client secret security definition.
10. Navigate to the Security section.
11. Select LDAP to apply the LDAP security definition to the API.

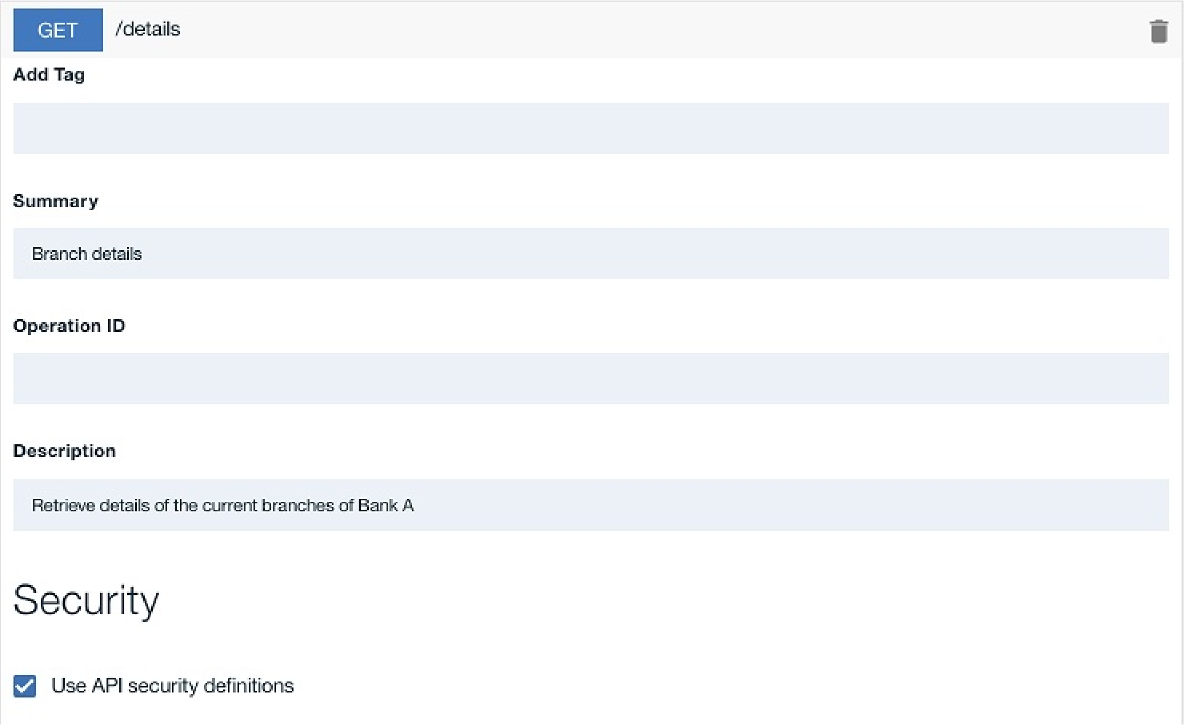


Security ⊕

Define security requirements for the API. Multiple alternative sets can be defined, any one of which can be satisfied to access the API.

Option 1
☑ LDAP (Basic)
☐ Client secret (API Key)
☑ clientIdHeader (API Key)

12. Navigate to the Paths section, click the GET method to expand its details, and ensure that Use API security definitions is selected; this setting means that all the security definitions that are defined for the API are inherited by the method.



Add Tag

Summary

Branch details

Operation ID

Description

Retrieve details of the current branches of Bank A

Security

☑ Use API security definitions

13. Click the Save icon 💾 to save your changes.

You have now secured the Branches API so that the user ID and password of a user in the Tutorials_LDAP registry must be supplied when the API is called.

Republishing a Product

For the security changes you have made to take effect, you must republish the Banking Services Product that contains the Branches API and Basic Plan.

To republish the Banking Services Product, complete the following steps:

1. Click ☁ Publish and then click the target that you created in the tutorial Tutorial: Creating an invoke REST API definition, for example Sandbox.
2. Select Select specific products and then select your Banking Services Product.
3. Click Publish.

Your Product and the API it contains are published to your specified target.

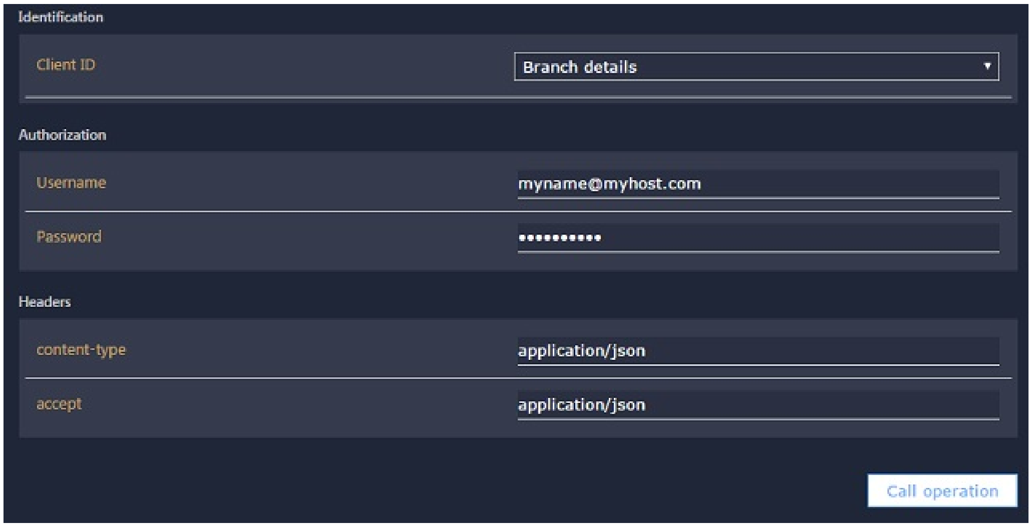
Resubscribing to the Basic Plan

Note: When a Product is republished to a non development Catalog, all subscriptions to the Plans in the Product are removed and you would need to resubscribe to the Basic Plan as described in the following steps. However, if you have published the Product to the Sandbox Catalog as described in these tutorials, you do not need to resubscribe to the Basic Plan, as subscriptions to a development Catalog automatically continue. You can therefore proceed directly to the Testing a secured API in the Developer Portal section.

1. Sign in to the Developer Portal.
2. Click API Products in the Developer Portal dashboard.
3. Click the Banking Services Product. The details of the Plan named Basic are displayed.
4. Click Subscribe.
5. Under the Application heading, select the Branch details radio button.
6. Click Subscribe.

Testing a secured API in the Developer Portal

1. In the Developer Portal, click API Products > Banking Services > Branches > GET /details. The GET /details operation is displayed.
2. In the "Try this operation" section of the console, use the drop-down menu to select the Branch details application.
3. Because you have secured the Branches API with an LDAP registry, an Authorization section is displayed. Enter your LDAP user name and password, then click Call operation.



Identification

Client ID: Branch details

Authorization

Username: myname@myhost.com

Password: ••••••••••

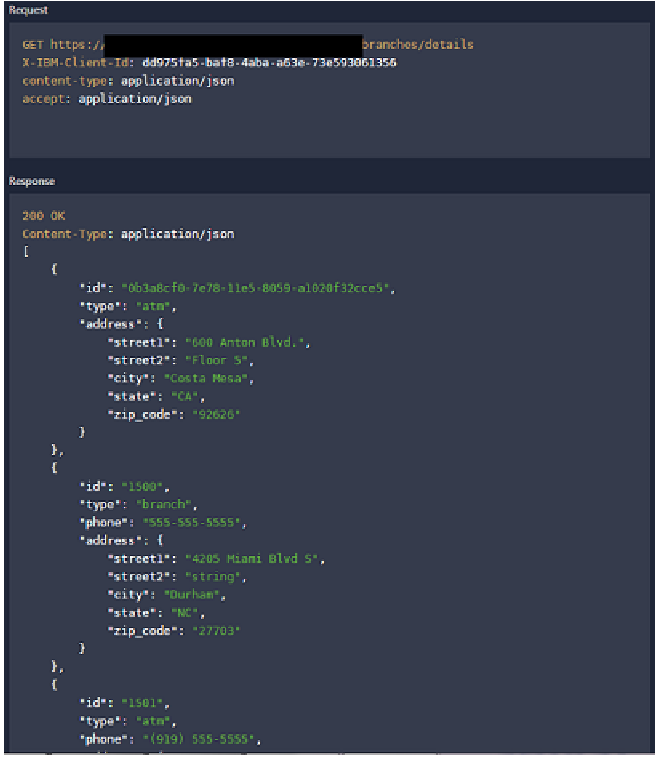
Headers

content-type: application/json

accept: application/json

Note: If you receive no response, 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.

4. A returned response of `200 OK` and the message body are displayed, indicating that the REST API operation call was successful.



```
GET https:/[redacted]branches/details
X-IBM-Client-Id: dd975fa5-baf8-4aba-a63e-73e593061356
content-type: application/json
accept: application/json
```

```
200 OK
Content-Type: application/json
[
    {
        "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": "1500",
        "type": "branch",
        "phone": "555-555-5555",
        "address": {
            "street1": "4205 Miami Blvd S",
            "street2": "string",
            "city": "Durham",
            "state": "NC",
            "zip_code": "27703"
        }
    },
    {
        "id": "1501",
        "type": "atm",
        "phone": "(919) 555-5555",
```

What you did in this tutorial

In this tutorial, you completed the following activities:

- Secured a Catalog with an LDAP registry.
- Secured an API with an LDAP registry.
- Republished a Product to a Catalog.
- Registered an application to call an API.
- Tested a secured API in the Developer Portal.

Related information

- IBM API Connect overview
- Using the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

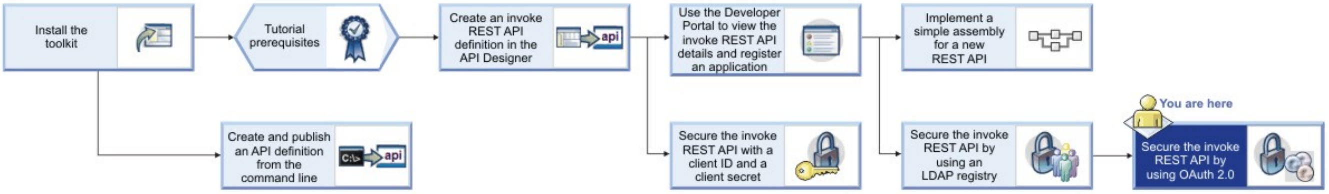
Tutorial: Securing an API by using OAuth 2.0

This tutorial shows you how to secure an API by using OAuth 2.0 so that an application can access the API on a user's behalf, in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
To complete this tutorial, you need an environment capable of sending HTTP requests and receiving HTTP responses. In these instructions, the **curl** command is used in a command line interface to demonstrate the OAuth flow without the need to write any application code. When you implement the OAuth flows for your application, make the same HTTP REST calls as used with **curl** in this tutorial.

Note: The commands and example commands in this tutorial might need to be adapted for the syntax of your own command line interface.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and client secret
Secure the invoke REST API by using an LDAP registry
You are here
Secure the invoke REST API by using OAuth 2.0

Note: You can secure your APIs with a third-party OAuth provider. For more information, see Integrating third party OAuth provider.

About this tutorial

You will modify the security settings for the Branches API, which you created in the tutorial Tutorial: Creating an invoke REST API definition, so that a calling application can make use of OAuth 2.0 to access the API on behalf of a user without requiring the user's password.

In this tutorial you will complete the following lessons:

- Choosing your OAuth Scheme
- Creating an OAuth 2.0 provider API
- Configuring the API Security Scheme
- Acquiring Access Tokens for Individual Schemes
- Using the Access Token

In OAuth 2.0, the following three parties are involved:

- The user, who possesses data that is accessed through the API and wants to allow the application to access it
- The application, which is to access the data through the API on the user's behalf
- The API, which controls and enables access to the user's data

Using OAuth 2.0, it is possible for the application to access the user's data without the disclosure of the user's credentials to the application.

The API will grant access only when it receives a valid access token from the application. How the application obtains an access token is dependent upon the OAuth scheme that is in use.

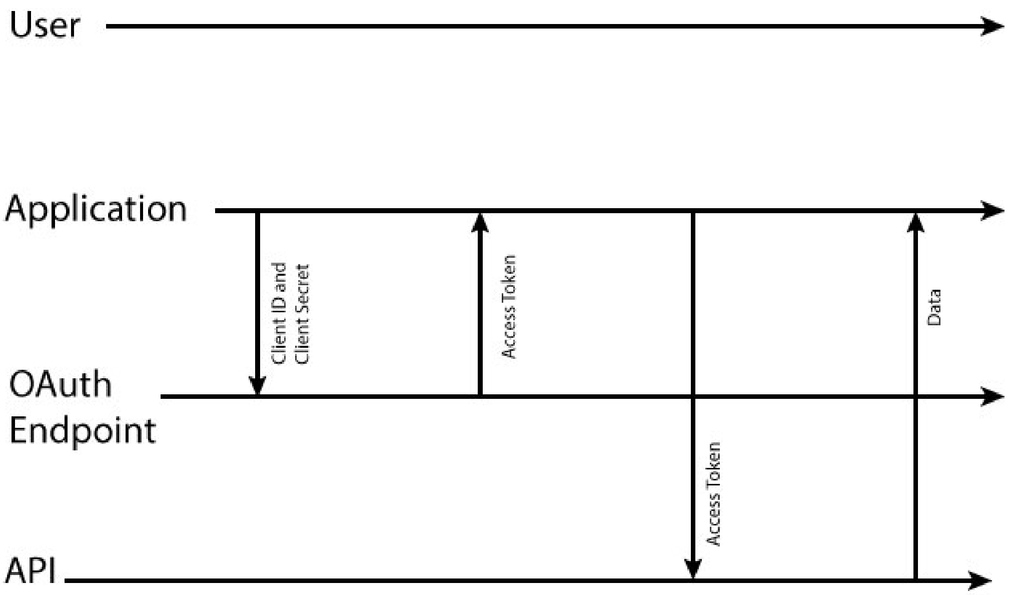
In this tutorial, you will be able to implement and test any of the following six OAuth schemes: implicit flow, application flow, confidential password flow, public password flow, confidential access code flow, and public access code flow. More information about these schemes is available in the following section.

Choosing your OAuth scheme

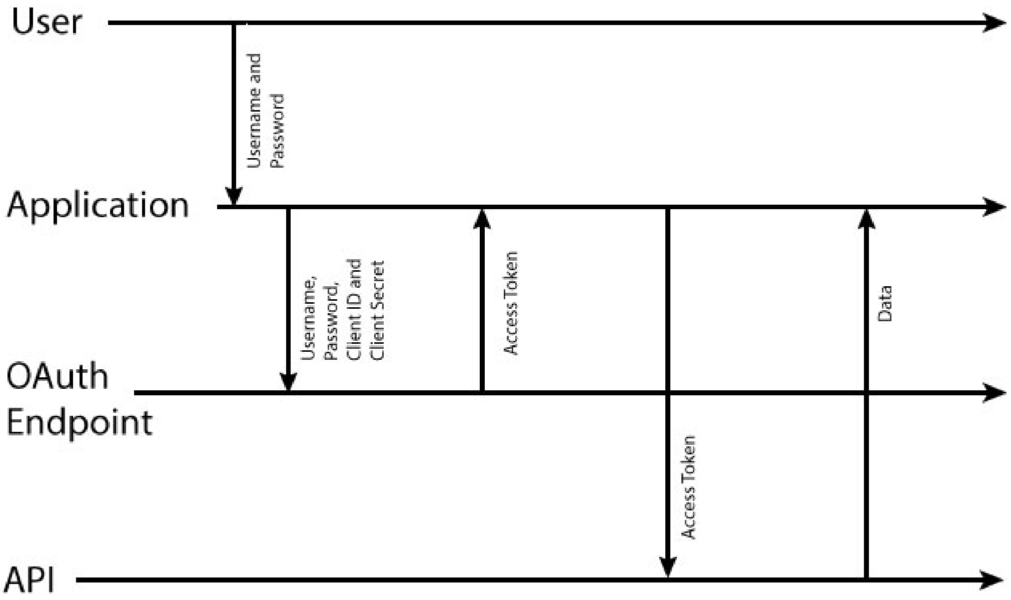
If you already know which OAuth scheme you intend to use, skip this section and proceed to Creating an OAuth 2.0 provider API.

To choose an OAuth scheme, you must first establish whether your implementation is considered public or confidential. This will narrow your choices to three schemes. A brief outline of each scheme and the characteristics of the three public and three confidential schemes follows:

- Confidential
  A confidential scheme is suitable when an application is capable of maintaining the secrecy of the client secret. This is usually the case when an application runs in a browser and accesses its own server when obtaining OAuth access tokens. As such, these schemes make use of the client secret.
  - Application flow
    In the application flow scheme, the user is not required to provide authorization at any stage. Instead, the application uses its client secret to obtain an access token. In this case, it is critical that the client secret is kept safe.

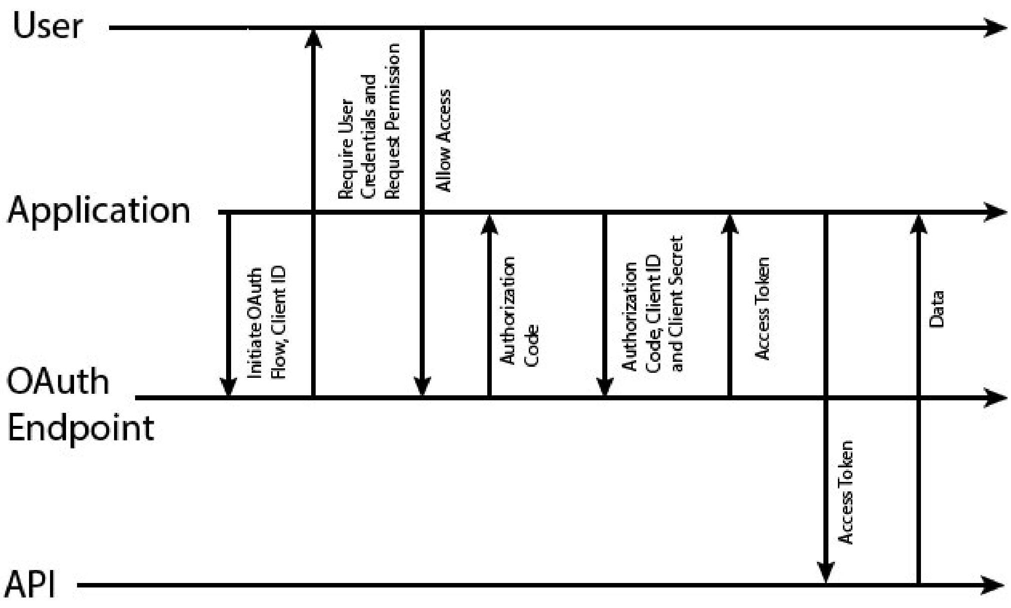


  - Password flow
    In the password flow scheme, the user provides the application with a user name and password that can be used to access the user's data. Following this, the client will directly contact the provider API to request an access token. In this case, trust must exist between user and application because the user's password is revealed to the application. However, this still has an advantage over the application using the password directly, because the validity of the access token or client ID can later be revoked without impacting other applications that do not need their access revoked. However, the application must be trusted to not store the user name and password.



  - Access code flow

In the access code flow, the application has the user provide authorization through a form provided by the gateway server, which, if they grant authorization, provides an authorization code to the application. The application sends the authorization code to the provider API and is granted an access token in return.

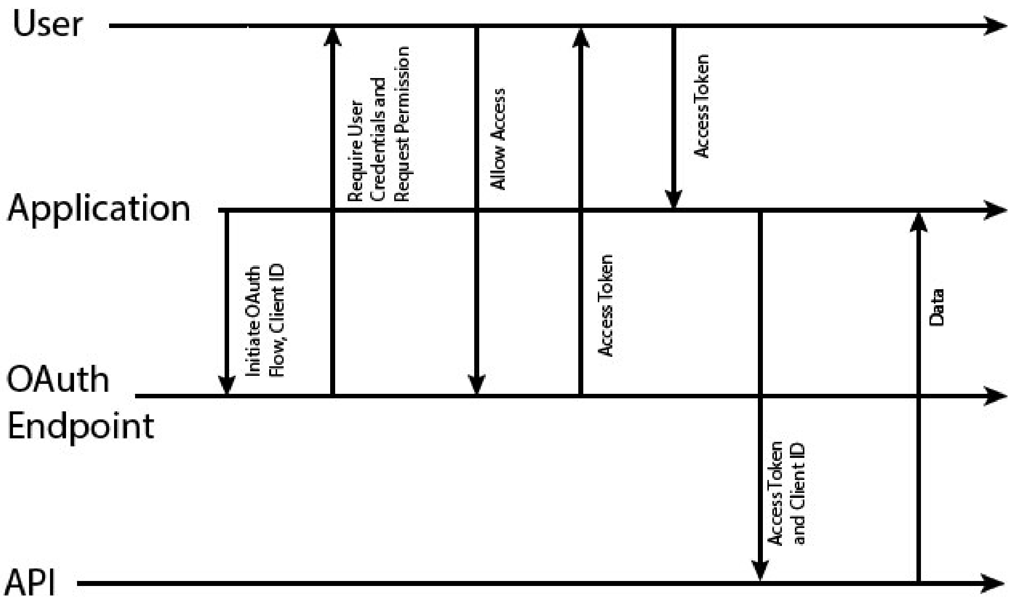

User
Application
OAuth Endpoint
API
Initiate OAuth Flow, Client ID
Require User Credentials and Request Permission
Allow Access
Authorization Code
Authorization Code, Client ID and Client Secret
Access Token
Access Token
Data

- Public

  A public scheme is suitable when an application is incapable of maintaining the secrecy of the client secret. This is usually the case when the application is native on a computer or mobile where the secret would have to be stored on the user's device, likely inside the source code of the application. As such, these schemes do not make use of the client secret.

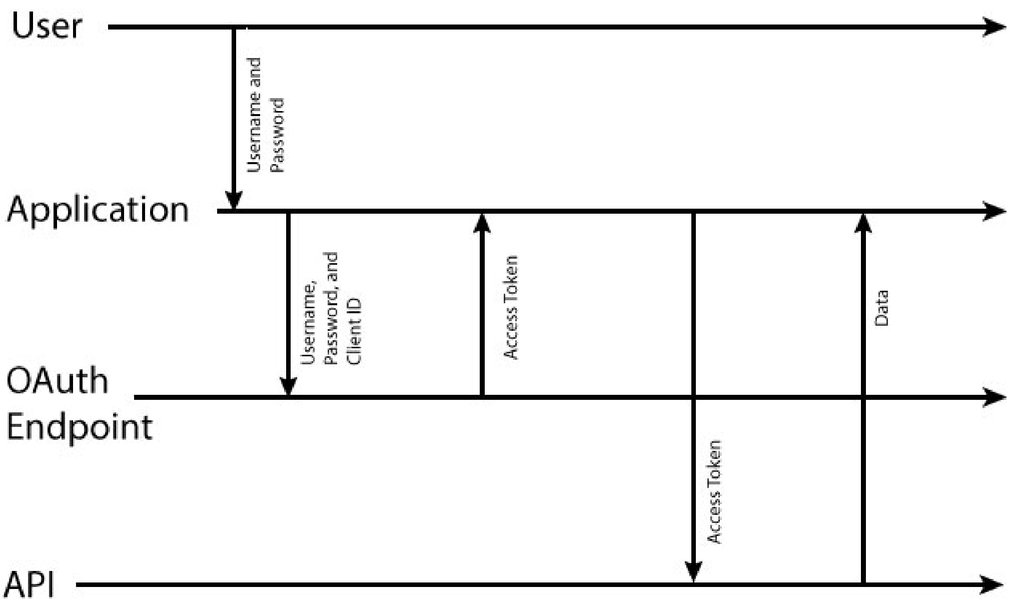
  - Implicit flow

    In the implicit flow scheme, the application requests an access token from the gateway server and the user grants permission, at which point an access token is provided to the user, who must then pass the token to the application

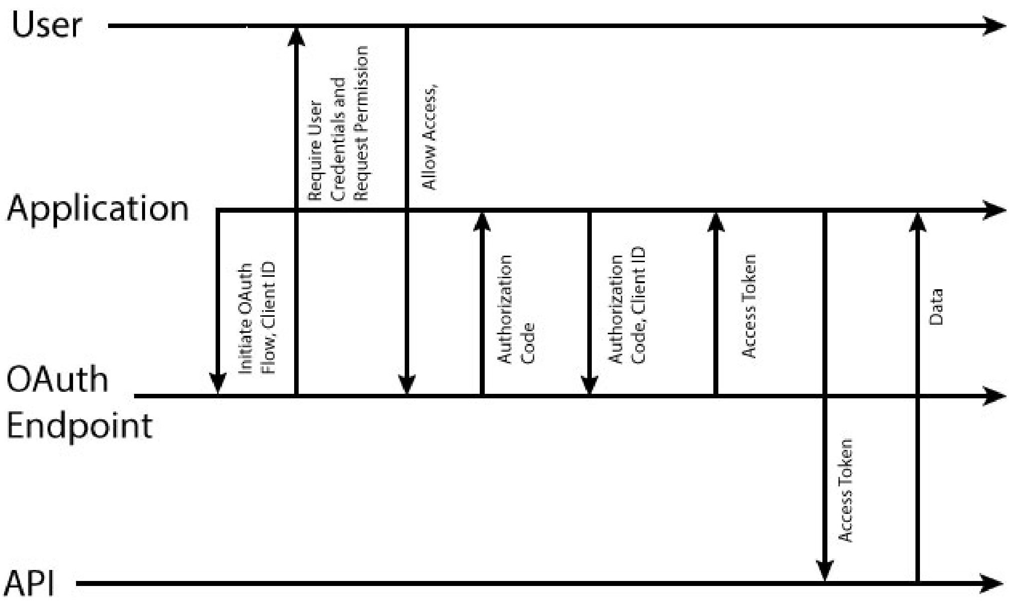

User
Application
OAuth Endpoint
API
Initiate OAuth Flow, Client ID
Require User Credentials and Request Permission
Allow Access
Access Token
Access Token
Access Token and Client ID
Data

  - Password flow

    In the password flow scheme, the user provides the application with a user name and password that can be used to access the user's data. Following this, the client will directly contact the server to request an access token. In this case, trust must exist between user and application because the user's password is revealed to the application. However, this still has an advantage over the application using the password directly, because the validity of the access token or client ID can later be revoked without impacting other applications that do not need their access revoked. However, the application must be trusted to not store the user name and password.



- Access code flow
  In the access code flow, the application has the user provide authorization through a form provided by the gateway server, which, if they grant authorization, provides an authorization code to the application. The application sends the authorization code to the provider API and is granted an access token in return.


User
Require User Credentials and Request Permission
Allow Access,
Application
Initiate OAuth Flow, Client ID
Authorization Code
Authorization Code, Client ID
Access Token
Data
OAuth Endpoint
Access Token
API

Creating an OAuth 2.0 provider API

To create an OAuth 2.0 provider API, complete the following steps:

1. In a command window, change to the project folder that you created in the tutorial Tutorial: Creating an invoke REST API definition.
2. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.

Note: If you need to run the editor on a different port, use the following command:

Linux Mac OS X

```
PORT=port_number apic edit
```

Windows

```
set PORT=port_number && apic edit
```

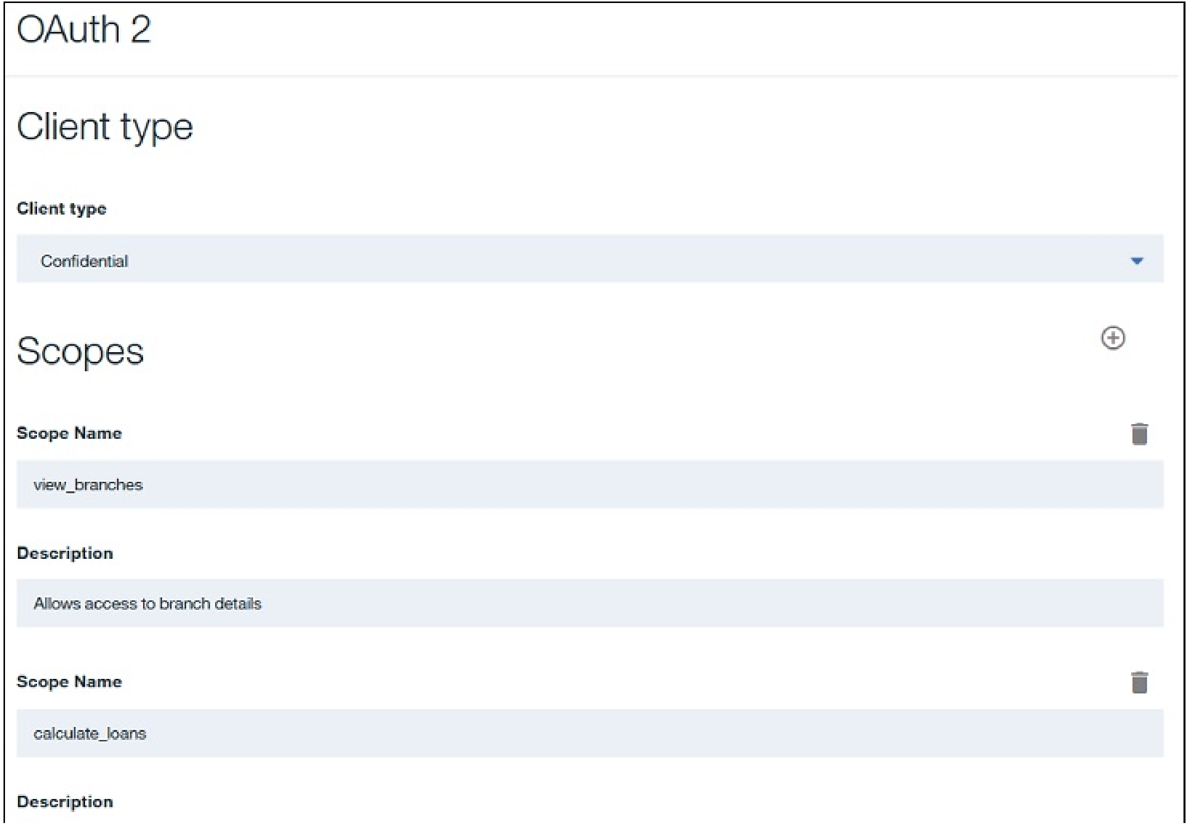
where *port_number* is the port number to use.

3. In the API Designer, click the APIs tab.
4. Click Add > OAuth 2.0 Provider API.
5. Complete the fields according to the following table:

Table 1.

Field	Contents
Title	OAuth Endpoint API
Name	oauth-endpoint-api
Version	1.0.0
Base Path	/oauth-end

6. Click Create API.
7. In the OAuth 2 section, configure the OAuth settings of your provider API.
   a. Depending on your chosen scheme, select Public or Confidential in the Client type field.
   b. Rename Scope 1 to `view_branches` by using the text field.
   c. In the Description field for view_branches, enter `Allows access to branch details`.
   d. Rename Scope 2 to `calculate_loans` by using the text field.
   e. In the Description field for calculate_loans, enter `Allows use of the loan calculator`.
   f. Delete Scope 3 by clicking the Remove scope icon 🗑 .
   g. In the Grants section, clear the check box of any grant types you do not want to use.
   Note: You must clear the check box of Implicit if you selected Confidential in step 7.a and you must clear the check box of Application if you selected Public in step 7.a.
   h. In the Collect credentials using field of Identity extraction, select Basic.
   i. In the Authenticate application users using field of Authentication, select User registry and, in the User registry field, enter `tutorialsldap` to reference the LDAP registry that you created in Tutorial: Securing APIs by using an LDAP user registry.
   j. In the Authorize application users using field of Authorization, select Default form.
   k. Ensure that the Enable refresh tokens and Enable revocation switches of Tokens are in the Off position.



OAuth 2

Client type

Client type: Confidential

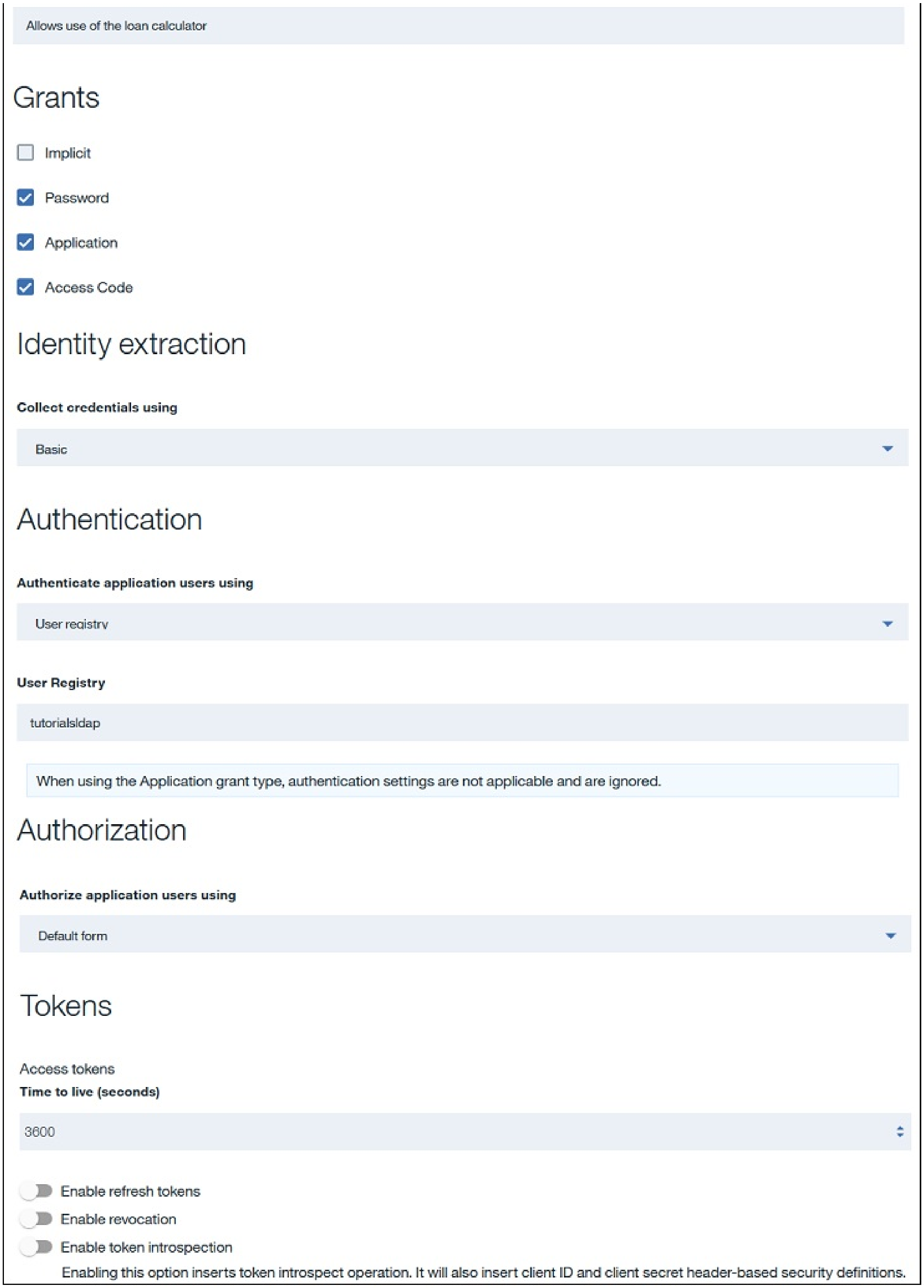
Scopes

Scope Name: view_branches

Description: Allows access to branch details

Scope Name: calculate_loans

Description



Allows use of the loan calculator

Grants

☐ Implicit

☑ Password

☑ Application

☑ Access Code

Identity extraction

Collect credentials using

Basic

Authentication

Authenticate application users using

User registry

User Registry

tutorialsldap

When using the Application grant type, authentication settings are not applicable and are ignored.

Authorization

Authorize application users using

Default form

Tokens

Access tokens

Time to live (seconds)

3600

○ Enable refresh tokens

○ Enable revocation

○ Enable token introspection

Enabling this option inserts token introspect operation. It will also insert client ID and client secret header-based security definitions.

8. Click the Save icon 💾 to save your changes.

Configuring the API security definition

To enable your chosen authentication scheme in API Designer, complete the following steps:

1. In the API Designer, click the APIs tab.
2. Click your Branches API definition.
3. In the Security Definitions section, click the Add Security Definition icon ⊕ and then click OAuth.
4. Scroll down to your newly created OAuth security definition.
5. In the Name field, rename your security definition as `OAuth definition`.
6. In the Flow field, select the type of flow you want to use.
7. In the Scopes section click the Add scope icon ⊕.
8. In the Scope Name field, rename the scope to `view_branches`.
9. If you are using a flow that uses an authorization URL, in the Authorization URL field, enter the following URL:

   ```
   https://Host_Address/Org_Segment/sb/oauth-end/oauth2/authorize
   ```

   where:
   - *Host_Address* is the address of your gateway host.
   - *Org_Segment* is the path segment of the organization that you want to use.

   Note: If you do not know your host address or organization segment, you can find them by completing the following steps:
   a. Log in to the API Manager user interface.
   b. From the Dashboard page, click the Sandbox Catalog to display its details.
   c. Click the Settings tab, then click Gateways. You can obtain the host address and organization segment from the displayed Base URL value.
10. If you are using a flow that uses an authentication URL, in the Token URL field, enter the following URL:

    ```
    https://Host_Address/Org_Segment/sb/oauth-end/oauth2/token
    ```

    where:
    - *Host_Address* is the address of your gateway host.
    - *Org_Segment* is the path segment of the organization that you want to use.

    Note: If you do not know your host address or organization segment, you can find them by completing the following steps:
    a. Log in to the API Manager user interface.
    b. From the Dashboard page, click the Sandbox Catalog to display its details.
    c. Click the Settings tab, then click Gateways. You can obtain the host address and organization segment from the displayed Base URL value.
11. In the Security section, select the check box for your OAuth definition and clear the check box for your LDAP (Basic) security definition.
    Note: You should have only your OAuth definition and clientID security definitions selected, together with the view_branches scope.
12. Click the Save icon 💾 to save your changes.
13. Click All APIs and then click the Products tab.
14. Click your Banking Services Product.
15. In the APIs section, click the Add API icon ⊕.
16. Select Branches and OAuth Endpoint API and then click Apply.
17. Click your Basic Plan to expand it and ensure that Branches and OAuth Endpoint API are selected.
18. Click the Save icon 💾 to save your changes.
19. Publish the Banking Services Product.
    a. Click Publish and then click the Sandbox Catalog for the host address and organization that you want to use.
    b. Select Select specific products and then select your Banking Services Product.
    c. Click Publish.

Acquiring access tokens for individual schemes

In this section you will acquire an access token to access your API by using OAuth 2.0. The different schemes each use a different method to acquire an access token.

1. In your Developer Portal, sign in to your developer account.
   Note: An administrator account cannot register applications and therefore cannot be used in this tutorial.
2. Click Apps
3. Click Create new App.
4. In the Title field, enter `OAuth Application`.
5. In the OAuth Redirect URI field, enter `https://example.com/redirect` and then click Submit.
6. Click Show Client Secret and click Show beside the Client ID field and then record your application's client secret and your application's client ID.
   Note: The Client Secret can be viewed only once; if you have viewed it before and not recorded it you will need to reset it, which will invalidate uses of the previous secret elsewhere.
7. Click API Products and then click your Banking Services Product.
8. For your Basic plan, click Subscribe.

9. Select your OAuth Application and then click Subscribe.
10. Acquire the access token. The process for obtaining an access token differs for each scheme. Click the link for your chosen scheme:
    - Confidential
        - Application flow
        - Password flow
        - Access code
    - Public
        - Implicit flow
        - Password flow
        - Access code

Using the access token

This section is common to all the OAuth schemes. You will access the Branches API and be authenticated.

1. From the API page of your Branches API in the Developer Portal, select the GET /branches/details operation.
2. Record the operation URL that is displayed.
3. Enter the following command into your command line interface (as one line):

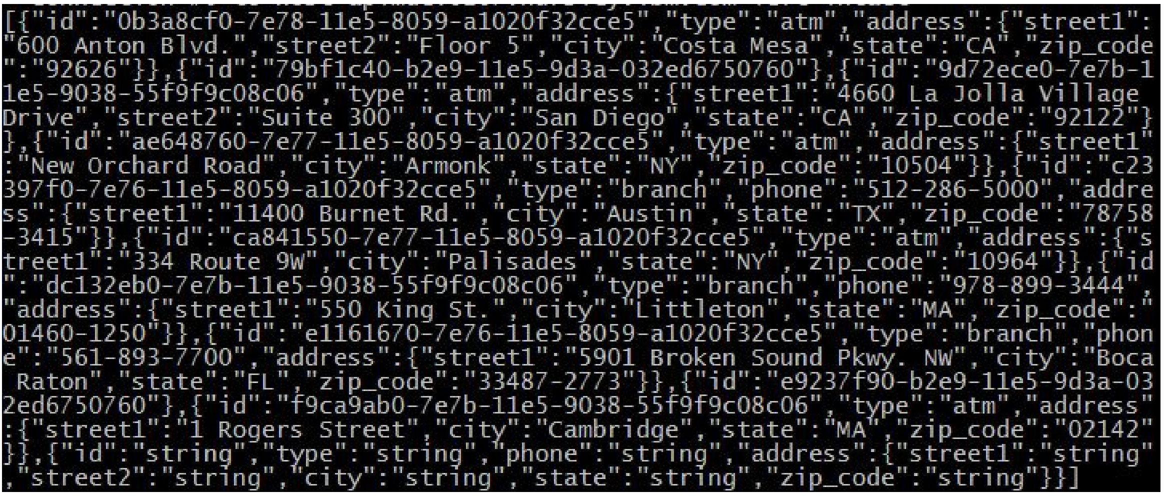
   ```
   curl -k -v -H "X-IBM-Client-Id: Client_ID" -H "Authorization: Bearer Access_Token" -X GET
   'Operation_URL'
   ```

   where:
    - *Client_ID* is as recorded in step 6 of the Acquiring access tokens for individual schemes section of this tutorial.
    - *Access_Token* is the access token that you recorded at the end of the scheme-specific section of this tutorial.
    - *Operation_URL* is the URL you would use to call the operation if it were to be unsecured, as recorded in step 2.

   For example:

   ```
   curl -k -v -H "X-IBM-Client-Id: 6e3f115d-5220-48bd-a019-3c9680e657b7"
    -H "Authorization: Bearer AAEkNmUzZjExNWQtNTIyMC00OGJkLWEwMTktM2M5Nj
   gwZTY1N2I3_3QEMOL24OpXJTnuA4y0qvqM_7HgX7ImJ0Uyl_Jcq8Xx8CRHTBtPoBukueb
   fgDK3BtZjId7_Yeyjd01vqx8Xl0ORdFxC0BF-grZLCG1BeaY"
   -X GET 'https://host.com/myorg/sb/branches/details'
   ```

   The response includes the data from the operation.

   ```
   [{"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":"79bf1c40-b2e9-11e5-9d3a-032ed6750760"},{"id":"9d72ece0-7e7b-1
   1e5-9038-55f9f9c08c06","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":"c23
   397f0-7e76-11e5-8059-a1020f32cce5","type":"branch","phone":"512-286-5000","addre
   ss":{"street1":"11400 Burnet Rd.","city":"Austin","state":"TX","zip_code":"78758
   -3415"}},{"id":"ca841550-7e77-11e5-8059-a1020f32cce5","type":"atm","address":{"s
   treet1":"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","phon
   e":"561-893-7700","address":{"street1":"5901 Broken Sound Pkwy. NW","city":"Boca
    Raton","state":"FL","zip_code":"33487-2773"}},{"id":"e9237f90-b2e9-11e5-9d3a-03
   2ed6750760"},{"id":"f9ca9ab0-7e7b-11e5-9038-55f9f9c08c06","type":"atm","address"
   :{"street1":"1 Rogers Street","city":"Cambridge","state":"MA","zip_code":"02142"
   }},{"id":"string","type":"string","phone":"string","address":{"street1":"string"
   ,"street2":"string","city":"string","state":"string","zip_code":"string"}}]
   ```

   .
4. To verify that access is granted only for the correct user ID and the correct access token, alter one or both in the previous command.

What you did in this tutorial

In this tutorial you have:

- Chosen your OAuth Scheme.
- Configured the API Security Scheme.
- Acquired an access token for your chosen scheme.
- Used the access token.

What to do next

- When acquiring an access token, use the `calculate_loans` scope, instead of the `view_branches` scope, to verify that a code requested for only the scope specified by the secured API can be used to access the API.
- Repeat the tutorial for a different OAuth flow.

Related concepts

- OAuth user scenario
- Tokens

Related tasks

- Configuring API security by using the API Designer
- Protecting an API with OAuth security definition

Related reference

- OAuth revocation URL

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for application flow

This tutorial shows you how to acquire an access token for the OAuth scheme *application flow* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the application flow scheme, which is only valid when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

   ```
   curl -v -u Client_ID:Client_Secret -k -X POST -d {}
   'Token_Endpoint_URL?grant_type=client_credentials&scope=Scope'
   ```

   Note: On Windows systems, you must modify the **curl** command to place quotation marks around the curly brace characters to change them from `{}` to `"{}"`.
   Where:
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Token_Endpoint_URL* is as recorded in step 2.

   Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
   For example:

```
curl -v -u 097a4830-eeb7-4f6e-ad4b-e507313a771e:lQ8hT1vA1mJ4qL0eL7nW0xL7wL5gE4hF8aL
6wH5fW0rB8rW3iD -k -X POST -d {} 'https://host.com/myorg/sb/oauth-end/oauth2/token?
grant_type=client_credentials&scope=view_branches'
```

A message is returned, of the following form:

```
{"token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFYy1hbGwiOugk49sRso0D1yKfD
i4Uny_W589WWa_Ea3eBG6ZbWwh-BT2zawnHK8sKM1EBUuAGTcyYh-n7sATyWi3-ElH8QBWqeadLE0h5c
JcsUxMHa_65VM_tI8KnnNphi7CIxx0NJRuMbCE8uOHRIPCmNon3", "expires_in":3600, "scope":"view_branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for password flow (confidential)

This tutorial shows you how to acquire an access token for the OAuth scheme *password flow (confidential)* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the password flow scheme when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

   ```
   curl -v -u Client_ID:Client_Secret -k -X POST -d
   'grant_type=password&scope=Scope&username=User_Name&
   password=Password'  'Token_Endpoint_URL'
   ```

   Note: On Windows systems, you must modify the **curl** command to place quotation marks around the curly brace characters to change them from `{}` to `"{}"`.
   where:
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Token_Endpoint_URL* is as recorded in step 2.
   - *User_Name* is a valid user name from your LDAP registry.
   - *Password* is the password that corresponds to *User_name* in your LDAP registry.

Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
For example:

```
curl -v -u 51dfaed9-9d07-46fc-83a6-52c09735a4a0:W3xB2tK0uP7vC8uX0rO8t
P5gE0yD7aV8eE3wN0jG1vY7aT1tL2 -k -X POST -d 'grant_type=password&scope=view_branches&
username=USER@example.com&password=PASSWORD'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

A message is returned, of the following form:

```
{ "token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFYy1hbGxug39ky5-MaadRBFgRB4Cn
hzClFNbidoP_UGs56vyTy6r-TbdqOLXuIaLQz8SKLdteXqN3VutqOnSkiB50fm9h6nvodXYrHkiLwGE9Sas
IzJImtTFuNNEdE03WL1BbLKnhNM_I0DxBoETUT0yIfNJToJJwnV4_yTaPuMwuL6ZjbQ", "expires_in":3600,
"scope":"view_branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for access code flow (confidential)

This tutorial shows you how to acquire an access token for the OAuth scheme *access code flow* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the access code flow scheme when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL and Authorization Endpoint URL values.
3. In your web browser, enter the following URL:

   ```
   Authorization_Endpoint_URL?response_type=code&
   redirect_uri=Redirect_URI&scope=Scope&
   client_id=Client_ID
   ```

   where:
   - *Authorization_Endpoint_URL* is as recorded in step 2.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

For example:

```
https://host.com/myorg/sb/oauth-end/oauth2/authorize?
response_type=code&redirect_uri=https://example.com/redirect&scope=view_branches&
client_id=51dfaed9-9d07-46fc-83a6-52c09735a4a0
```

4. When prompted, authenticate using the LDAP registry you specified when creating your API.
   You are presented with a prompt asking you to confirm that you want to grant your OAuth Application access to your Branches API.

5. Click Allow Access.
   You are redirected to your Redirect URI, which will have additional information appended. This URL takes the following form:

   ```
   Redirect_URI?code=Authorization_Code
   ```

   For example:

   ```
   https://example.com/redirect?code=AAINXmK3Qp91jot1nqrqPDgrhpDJNOD
   1oywxOHeJ2sdYfOLLcwZZOa82tvUDpknWtfdQe1s59DYGOnR9XQn6jJg7D4HPf2HGsofwkvZPJH
   brMubPuJtGgjWYQa-Ra_bScJsc5LF9KznPEBXFG2QjqJNqlHrtziyZAqJfWFBuIqYwQg
   ```

   Record the *Authorization_Code* value.
6. From your command line interface, enter the following command (as one line):

   ```
   curl -v -k -u Client_ID:Client_Secret -X POST -d
   'grant_type=authorization_code&redirect_uri=Redirect_URL&code=Authorization_Code'
   'Token_Endpoint_URL'
   ```

   where:
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Redirect_URL* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Authorization_Code* is as recorded in step 5.
   - *Token_Endpoint_URL* is as recorded in step 2.

   Note: If you are adapting the command for use with your command line interface, you must send `application/x-www-form-urlencoded` content.
   For example:

   ```
   curl -v -k -u 094d3694-5c35-43b4-bea6-13a9ab0fdb89:nR7yC0dS4fH0tL8dQ2aV5m
   S2aF7lC5uT7vJ6vP6fI6gS6dH5pE -X POST -d 'grant_type=authorization_code&
   redirect_uri=https://example.com/redirect&code=AAIdK4UzxGPkKVYGcfViaTui8I1OpJs4nrv
   U4xuuD4uNlMuR-lR8XBM6zfk6aPBU5NeTHQUlE2Z-OFKYmXasCexqocPMAV-GLxf539m6x0-FT3LBTh14s_
   DBgsyNpSxAC7q3HhkxKbquGhGH2paD4ogUIAvwJDpXZaV-ekfqVLAZzA'
   'https://host.com/myorg/sb/oauth-end/oauth2/token'
   ```

   The response takes the following form:

   ```
   { "token_type":"bearer", "access_token":"Access_Token",
   "expires_in":3600, "scope":"Scope}
   ```

   Record the *Access_Token* value that is displayed in the message.
   For example:

   ```
   { "token_type":"bearer", "access_token":"AAEFYy1hbGy8uvozvrfdHMbA1kw5BTt1yuaoO2B
   QYaInzRX7cSI5HB9OqsM2LxVs7mJEfDGRHgSqlWyXZ1zMFusHqRwBgur_fEi0k7xYzoTG1IKWaL990HHpZwuGYclf
   wPaOcq29AWaUY21x1E7XYVaG-k5hJrEwiBa62IeMk_inl6nDnqfgAA", "expires_in":3600, "scope":"view_branches"
   }
   ```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for implicit authorization

This tutorial shows you how to acquire an access token for the OAuth scheme *implicit authorization flow* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the implicit authorization flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Authorization Endpoint URL value.
3. Open a new browser tab or window and enter the following URL (as one line):

   ```
   Authorization_Endpoint_URL?response_type=token&
   redirect_uri=Redirect_URL&scope=Scope&
   client_id=Client_ID
   ```

   where:
   - *Authorization_Endpoint_URL* is as recorded in step 2.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

   For example:

   ```
   https://host.com/myorg/sb/oauth-end/oauth2/authorize?
   response_type=token&redirect_uri=https://example.com/redirect&scope=view_branches&
   client_id=95de3c66-5c52-486b-9229-7985b16bcd8a
   ```

4. Authenticate with your LDAP registry when prompted to do so. You are presented with a page requesting permission for your OAuth Application to access your Branches API.
5. Click Allow Access. You are redirected to your redirection URL with additional information appended to the URL. This has the following form:

   ```
   Redirect_URL#access_token=Access_Token&expires_in=3600&
   scope=Scope&token_type=bearer
   ```

   For example:

   ```
   https://example.com/redirect#access_token=AAEFcC1hbGx0zz3KhAki_
   vD-PSaAIMx6tbsxBJ858oYRWRUwP6aO7BQPySCURVj0e2FJ49pIvooAZYUZfNCz2cfqHMhiN
   9IR6ID71c563oBiCV_f_o7lvTC68Wad2CTTn5Ys7hrq0Z-hBbxBe-m3hI7bZBf44reE821Oq
   rSX2EqwvXVaalkqag&expires_in=3600&scope=view_branches&token_type=bearer
   ```

   Record the *Access_Token* value that is displayed in the query parameters.

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for password flow (public)

This tutorial shows you how to acquire an access token for the OAuth scheme *password flow (public)* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the password flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

   ```
   curl -v -k -X POST -d 'grant_type=password&scope=Scope&
   username=User_Name&password=Password&client_id=Client_ID'
   'Token_Endpoint_URL'
   ```

   where:
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Token_Endpoint_URL* is as recorded in step 2.
   - *User_Name* is a valid user name from your LDAP registry.
   - *Password* is the password that corresponds to *User_name* in your LDAP registry.

   Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
   For example:

   ```
   curl -v -k -X POST -d 'grant_type=password&scope=view_branches&
   username=USER@example.com&password=PASSWORD&client_id=552b44f4-d533-4593-bec7-24dacf129847'
   'https://host.com/myorg/sb/oauth-end/oauth2/token'
   ```

   A message is returned, of the following form:

   ```
   { "token_type":"bearer", "access_token":"Access_Token",
   "expires_in":3600, "scope":"Scope}
   ```

   Record the *Access_Token* value that is displayed in the message.
   For example:

   ```
   { "token_type":"bearer", "access_token":"AAEFcC1hbGxDUs-Pg6-O_
   6N85VSN5uCNrk83vDCqK6cksws086DsKWY-zfr0ANPD9CtcY4PgSq5bC_mPnSCUyhSlEj1an
   qOj1YXEWJEW-iN2xNlm4Uwm1l2x_yJrEEud_uBcX5m4L1Jev7iM082Ozb6j4aJCGsN5CL69-
   7JqZSLMxCgb-Jwbkg", "expires_in":3600, "scope":"view_branches" }
   ```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for access code (public)

This tutorial shows you how to acquire an access token for the OAuth public scheme *access code flow* in IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the Password Flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL and Authorization Endpoint URL values.
3. Open a new browser tab or window and enter the following URL (as one line):

   ```
   Authorization_Endpoint_URL?response_type=code&
   redirect_uri=Redirect_URL&scope=Scope&client_id=Client_ID
   ```

   where:
   - *Authorization_Endpoint_URL* is as recorded in step 2.
   - *Scope* is `view_branches`, as specified for the secured API in step 8 of the Securing an API by using OAuth 2.0 tutorial.
   - *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

   For example:

   ```
   https://host.com/myorg/sb/oauth-end/oauth2/authorize?
   response_type=code&redirect_uri=https://example.com/redirect&scope=view_branches&client_id=
   801a94e9-556f-4034-b333-f0f2c680f5f2
   ```

4. When prompted, authenticate using the LDAP registry you specified when creating your API. You are presented with a prompt asking you to confirm that you want to grant your OAuth Application access to your Branches API.
5. Click Allow Access. You are redirected to your Redirect URI which will have additional information appended. It will take the following form:

   ```
   Redirect_URI?code=Authorization_Code
   ```

   Record the *Authorization_Code*.
   For example:

   ```
   https://example.com/redirect?code=AAKjnhCDCUsQszr5osEEOo2ZDH7WN80znOq0nwEVzzUEub
   HWu7etbQzXEOIB5EjFfYQcQHegAv5BYNufBK2_HG-cXvi4eW_3a8wNga4NbDsuGGnQNBdNcWkR1U7XKUXrhOVZ10hN
   klII4WjCuWpjDtsYD3U6ihdxcITUY1IhMjFxqQ
   ```

6. From your command line interface, enter the following command (as one line):

   ```
   curl -v -k -X POST -d 'grant_type=authorization_code&code=
   Authorization_Code&redirect_uri=Redirect_URI&
   client_id=Client_ID' 'Token_Endpoint_URL'
   ```

   where:
   - *Authorization_Code* is as you have previously recorded.
   - *Token_Endpoint_URL* is as recorded in step 2
   - *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

   Note: If you are adapting the command for use with your command line interface, you must send `application/x-www-form-urlencoded` content.
   For example:

   ```
   curl -v -k -X POST -d 'grant_type=authorization_code&redirect_uri=
   https://example.com/redirect&code=AAI3hfmo9HroOLzONkJuCUkZVUlUzSZXC4LK6xGC2Ta
   5bODaDPBBZBeRnV2wNWSuibINAl7mJ65RWZJ_IQ6wqG26KW8ipdlj2KUDnjq1E_hjIE5GxvWk0avj
   FbBPqm9sc-IKi27l36Ps01pPAQpTweB6LNxyvlqyOcHH4pwTjqFwxA&
   ```

```
client_id=9a131485-75f4-4e70-8657-a844b4a279af'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

The response takes the following form:

```
{"token_type":"bearer", "access_token":"Access Token",
"expires_in":3600, "scope":"scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFcC1hbGyoPkKTr_0_HQ3e7Q_u1Gx
_aejgznGs7im6H_jNR3xVSZrTcTCuuyvvj9u0K1xqpLXDQOjfCFhGfWOMql-LQ9F9-NjStVeIIpWpJ5Wjx
yCW1oU459k_Uvmm03SdHV4eS1PSiG_YroxqCgWDkC1b2UKFM5_72HYglbCsvBEstSuqJQ",
"expires_in":3600, "scope":"view_branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

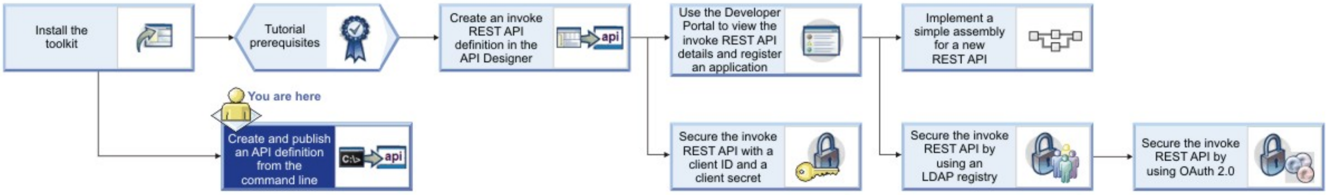
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating and publishing an API definition from the command line

This tutorial shows you how to use the developer toolkit to create an API definition, include it in a Product, and publish the Product to a Catalog in API Manager by using IBM® API Connect Version 5.0.7 and later.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.6 and earlier, see Developer toolkit tutorials for V5.0.6 and earlier.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
You are here
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

With the developer toolkit, you can create API definition files locally and develop them by using your chosen editor. When you are ready to make them available to application developers, you publish them to a Catalog in API Manager, where they can be managed through their lifecycle. In this tutorial, you will create and publish an API.

To make an API available, it must be included in a Plan, and the Plan must be included in a Product; you then stage the Product to API Manager. In this tutorial, you first create an API, then you create a Product that includes the API. A default Plan, that contains the API, is created automatically in the Product.

Instructions are provided to create and publish an API that represents a banking service for obtaining loan quotations; however, if you want to create and publish your own API, you can adapt the instructions accordingly.

In this tutorial, you will complete the following activities:

- Create a local API definition YAML file.
- Validate the API definition.
- Create a local Product definition.
- Validate the Product definition.
- Log in to API Manager.
- Publish the Product to API Manager.

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`**.

Create a local API definition YAML file

To create a local API definition YAML file, you use the **`apim create`** command.

Complete the following steps:

1. Create a folder to hold your API and Product definitions, and change to that folder in a command window.
2. Enter the following command:

   ```
   apic create --type api --title Loans
   ```

   This command creates a file called loans.yaml, and assumes the following defaults:

version	1.0.0
basepath	/loans

   However, you can supply further command options to override these values.
3. Examine the contents of the loans.yaml file by opening it in your chosen editor.
   Although not part of this tutorial, the API can be further developed by modifying the OpenAPI (Swagger 2.0) definition in the YAML file.

Validate the API definition

To validate the syntactical correctness of the API definition YAML file, you use the **`apim validate`** command.

Enter the following command:

```
apic validate loans.yaml
```

The validation should complete successfully.

Create a local Product definition

To create a local Product definition YAML file, you use the **`apim create`** command.

Complete the following steps:

1. Enter the following command:

   ```
   apic create --type product --title "Banking Services" --apis loans.yaml
   ```

   This command creates a product definition file called banking-services.yaml that includes a reference to the loans.yaml API definition file that you defined previously. The following defaults are assumed:

name	banking-services
version	1.0.0

   However, you can supply further command options to override these values. The value of the name property determines the file name.
2. Examine the contents of the banking-services.yaml file by opening it in your chosen editor. Note that a reference to the loans API is included.
   Although not part of this tutorial, you can modify the Product definition in the YAML file; for example, you can add further Plans and APIs.

Validate the Product definition

To validate the syntactical correctness of the Product definition YAML file, you use the **`apim validate`** command.

Enter the following command:

```
apic validate banking-services.yaml --product-only
```

The validation should complete successfully.
By default, the **apic validate** command validates the Product definition YAML file and all the API definition files that it references. If you supply the **--product-only** option, only the Product definition is validated.

Log in to API Manager

Complete the following steps:

1. Enter the following command:

   ```
   apic login
   ```

2. At the prompts, enter the following information:
   - Server: The virtual host name or virtual IP address of your Management cluster.
   - Username: The user name with which you are registered in API Manager.
   - Password: Your API Manager password.

   Note: You can also supply your credentials as command parameters. However, if you use the command interactively, your password is hidden.

Stage the Product to API Manager

For application developers to be able to use an API, it must be published to a Catalog. A Catalog has an associated Developer Portal and runtime capability. For example, a simple provider organization might consist of a development Catalog and a production Catalog.

Complete the following steps:

1. To list the Catalogs that are available in API Manager, enter the following command:

   ```
   apic catalogs --all-organizations --server management_cluster_hostname_or_address
   ```

   You should see the identifier for a Catalog called **sb**, which corresponds to the development Catalog that is created by default when IBM API Connect is installed; the identifier includes the organization that contains it. For example:

   ```
   apic-catalog://myserver.mydomain.com/orgs/myorg/catalogs/sb
   ```

2. Publish the Product to the development Catalog by entering the following command (all on one line):

   ```
   apic publish banking-services.yaml --catalog sb --organization
   URL_path_segment_of_your_provider_organization
   --server management_cluster_hostname_or_address
   ```

   Note: You can choose only to stage the Product, by supplying the **--stage** option. If a Product is staged, its APIs are not yet available to application developers, and the Product must be subsequently published by using the API Manager user interface. For more information, see Managing your Products.
3. Confirm that your Product and its referenced APIs have been published to the development Catalog by entering the following commands:

   ```
   apic products --catalog sb --organization URL_path_segment_of_your_provider_organization --server
   management_cluster_hostname_or_address
   apic apis --catalog sb --organization URL_path_segment_of_your_provider_organization --server
   management_cluster_hostname_or_address
   ```

   You should see the **banking-services** Product and the **loans** API listed.

Note: You can define default values for your management server, Catalog and provider organization so that you do not have to supply them as command options; enter the following command (all on one line):

```
apic config:set catalog=apic-catalog://management_cluster_hostname_or_address
/orgs/URL_path_segment_of_your_provider_organization
/catalogs/URL_path_segment_of_your_catalog
```

For example:

```
apic config:set catalog=apic-catalog://myhost.com/orgs/myorg/catalogs/sb
```

You can obtain the required value of the **catalog** configuration property for a specific Catalog from the API Manager user interface; for more information, see Obtaining the publish target URL for a Catalog.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a local API definition YAML file.
- Validated the API definition.
- Created a local Product definition.
- Validated the Product definition.
- Logged in to API Manager.
- Defined your Management cluster and your organization.
- Staged the Product to API Manager.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 and earlier

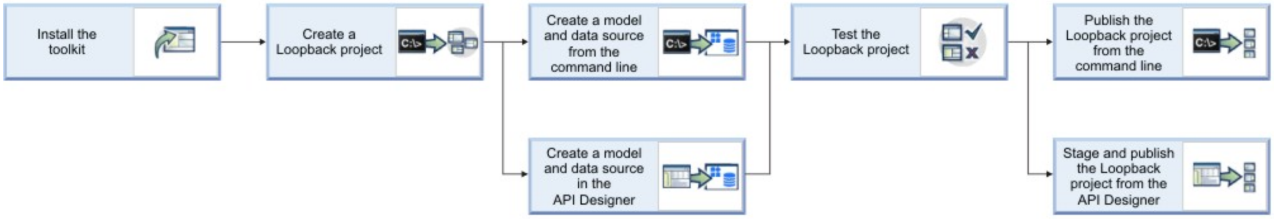
Developer toolkit tutorials for V5.0.6 and earlier

Tutorials for using the developer toolkit in IBM® API Connect Version 5.0.6 and earlier.

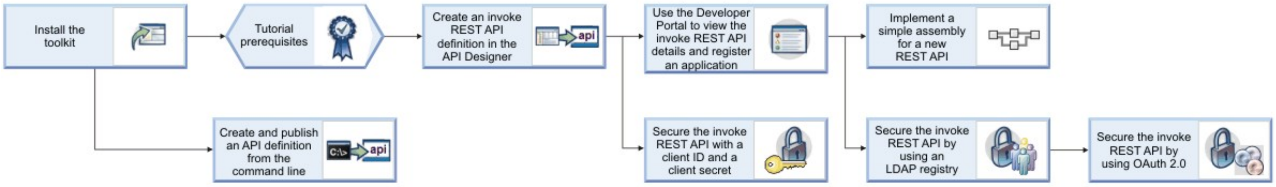
Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
With the developer toolkit tutorials you can either create API definitions that invoke an existing API implementation, or you can create your own API implementations with LoopBack®.

The following diagrams shows the sequential flow through the IBM API Connect Developer toolkit tutorials. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.

Working with LoopBack projects


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

Working with API definitions that call an existing endpoint


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

The following sections contain the tutorials that take you through each of these scenarios:

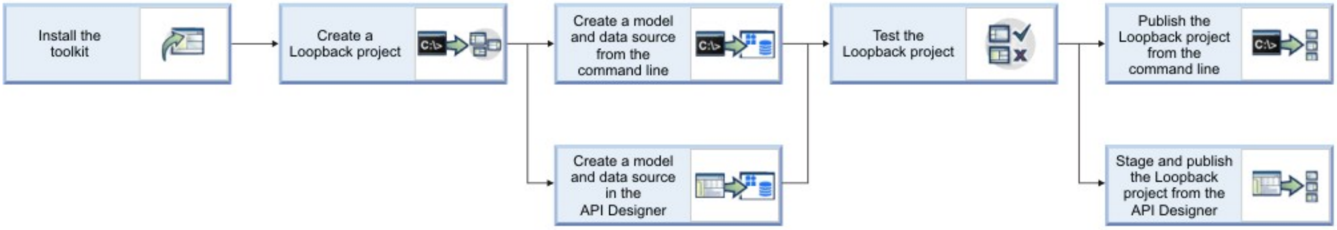
- **Tutorials for working with LoopBack projects**
  Tutorials for using LoopBack functions in the developer toolkit in IBM API Connect Version 5.0.6 and earlier.
- **Tutorials for working with API definitions that call an existing endpoint**
  Tutorials for creating API definitions to expose and secure BankA APIs in IBM API Connect Version 5.0.6 and earlier.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials for working with LoopBack projects

Tutorials for using LoopBack® functions in the developer toolkit in IBM® API Connect Version 5.0.6 and earlier.

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

- **Tutorial: Creating a LoopBack project from the command line**
  This tutorial shows you how to create a new LoopBack project using the command line in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Creating a model and a data source from the command line**
  This tutorial shows you how to add a LoopBack model and data source to a project using the `apic` command-line tool in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Creating a model and a data source in the API Designer**
  This tutorial shows you how to add a new model and data source to a LoopBack project by using the API Designer in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Installing LoopBack connectors**
  This tutorial shows you how to install a LoopBack data source connector manually by using the command line in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Testing a LoopBack project**
  This tutorial shows you how to run a LoopBack project locally for testing by using either the command line or the API Designer Explore tool in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Publishing a project from the command line**
  This tutorial shows you how to publish a LoopBack project from the command line in IBM API Connect Version 5.0.6 and earlier. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.
- **Tutorial: Staging and publishing a project from the API Designer**
  This tutorial shows you how to stage and publish a project using the API Designer in IBM API Connect Version 5.0.6 and earlier. *Staging* a project copies all the files to the target, but does not run the project application code. *Publishing* a project copies all the project files to the target and runs the project application code. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating a LoopBack project from the command line

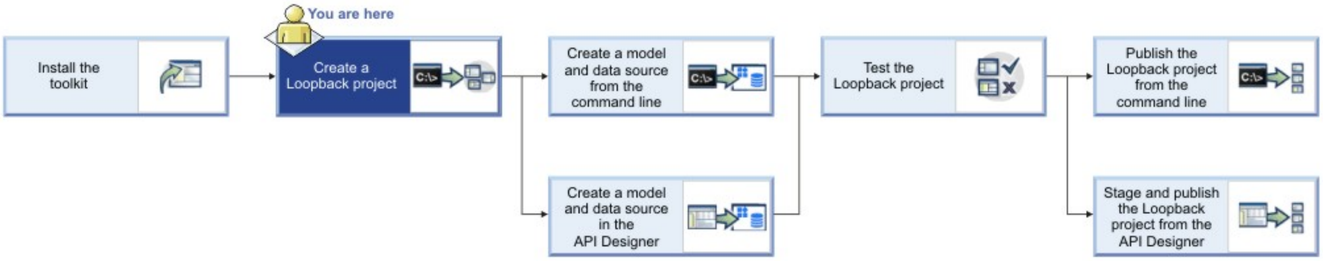
This tutorial shows you how to create a new LoopBack project using the command line in IBM® API Connect Version 5.0.6 and earlier.

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
You can accomplish the same thing using API Designer; for more information, see Creating new projects with API Designer. For more information on LoopBack, see LoopBack documentation.

Before you begin

Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

Note: To complete this tutorial, you must have internet access, because LoopBack installs dependencies from the public npm repository. Alternatively, you can configure a private npm repository.

Procedure

Complete the following steps:

1. From the command-line interface, enter the following command, which is used to create and manage LoopBack applications:

   ```
   apic loopback
   ```

   You will now create a project called "acme-bank".
2. At the prompt, enter `acme-bank` as the project name. Then, press Enter.

   ```
   ? What's the name of your application? acme-bank
   ```

   Note: In general, a project name can contain any characters except blank space (" "), forward slash ("/"), ampersand ("&"), at ("@"), plus ("+"), percent ("%"), and colon (":") .
3. Enter the name of the directory in which to create the project. You can press Enter to use a directory with the same name as the project, or type a new name and press Enter.

   ```
   ? Enter name of the directory to contain the project: acme-bank
   ```

4. Select the version of LoopBack to use. Choose the current production version, 3.x.

   ```
   ? Which version of LoopBack would you like to use? 3.x (current)
   ```

5. Specify the kind of application that you want to create by using the arrow keys to select empty-server:

   ```
   ? What kind of application do you have in mind? (Use arrow keys)
   ❯ empty-server (An empty LoopBack API, without any configured models or datasources)
     hello-world (A project containing a basic working example, including a memory datab
   ase)
     notes (A project containing a basic working example, including a memory database)
   ```

   Then, press Enter to create an empty LoopBack API.

The tool displays a number of messages as it creates the project directory and adds a number of directories and files to it. It also runs **npm install** to install all the project dependencies, as specified in **package.json**. This process creates a **node_modules** directory and might take some time.

An empty LoopBack project contains the following directories:

- **server**: contains server model and data source definitions, and other server code.
- **client**: contains client model definitions, and static assets such as HTML, CSS, JavaScript files, and so on.
- **definitions**: contains YAML definition files.

For more information on the contents of a LoopBack project, see Project layout reference (LoopBack documentation).

What you did in this tutorial

In this tutorial, you created a new LoopBack project called "acme-bank".

What to do next

Create a model and a datasource by following either of these tutorials:

- Tutorial: Creating a model and a data source from the command line.
- Tutorial: Creating a model and a data source in the API Designer.

Tutorial: Creating a model and a data source from the command line

This tutorial shows you how to add a LoopBack® model and data source to a project using the **apic** command-line tool in IBM® API Connect Version 5.0.6 and earlier.

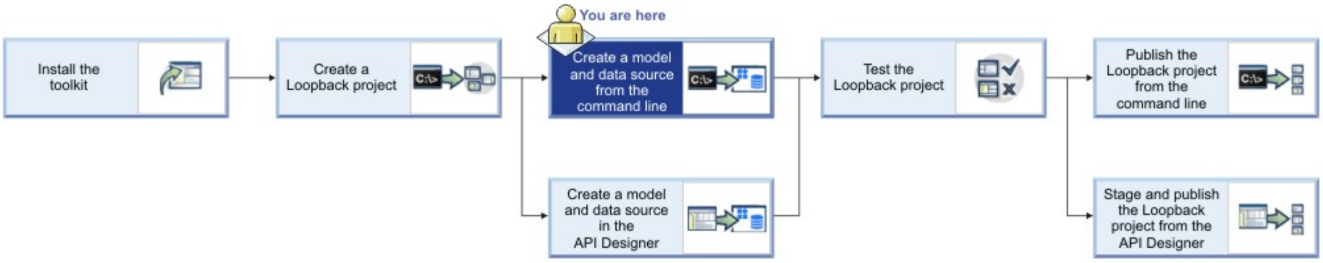
Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also create a LoopBack project (the "acme-bank" project) as described in Tutorial: Creating a LoopBack project from the command line and make sure the current working directory is the project root directory:

```
cd acme-bank
```

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

About this tutorial

You're going to start defining the acme-bank API that contains models for bank branches, accounts, and so on. In this tutorial, you will complete the following activities:

- Add a new data source to your LoopBack project.
- Add a model to your LoopBack project.

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**.

Adding a data source to a project

Complete the following steps:

1. From the command line, enter the following command:

   ```
   apic create --type datasource
   ```

2. At the prompt, enter `bankDS` as the name of the data source:

   ```
   ? Enter the data-source name: bankDS
   ```

   Note: In general, you can use any alphanumeric character, dashes, and underscores in a data source name.
   The tool prompts you to select the connector to use for the data source:

   ```
   ? Select the connector for myds: (Use arrow keys)
   ❯ In-memory db (supported by StrongLoop)
     IBM DB2 (supported by StrongLoop)
     IBM DashDB (supported by StrongLoop)
   ```

```
  IBM MQ Light (supported by StrongLoop)
  IBM Cloudant DB (supported by StrongLoop)
  IBM DB2 for z/OS (supported by StrongLoop)
  MongoDB (supported by StrongLoop)
(Move up and down to reveal more choices)
```

3. Press Enter to choose the in-memory data source. This data source, suitable for development and testing, is built in to LoopBack.
4. When prompted for the key to use for client persistence, press Enter without typing a value:

   ```
   ? window.localStorage key to use for persistence (browser only):
   ```

   For now, this will be a server-only data source, so this setting is not relevant.
5. When prompted for the path to a file to use for server persistence, press Enter without typing a value:

   ```
   ? Full path to file for persistence (server only):
   ```

   For now, leave this empty since you won't need to persist data across server restarts.

The tool updates the app's OpenAPI (Swagger 2.0) definition file and the **server/datasources.json** file with settings for the new data source. For more information, see Connecting models to data sources (LoopBack documentation).

Note:
The in-memory data source is built in to LoopBack and is suitable only for development and initial testing. When you are ready to connect your models to a real data source such as database server, follow the same procedure, but choose the connector for your back-end data store. The tool will prompt you for additional settings and will automatically install the appropriate connector package from npm.

The Oracle, DB2, and SQLLite connectors have additional prerequisites; for more information, see Tutorial: Installing LoopBack connectors.

Adding a model to a project

Complete the following steps:

1. From the command line, enter the following command:

   ```
   apic create --type model
   ```

2. At the prompt, enter `branch` as the name of the model:

   ```
   ? Enter the model name: branch
   ```

   Note: In general, you can use any alphanumeric characters plus dashes and underscores in a model name.
   The tool prompts you to select the data source to use from a list that includes the in-memory data source that you just added to the project:

   ```
   ? Select the data-source to attach item to: (Use arrow keys)
     (no data-source)
   ❯ bankDS (memory)
   ```

3. Use the arrow key to select bankDS (memory), and then press Enter.
   The tool prompts you to select the model's base class from a list that includes the LoopBack built-in models and any other models defined in the project.

   ```
   ? Select model's base class (Use arrow keys)
     Model
   ❯ PersistedModel
     ACL
     AccessToken
     Application
     Change
     Checkpoint
   (Move up and down to reveal more choices)
   ```

   If you were defining a user model, you would generally select the **User** as the base model; otherwise in most cases, you would choose PersistedModel as the base class for a custom model.

4. For this tutorial, use the arrow keys to select PersistedModel, and then press Enter.
5. When prompted as to whether you want to expose the model's REST API, press Enter to choose the default (yes):

   ```
   ? Expose branch via the REST API? (Y/n)
   ```

   Tip: If the model is exposed over REST, then all the standard create, read, update, and delete operations are available via REST endpoints; see PersistedModel REST API for more information.
   Since you chose to expose the model over REST, the tool prompts you for the plural form of the model name.

   ```
   ? Custom plural form (used to build REST URL):
   ```

6. Press Enter to use the default standard English rules for pluralization (in this case, "branches").
7. When prompted as to whether you want to create a server-only model, or a common model that can be used in both server or client LoopBack API, press Enter to keep the default (common):

   ```
   ? Common model or server only? (Use arrow keys)
   ```

8. When prompted to add properties to the model, enter `type` for the first property:

   ```
   Let's add some branch properties now.

   Enter an empty property name when done.
   ? Property name: type
   ```

9. When prompted to select the data type of the property, press Enter to select the string type:

   ```
   ? Property type: (Use arrow keys)
   ❯ string
     number
     boolean
     object
     array
     date
     buffer
   ```

10. To indicate that the property is required, enter `y`:

    ```
    ? Required? (y/N) y
    ```

    Then, press Enter.

11. When prompted for a default value, press Enter for no default value:

    ```
    ? Default value [leave blank for none]:
    ```

12. When prompted to add another property, add a property called `phone`:

    ```
    Let's add another branch property.
    Enter an empty property name when done.
    ? Property name: phone
    ```

13. Define the property type as **string** and set the property to **not required**.
14. When prompted to add another property, just press Enter to finish adding the model.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Added a data source to your project using the command-line tool.
- Added a model to your project using the command-line tool.

What to do next

Test your LoopBack project.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating a model and a data source in the API Designer

This tutorial shows you how to add a new model and data source to a LoopBack® project by using the API Designer in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.

Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

You must also do the following:

1. Create a LoopBack project. For more information, see Tutorial: Creating a LoopBack project from the command line.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

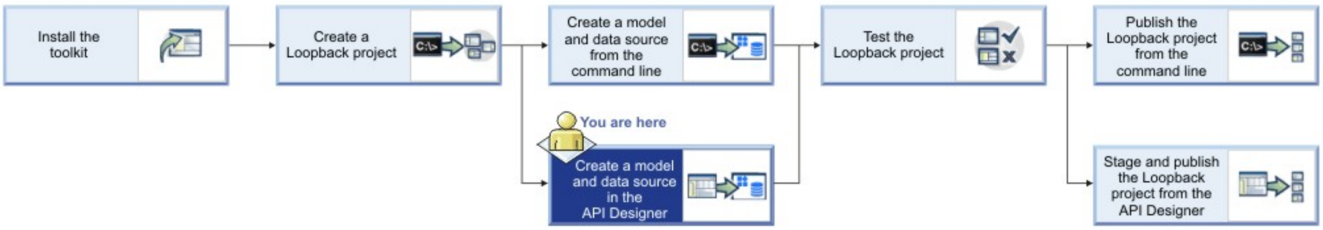
   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
You are here
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

About this tutorial

This tutorial builds on Tutorial: Creating a LoopBack project from the command line. In this tutorial, you will complete the following activities:

- Create a model with the API Designer.
- Create a data source with the API Designer.

Adding a data source

By default, an empty LoopBack project does not have any data sources defined. For now, you will add the in-memory data source, that is suitable for development and testing.

Complete the following steps:

1. Click Data Sources.
2. Click Add. The New LoopBack Data Source window opens.
3. Enter `bankDS` in the Name text field.
   Note: You can use any alphanumeric characters, dashes, and underscores in a data source name.
4. Click **New**.
5. By default, the Connector setting shows In-memory db and the other settings are blank. Keep the default settings for now, and API Designer automatically saves the new data source.
   Note: The In-memory data source is built in to LoopBack and is suitable only for development and initial testing. When you are ready to connect your models to a real data source such as database server, change the Connector setting accordingly then install the data source connector by following the instructions in Tutorial: Installing LoopBack connectors. Then enter the connector settings (host name, port, database name, user name, password) as appropriate for your Connector type, and click the Save icon . Then, API

Designer automatically tests the connection to the data source. If the test is successful, it displays the message Success - Data source connection test succeeded.

6. You can test the data source connection by clicking 🏃. The message Success - Data source connection test succeeded is displayed.
7. Click All Data Sources. The data source will appear in the list of data sources, and the editor updates the **`server/datasources.json`** file with settings for the new data source.

Adding a model

Complete the following steps:

1. Click Models .
2. Click Add. The New LoopBack Model window opens.
3. Enter `branch` in the Name text field, then click New.
   Note: You can use any alphanumeric characters, dashes, and underscores in a model name.
4. In the Data Source field, select bankDS.
5. In the Properties section, click the Add property icon ⊕ .
6. In the Property Name text field, enter `type`.
7. For Type, select string.
8. Select Required to make the property required. This means that it must have a value when you add or update a model instance. For now, keep the default values for the other settings:
   - Is Array: Whether the property is a JavaScript array with elements of the specified type.
   - ID: Whether the property is a unique identifier.
   - Index: Whether the property represents a column (field) that is a database index.
   - Description: Text description of the property.

   For more information, see Model definition JSON file reference.
9. Click the Add property icon ⊕ again to add another property.
10. In the Property Name text field, enter `phone`.
11. For Type, select string. Leave the other settings with their default values.
12. Click the Save icon 💾 to save your changes.
13. Click All Models to finish editing the model.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a new LoopBack model.
- Created a new LoopBack data source.

What to do next

Test your LoopBack project.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Installing LoopBack connectors

This tutorial shows you how to install a LoopBack data source connector manually by using the command line in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.

To install a data source connector, you must be able to run an `npm install` command, which requires a connection to the internet. Alternatively, you can configure a private npm repository.

The Oracle, DB2, and SQLLite connectors require C compiler tools to build and install binary extensions. The exact requirements depend on your operating system. Once you have installed compiler tools, you can install the connector following the instructions in the following procedure.

Windows

Note: You may need to restart your computer at various points during installation.
Install:

- Microsoft .NET Framework 4.0
- Visual Studio. Use **Visual Studio Community** unless you want purchase Visual Studio Enterprise. Run the installer, check "Visual C++" under "Programming Languages", and accept the default installation location.
  Note: The Visual Studio installation can take a long time.
- Windows SDK. If the installation fails, look for **C++ 2010 x64&x86 Redistributable** in your installed programs and uninstall it.
- Python 2.7.10.
- Version 3 of npm. Enter the following command:

  ```
  npm install -g npm
  ```

  Then ensure the `npm` command uses the correct version:

  ```
  npm -v
  ```

  If the version shown is not 3.*x*.*x*, then edit your system PATH to ensure that **C:\Users\*username*\AppData\Roaming\npm** supersedes any other entries.

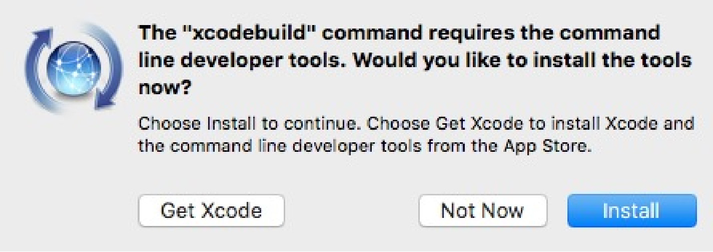
- For 64-bit builds of Node.js and native modules you also need the Windows 7 64-bit SDK. If the installation fails, try uninstalling any C++ 2010 x64&x86 redistributable that you have installed first. If you get errors that the 64-bit compilers are not installed you may also need the compiler update for the Windows SDK 7.1.

Note: Do not use Cygwin (Windows bash shell emulator). Use the Windows command shell instead.

Mac OS X

You must have the command-line developer tools (or full installation of Xcode) and Python.

If you don't already have Xcode or the command-line tools installed, installation will prompt you as follows:



The "xcodebuild" command requires the command line developer tools. Would you like to install the tools now?

Choose Install to continue. Choose Get Xcode to install Xcode and the command line developer tools from the App Store.

Click Install to install the command-line developer tools only. Alternatively, you can click Get Xcode to install the full Xcode product, but doing this can take a long time.

Python is also required. Most versions of OSX come with Python by default. If for some reason you don't have it, download and install Python.

Linux

Many Linux® systems come with the necessary tools. The specific requirements are:

- Python version 2.7. **NOTE**: version 3.*x* is not supported.
- `make`
- A C/C++ compiler toolchain, like GCC. **NOTE**: g++ version 4.2 or later is required.

On Debian and Debian-derived distributions (Ubuntu, Mint, and so on), use the command:

```
apt-get install build-essential
```

About this task

Before you can use a LoopBack data source to access data in a backend system such as a database, you must install the data source connector. The In-memory and email connectors are built-in to LoopBack®, so you don't need to install them. The command-line tool and API Designer will automatically prompt you to install connectors as needed, but you can also do it manually

Install a new data source connector manually by following these steps:

Procedure

1. Open a new shell window.
2. Install the connector package by entering this command in the project root directory:

   ```
   npm install --save connector-package
   ```

   where *connector-package* is the name of the npm package for the LoopBack connector, as shown in the table.

   Table 1. LoopBack Connectors

Data Source	Connector Package
Database connectors	
Cloudant	loopback-connector-cloudant
DashDB	loopback-connector-dashdb
DB2	loopback-connector-db2
DB2 for zOS	loopback-connector-db2z
Microsoft SQL Server	loopback-connector-mssql
MongoDB	loopback-connector-mongodb
MQ Light	loopback-connector-mqlight
MySQL	loopback-connector-mysql
Oracle	loopback-connector-oracle
PostgreSQL	loopback-connector-postgresql
Other connectors	
SOAP web services	loopback-connector-soap
REST web services	loopback-connector-rest

3. For Oracle, you must modify /etc/environment, reboot the system, then enter the following commands:

   ```
   $ echo "/home/strongbot/orapp2/node_modules/instantclient" | sudo tee -a
   /etc/ld.so.conf.d/loopback_oracle.conf
   $ sudo ldconfig
   ```

Results

You successfully installed a new data source connector for LoopBack.

Installing the Oracle connector

About this task

The LoopBack Oracle connector might require some additional installation and troubeshooting steps. Either follow the manual installation and troubleshooting steps or install the LoopBack CLI that provides an additional command to install and troubleshoot the Oracle connector:

- Follow the instructions in Installing the Oracle connector for your platform.
- If you prefer (or if you still have issues with the connector), install the LoopBack CLI by using the comand:

  ```
  npm install -g loopback-cli
  ```

  Then enter the following command to install and troubleshoot the Oracle connector:

  ```
  lb oracle
  ```

  This command determines if the connector is ready to use. If it is, the tool will print "Oracle connector is ready" and exit. Otherwise, it will prompt you to install Oracle Instant Client, `loopback-connector-oracle`, and the `oracledb` module. For more information, see Oracle installer command.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

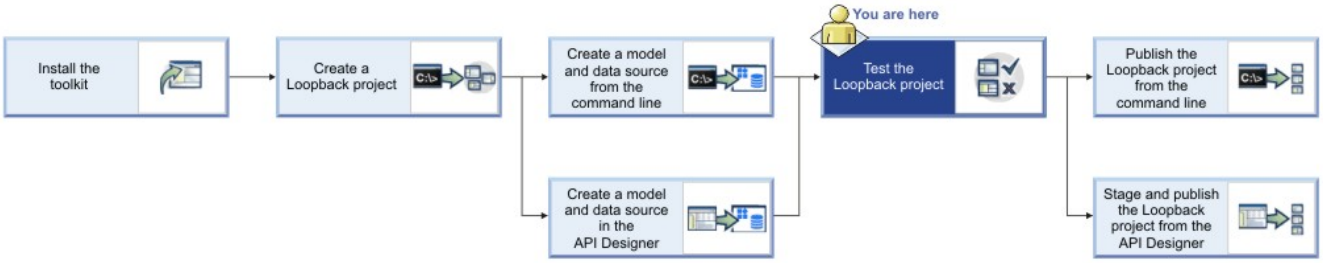
Tutorial: Testing a LoopBack project

This tutorial shows you how to run a LoopBack® project locally for testing by using either the command line or the API Designer Explore tool in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.



You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
Stage and publish the Loopback project from the API Designer

Also ensure your current working directory is the project top-level directory. Enter the following command:

```
cd acme-bank
```

About this tutorial

This tutorial builds on Tutorial: Creating a model and a data source in the API Designer or Tutorial: Creating a model and a data source from the command line (you only need to complete one of these). In this tutorial, you will complete the following activities:

- Test a LoopBack project by using the command line
- Test a LoopBack project by using the API Designer Explore tool

Note: You can test a LoopBack project either by using the command line or the API Designer, depending on your preference. The procedures accomplish the same result, which is to run the LoopBack project (Node.js application) and the Micro Gateway on your local system.

Test a LoopBack project by using the command line

Complete the following steps:

1. Enter the following command:

   ```
   apic start
   ```

   This runs the LoopBack project (API) and the Micro Gateway locally. You will see the message:

   ```
   Service acme-bank (id 1) started on port 4001
   Service acme-bank-gw (id 2) started on port 4002
   ```

   Note: If you previously ran other projects, you may see different port numbers.
2. To confirm that the project is running locally, open `http://localhost:4001` in your browser. For the default LoopBack (empty or hello-world) project, you'll see something like this:

   ```
   {"started":"2016-03-07T22:24:55.322Z","uptime":35.839}
   ```

3. You can then test any of the API endpoints by using `curl`. For example, to display all the model instances in the "acme-bank" project, enter the following command:

   ```
   curl --request GET \
     --url 'https://localhost:4002/api/branches' \
     --header 'accept: application/json' \
     --header 'content-type: application/json' \
     --header 'x-ibm-client-id: default' \
     --header 'x-ibm-client-secret: SECRET'
   ```

If the project has some model instances, the console will display the JSON data. Otherwise, the console will display an empty array `[]`.

Test a LoopBack project by using the API Designer Explore tool

To test your API endpoints by using the API Designer Explore tool, complete the following steps:

1. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

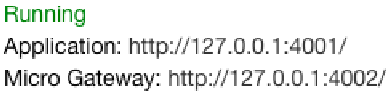
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

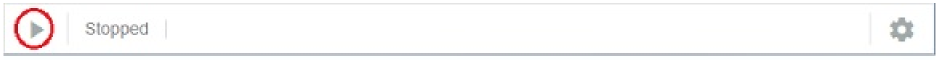
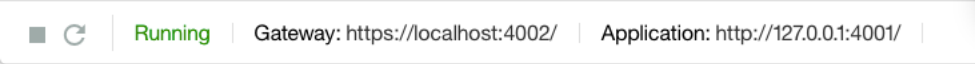
   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
2. V5.0.0 ONLY Start the local test servers by completing the following steps:
   a. Click Run. then click Start.

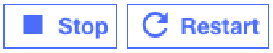
   b. Wait until the **Running** message is displayed:
   
   Running
   Application: http://127.0.0.1:4001/
   Micro Gateway: http://127.0.0.1:4002/
   Depending on your project configuration and whether other processes are running, different port numbers might be displayed.

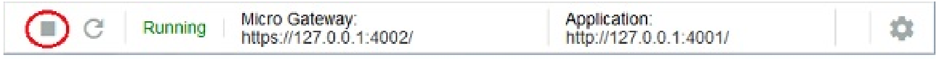
3. Start the local test servers by completing the following steps:
   a. In the test console at the bottom of the screen, click the Start the servers icon:
   
   Stopped
   b. Wait until the **Running** message is displayed:
   
   Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/
   Depending on your project configuration and whether other processes are running, different port numbers might be displayed.

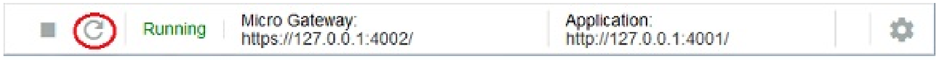
4. Click http://127.0.0.1:*port_number* to display the API root endpoint. For the default LoopBack (empty or hello-world) project, you'll see something like this:

   ```
   {"started":"2016-03-07T22:24:55.322Z","uptime":35.839}
   ```

   Note:
   - V5.0.0 ONLY To stop your project, click Stop. To restart it, click Restart

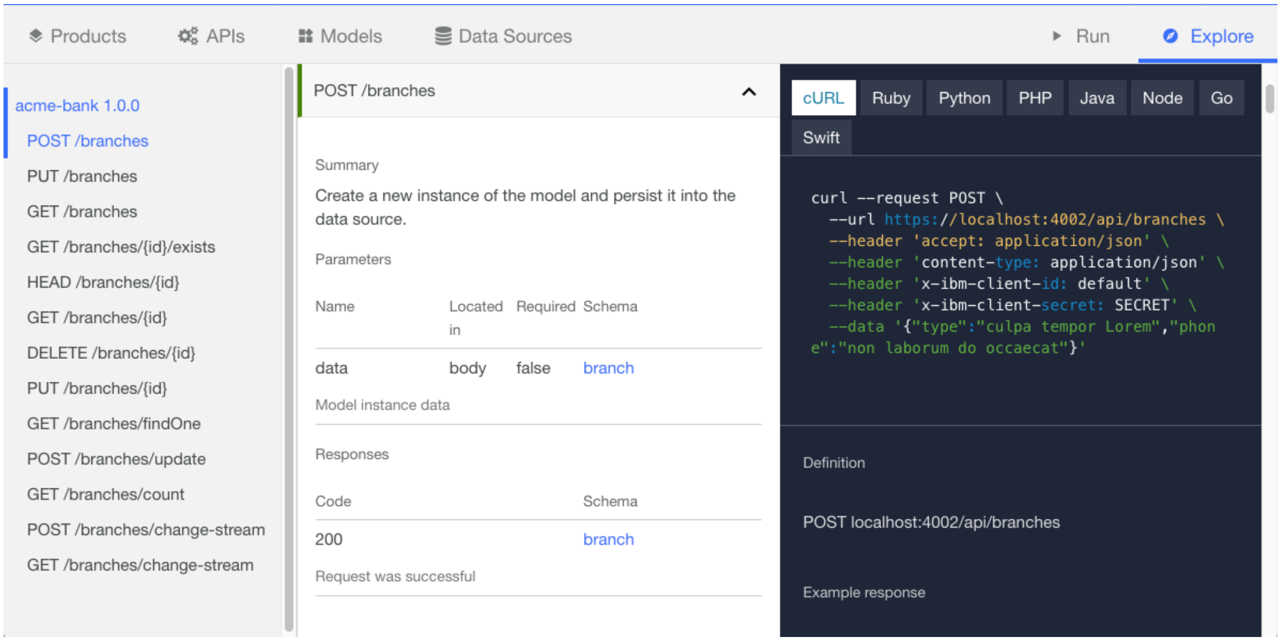
     

   - V5.0.1+ To stop your project, click the Stop the servers icon:
     
     Running | Micro Gateway: https://127.0.0.1:4002/ | Application: http://127.0.0.1:4001/

     To restart it, click the Restart the servers icon:
     
     Running | Micro Gateway: https://127.0.0.1:4002/ | Application: http://127.0.0.1:4001/

   .
5. Click . You will see the API Explore tool. The side bar shows all the REST operations for the LoopBack models in the API. Models that are based on **PersistedModel** by default have a standard set of create, read, update, and delete operations.

6. Click POST /branches in the left pane to display the endpoint to create a new model instance.



POST /branches

Summary

Create a new instance of the model and persist it into the data source.

Parameters

Name	Located in	Required	Schema
data	body	false	branch

Model instance data

Responses

Code	Schema
200	branch

Request was successful

```
curl --request POST \
  --url https://localhost:4002/api/branches \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --header 'x-ibm-client-id: default' \
  --header 'x-ibm-client-secret: SECRET' \
  --data '{"type":"culpa tempor Lorem","phon
e":"non laborum do occaecat"}'
```

Definition

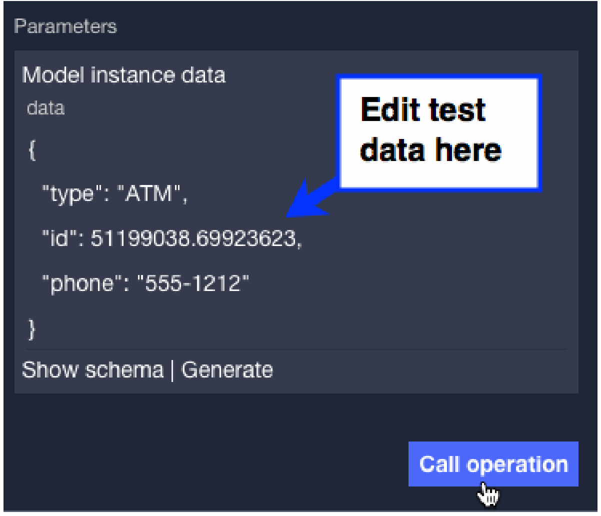
POST localhost:4002/api/branches

Example response

The center pane displays summary information about the endpoint, including its parameters, model instance data, and response codes. The right pane provides template code to call the endpoint using the **curl** command, and languages such as Ruby, Python, Java, and Node.

7. To test the REST endpoints in the API Explore tool, scroll down the right pane and under Parameters click Generate to generate dummy data. By default, the generated data includes only the **type** property, since it was required; the **phone** property is not required, so it's not generated by default (you can add it in the JSON if you wish). Then click Call Operation.
   Note: If you see an error message due to an untrusted certificate for **localhost**, click the link provided in the error message in API Explore to accept the certificate, then proceed to call the operations in your web browser. The exact procedure depends on the web browser you are using.
   If you load the REST endpoints directly in your browser, you will see the message: **{"name":"PreFlowError","message":"unable to process the request"}**. You must use API Explore to test REST endpoints in your browser because it includes the requisite headers and other request parameters.

8. Edit the values in the JSON shown in the Model instance data section. Try changing the generated dummy data, then click Call Operation again.



```
{
  "type": "ATM",
  "id": 51199038.69923623,
  "phone": "555-1212"
}
```

You should see the request and response parameters, along with the JSON instance data that you entered.

9. Test the REST endpoints by using the **curl** command shown, for example:

```
curl --request POST \
  --url https://localhost:4002/api/branches \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --header 'x-ibm-client-id: default' \
  --header 'x-ibm-client-secret: SECRET' \
  --data '{"type":"ATM location"}' -k
```

10. Paste the command into a console window and add **-k** at the end of the command (as shown in the example) to avoid certificate errors. If you wish, edit the JSON data to make it more meaningful. When you enter the command, the console will show the data

entered, for example `{"type":"ATM location","id":1}`.

11. To confirm that the operation added a model instance, click GET /branches then click Call Operation to display all branch instances. For example (with two model instances):

```
[
{
     "type": "standalone",
     "phone": "831-555-1212",
     "id": -76211634.73882793
   },
   {
     "type": "ATM",
     "phone": "408-555-1212",
     "id": -49757341.14205667
   }
 ]
```

12. To display all branch instances, click Call Operation without any filters. For example (with two model instances):

```
[
{
     "type": "standalone",
     "phone": "831-555-1212",
     "id": -76211634.73882793
   },
   {
     "type": "ATM",
     "phone": "408-555-1212",
     "id": -49757341.14205667
   }
 ]
```

You can experiment with other operations if you wish, to get a feeling for the standard REST endpoints of a LoopBack `PersistedModel`.

V5.0.6 + Note: You can now also run the API Explore tool by using the command line. Ensure that your local test servers are running, then run the command **apic explore**. The API Explore tool opens, and shows the operations, definitions, and documentation for all of the APIs that are contained in your project directory. You can specify a single API to explore by specifying the name of the API in the command. Including the option `-e` or `--external` in the command, opens the Explore tool on 0.0.0.0 instead of the default 127.0.0.1. This option binds the server to all IP addresses on the machine, and makes the tool accessible on the wider network.

What you did in this tutorial

In this tutorial, you completed at least one of the following activities:

- Tested a LoopBack project from the command line.
- Tested a LoopBack project from the API Designer Explore tool.

What to do next

Publish your project by following either of these tutorials:

- Tutorial: Publishing a project from the command line
- Tutorial: Staging and publishing a project from the API Designer

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

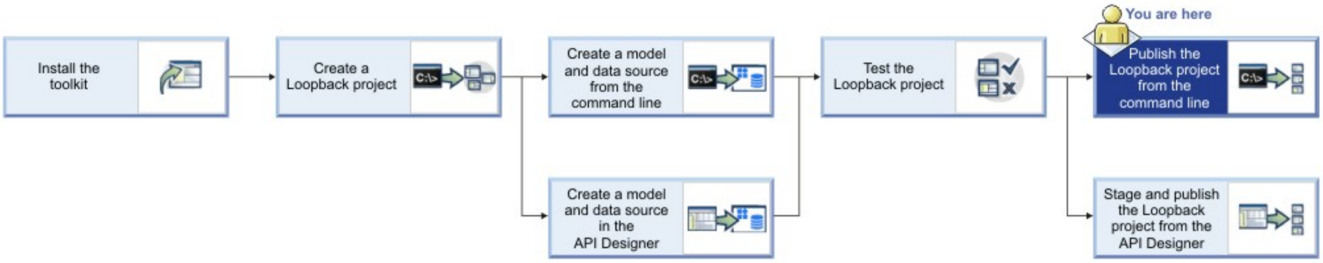
Tutorial: Publishing a project from the command line

This tutorial shows you how to publish a LoopBack® project from the command line in IBM® API Connect Version 5.0.6 and earlier. You publish a LoopBack project to make its APIs available to application developers through the Developer Portal.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.



You are here
Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Test the Loopback project
Publish the Loopback project from the command line
Create a model and data source in the API Designer
Stage and publish the Loopback project from the API Designer

Also ensure your current working directory is the project top-level directory. Enter the following command:

```
cd acme-bank
```

To stage or publish to API Connect collective, you must install and configure an API Connect collective. You must add a Collective and a Collective controller by using the Cloud Manager. See Installing an API Connect collective.

You must also fulfill the following additional prerequisites to stage or publish project:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.
2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.

About this tutorial

In this tutorial, you will publish a LoopBack project to an API Connect collective from the command line.

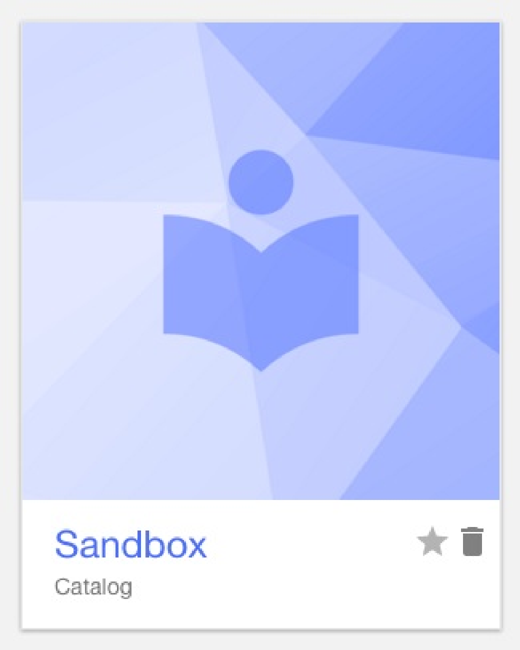
Display general command-line help information by entering the following command: `apic --help` or `apic -h`. Display help information for a specific `apic` command by entering the following command: `apic` ***command_name*** `--help` or `apic` ***command_name*** `-h`.

Publishing a project to API Connect collective from the command line

Be sure you have added the API Connect collective and a Collective controller to the Cloud Manager as described in Installing an API Connect collective.

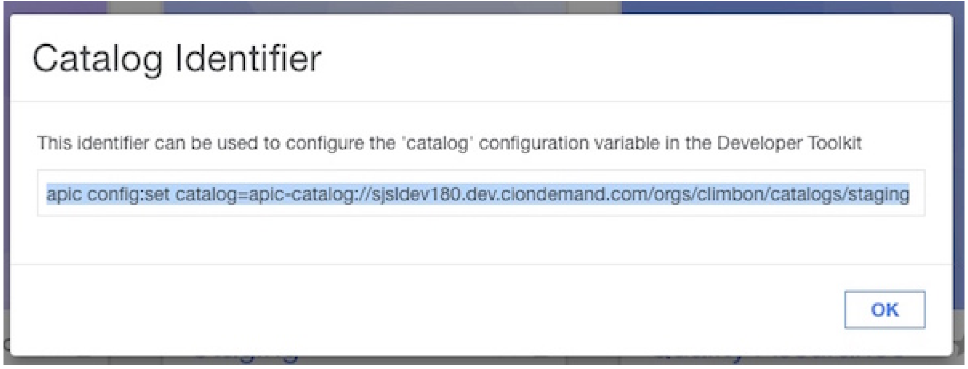
Note: If your application has binary modules (for example, if it uses the DB2®, SQLLite3, or Oracle connectors), then you must create the application on the same type of platform as that of the target API Connect collective.

1. Obtain the identifier for the development Catalog, to which you will publish your project, by completing the following steps in the API Manager user interface:
   a. Log in to the API Manager user interface.
   b. On the Dashboard, by default, you see the development Catalog, called Sandbox.



c. Click the Show catalog identifier icon: ⊂⊃.

d. Copy the Catalog identifier from the Catalog Identifier dialog.



Catalog Identifier

This identifier can be used to configure the 'catalog' configuration variable in the Developer Toolkit

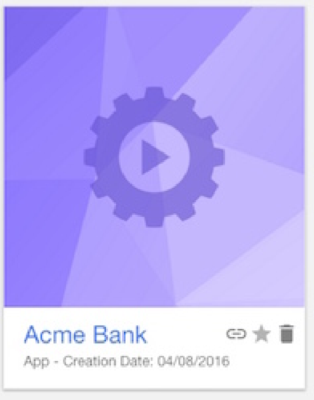
```
apic config:set catalog=apic-catalog://sjsldev180.dev.ciondemand.com/orgs/climbon/catalogs/staging
```

Keep this string handy, because you use it in the following procedure.

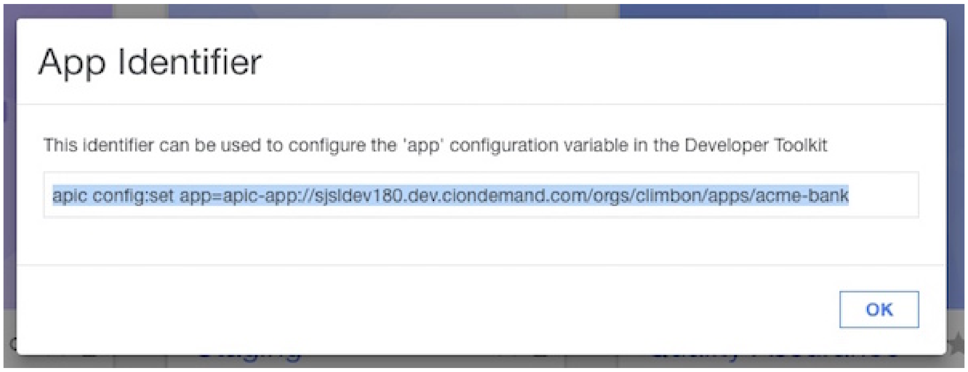
e. Click ⊕ Add > App.

f. In the Add App dialog, enter `Acme Bank` for Display Name, and keep the automatically-generated default value of Name. Select the collective you created previously, then click Add.

g. The API Manager dashboard displays the App tile.



h. Click the Show app identifier icon ⊂⊃ and copy the App identifier from the App Identifier window.



App Identifier

This identifier can be used to configure the 'app' configuration variable in the Developer Toolkit

```
apic config:set app=apic-app://sjsldev180.dev.ciondemand.com/orgs/climbon/apps/acme-bank
```

Keep this string handy, since you use it in the procedure below.

2. Complete the following steps in a command console on the machine where you installed the toolkit:
   a. If you haven't already done so, log in to API Manager by entering the following command:

      ```
      apic login -s API_manager_hostname -u username -p password
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager, *username* is your API Manager user name and *password* is your API Manager password.
   b. Paste the command strings you copied in steps 1.d and 1.h into your command line console. For example:

      ```
      apic config:set catalog=apic-catalog://API_manager_hostname/orgs/climbon/catalogs/sb
      apic config:set app=apic-app://API_manager_hostname/orgs/climbon/apps/acme-bank
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager.
   c. Ensure your current working directory is the project root directory (`acme-bank`). Publish the Product by entering the following command:

      ```
      apic publish -s API_manager_hostname definitions/acme-bank-product.yaml
      ```

      where *API_manager_hostname* is the server name or IP address of the API Manager. The console displays messages that confirm the Product has been published, for example:

      ```
      Staged definitions/acme-bank-product.yaml to climbon:sb [acme-bank1:1.0.0]
      Published definitions/acme-bank-product.yaml to climbon:sb [acme-bank1:1.0.0]
      ```

   d. Publish the application by entering the following command:

      ```
      apic apps:publish
      ```

      The console displays messages such as this:

      ```
      ...preparing project
      ...building package for deploy
      ...uploading packages to 161.51.151.222:9443, scale: 1
      Upload successful: acme-bank-5707fc46e4b07dfbe0999be6-1460148297009-package.tgz
      Upload successful: acme-bank-5707fc46e4b07dfbe0999be6-1460148297009.deploy.xml
      Upload successful: apic.acme-bank-5707fc46e4b07dfbe0999be6-1460148297009.scalingPolicy.xml

      Upload to liberty server completed succesfully.
      Applications may take a few minutes to update and start.
      Collectives admin center: https://161.51.151.222:9443/adminCenter.
      ```

The project is now published to your API Connect collective.

What you did in this tutorial

In this tutorial, you published a project to API Connect collective by using the command-line tools.

What to do next

If you want to use an application with a DataPower® Gateway, configure your collective and gateway and then modify your assembly. For more information, see V5.0.2 + Configuring your DataPower Gateway and API Connect collective controller to communicate and V5.0.2 + Modifying the assembly to call an application endpoint hosted on a collective.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Staging and publishing a project from the API Designer

This tutorial shows you how to stage and publish a project using the API Designer in IBM® API Connect Version 5.0.6 and earlier. *Staging* a project copies all the files to the target, but does not run the project application code. *Publishing* a project copies all the project files to the target and runs the project application code. You publish a LoopBack® project to make its APIs available to application developers through the Developer Portal.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
Before you begin, you must install the developer toolkit on your local machine. For details, see Installing the toolkit.

To stage or publish to API Connect collective, you must install and configure an API Connect collective. You must add a Collective and a Collective controller by using the Cloud Manager. See Installing an API Connect collective.

You must also fulfill the following additional prerequisites to stage or publish project:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.
2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.

You must also do the following:

1. Create a LoopBack project. For more information, see Tutorial: Creating a LoopBack project from the command line.
2. Change directories to your LoopBack project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

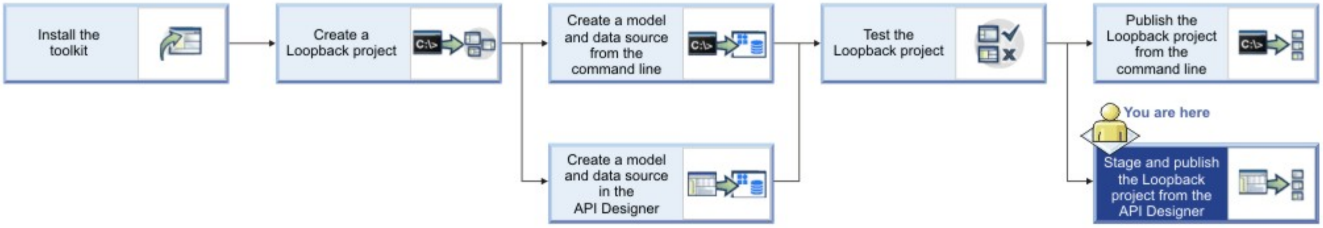
   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with LoopBack projects. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Create a Loopback project
Create a model and data source from the command line
Create a model and data source in the API Designer
Test the Loopback project
Publish the Loopback project from the command line
You are here
Stage and publish the Loopback project from the API Designer

About this tutorial

In this tutorial, you will stage and publish a LoopBack project to an API Connect collective from the API Designer.

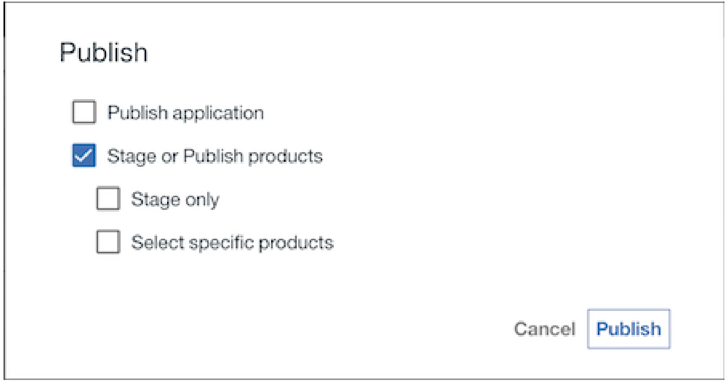
Publishing a project to an API Connect collective

Note: If your app has binary modules (for example, if it uses the DB2®, SQLLite3, or Oracle connectors), then you must create the app on the same type of platform as that of the target API Connect collective.

1. In the API Designer, click Publish then click Add and Manage Targets.

2. Click Add a different target. You'll see the Sign in to IBM API Connect dialog.
3. Enter your API Connect host address, Username, and Password. Click Sign in. You'll see the Select an organization and catalog dialog.
4. Select the desired organization, and select the Sandbox Catalog.
5. Click Next. You'll see the Select an App dialog. Click the desired app, and then click Save.
6. In the API Designer, click Publish again.

7. In the drop-down selection, under Other Orgs, click the target you just created. You'll see the Publish dialog:



Publish

□ Publish application

☑ Stage or Publish products

□ Stage only

□ Select specific products

Cancel Publish

8. Select Publish application and Stage or publish products, then select Select specific products. Select acme-bank, and then click Publish.

While the project is being published, the console displays messages like the following output:

```
Staged /Users/john/acme-bank/definitions/acme-bank-product.yaml to climbon:sb [acme-bank:1.0.0]
Published /Users/john/acme-bank/definitions/acme-bank-product.yaml to climbon:sb [acme-bank:1.0.0]
Successfully published products
...building package for deploy
Creating keys...this may take some time
...uploading packages to 169.53.159.240:9443, scale: 1
Upload successful: acme-bank-57057c0ee4b011906e320bda-1459977334690-package.tgz
Upload successful: acme-bank-57057c0ee4b011906e320bda-1459977334690.deploy.xml
Upload successful: apic.acme-bank-57057c0ee4b011906e320bda-1459977334690.scalingPolicy.xml

Upload to liberty server completed succesfully.
Applications may take a few minutes to update and start.
```

Once the project is published, the API Designer displays a Success message.

If you want to confirm that the application was published in your API Connect collective, you can connect to the Liberty Admin Center by clicking Manage this app from within your App in the API Manager.

What you did in this tutorial

In this tutorial, you published a project to an API Connect collective by using the API Designer.

What to do next

If you want to use an application with a DataPower® Gateway, configure your collective and gateway and then modify your assembly. For more information, see V5.0.2 + Configuring your DataPower Gateway and API Connect collective controller to communicate and V5.0.2 + Modifying the assembly to call an application endpoint hosted on a collective.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials for working with API definitions that call an existing endpoint

Tutorials for creating API definitions to expose and secure BankA APIs in IBM® API Connect Version 5.0.6 and earlier.

Introduction

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
BankA has an existing set of REST-based services that it wants to expose through APIs, to help it grow within the mobile and device application market. The BankA business team knows that an increased mobile and device application presence will promote their brand image and result in increased customer numbers.

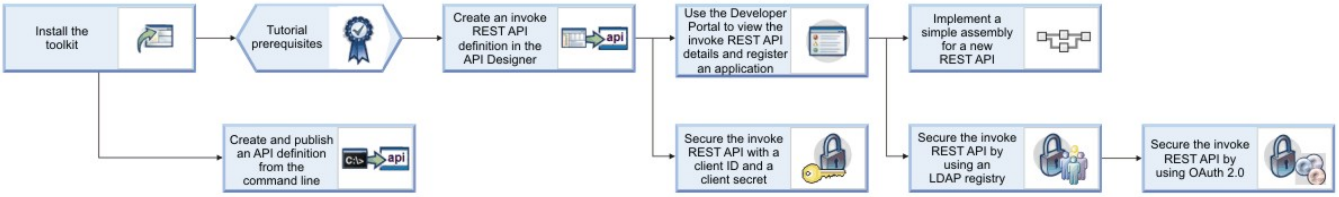
In the following tutorials, you develop the BankA API management solution. The initial solution includes the documentation of a BankA branch information API and the implementation of a pure proxy to access the branch information REST service. This simple proxy allows BankA to monitor the use of the service and set rate limits on the API. You will also document, create, and implement a new operation by using an existing RESTful service.

After you define the API operations, you create and publish a Plan to socialize the API operations.

Learning objectives

During these tutorials you will learn how to create, define, and test an API. You will also learn how to create and test an assembly API, and how to use the Developer Portal. Security options are also demonstrated.

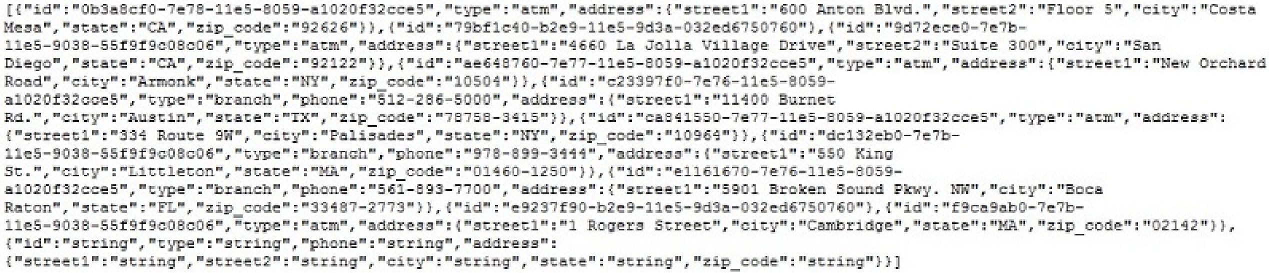
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.



Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

Prerequisites

1. Install the developer toolkit if you have not already done so. For more information, see Installing the toolkit.
2. Open your browser.
3. To ensure that the JSON formatted response of the BankA branch information API is operational, enter the URL **`https://apictutorials.mybluemix.net/branches`** and verify that the API provides a response that is similar to the following example:



```
[{"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":"79bf1c40-b2e9-11e5-9d3a-032ed6750760"},{"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":"e9237f90-b2e9-11e5-9d3a-032ed6750760"},{"id":"f9ca9ab0-7e7b-
11e5-9038-55f9f9c08c06","type":"atm","address":{"street1":"1 Rogers Street","city":"Cambridge","state":"MA","zip_code":"02142"}},
{"id":"string","type":"string","phone":"string","address":
{"street1":"string","street2":"string","city":"string","state":"string","zip_code":"string"}}]
```

To complete the tutorial tasks that involve publishing your API, you must fulfill the following additional prerequisites:

1. You must be the owner of a provider organization, or have been added to a provider organization as a user with the Publisher role. For details on creating a provider organization and specifying the owner, see Creating a provider organization account. For details on adding a user to a provider organization with the Publisher role, see Adding users and assigning roles.

2. You must have followed the instructions in the provider organization email invitation to activate your API Connect account.
3. If you have internet connectivity, to run the API Designer you must have a IBM Cloud account. To create a IBM Cloud account, use the IBM Cloud registration page.

You publish an API to make it available to application developers through the Developer Portal.

- **Tutorial: Creating an invoke REST API definition**
  This tutorial shows you how to define and implement a REST API definition that proxies an existing service in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Securing an API with a client ID and client secret**
  This tutorial shows you how to secure an API so that a calling application must supply a client ID and a client secret in IBM API Connect Version 5.0.6 and earlier. This option is similar to requiring a user ID and password to be supplied.
- **Tutorial: Securing APIs by using an LDAP user registry**
  This tutorial shows you how to secure an API with an LDAP registry so that LDAP credentials must be supplied when the API is called, in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Securing an API by using OAuth 2.0**
  This tutorial shows you how to secure an API by using OAuth 2.0 so that an application can access the API on a user's behalf, in IBM API Connect Version 5.0.6 and earlier.
- **Tutorial: Creating and publishing an API definition from the command line**
  This tutorial shows you how to use the developer toolkit to create an API definition, include it in a Product, and publish the Product to a Catalog in API Manager by using IBM API Connect Version 5.0.6 and earlier.

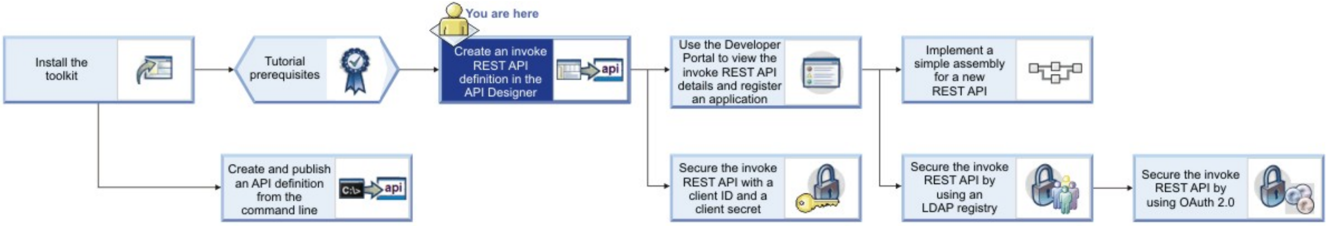
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating an invoke REST API definition

This tutorial shows you how to define and implement a REST API definition that proxies an existing service in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


You are here
Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

In this tutorial you will complete the following lessons:

1. Creating a REST API definition
2. Testing the REST API
3. Creating a Product and a Plan for the REST API
4. Publishing your Product

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. Create a folder to hold your API and Product definitions, and change to that folder in a command window.
2. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

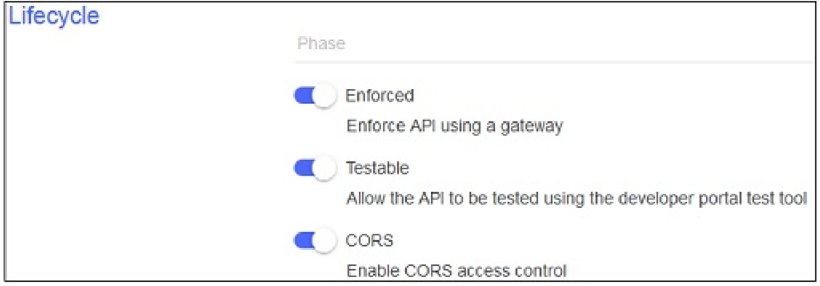
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
3. Log in to APIConnect Designer with the appropriate method.
4. V5.0.6 + Click Add > New API.
   V5.0.4 + Add > New Open API from scratch.
   V5.0.3 and earlier Click Add > API.
5. Enter the appropriate information to create a REST API definition.
   a. In the Title field, enter `Branches`.
   b. The Name and Base Path fields autopopulate with the terms `branches` and `/branches` respectively.
   c. Leave the Version field at `1.0.0`.
   d. Leave the default Additional properties as they are.
6. V5.0.3 and earlier Click Next, ensure that Don't add to a product is selected and then click Add.
7. V5.0.4 + You do not add a product at this time, click Create API.
8. If the API Editor help screen appears, click the sentence Learn more about composing APIs, or click Got it! to access the main screen immediately.
9. In the side bar, click Lifecycle to display the Lifecycle panel. Ensure that the Enforced, Testable, and CORS toggles are set to the On position as shown in the following screen capture:



Lifecycle

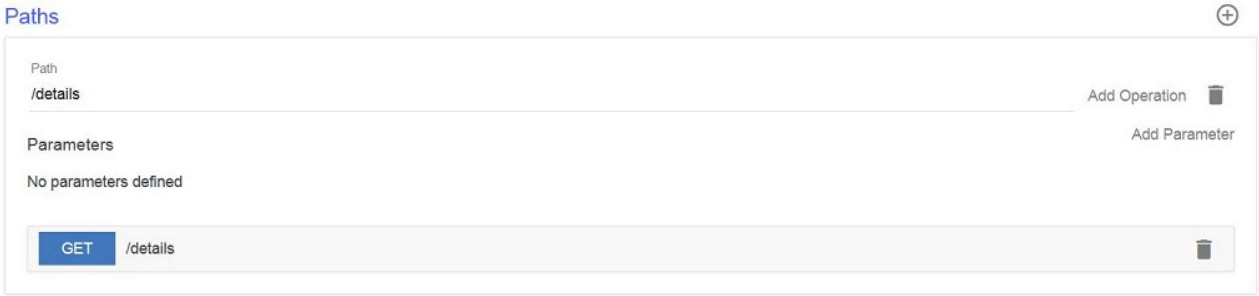
Phase

Enforced
Enforce API using a gateway

Testable
Allow the API to be tested using the developer portal test tool

CORS
Enable CORS access control

10. In the side bar, click Security Definitions to display the Security Definitions panel. Notice that `clientIdHeader` security definition already exists, and in the Security section you see that `Option 1` is active with `clientIdHeader (API Key)`.
11. In the side bar, select Paths to display the Paths panel. Create a new path by clicking the Add Path icon ⊕.
12. In the Path field, replace the default path segment with `/details`. When an operation is called, this path segment is appended to the URL of your API.



Paths

Path
/details

Parameters

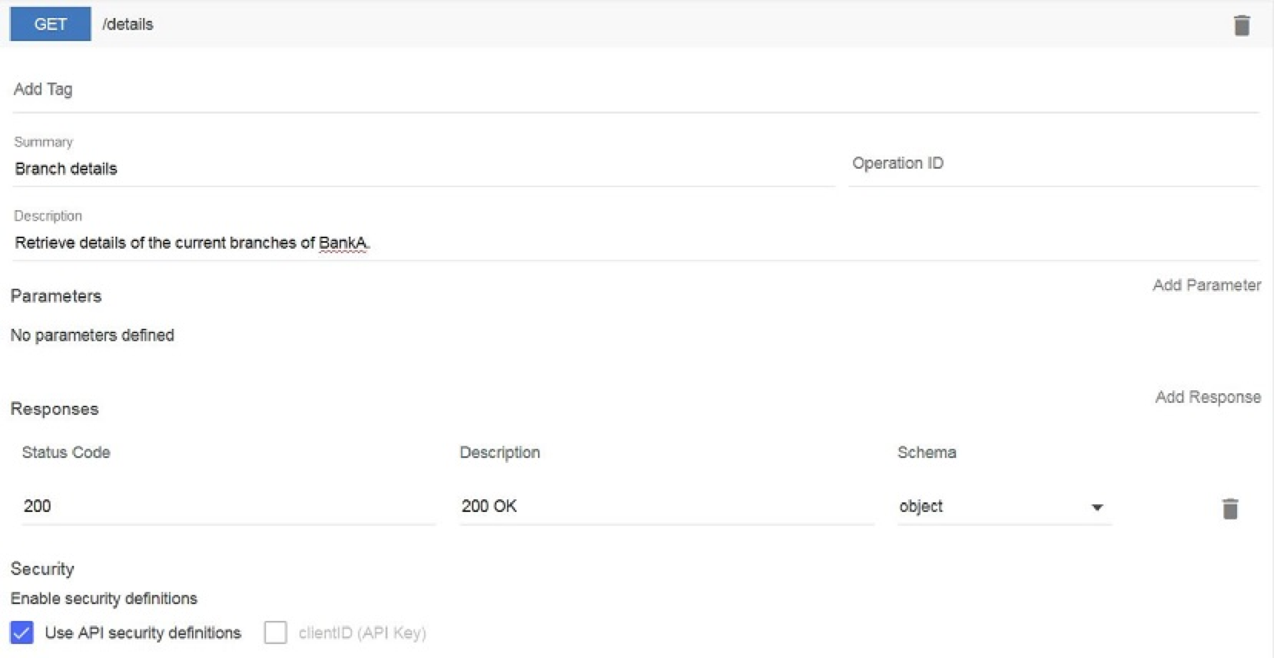
No parameters defined

GET /details

13. By default, a single GET operation is already in your Path. Click the GET button to expand the setting dialog.
14. For the operation, provide a summary and a description as in the following table.

Table 1. Operation definition values

Field	Value
Summary	`Branch details`
Description	`Retrieve details of the current branches of BankA`



GET /details

Add Tag

Summary
Branch details

Operation ID

Description
Retrieve details of the current branches of BankA.

Parameters

Add Parameter

No parameters defined

Responses

Add Response

Status Code	Description	Schema
200	200 OK	object

Security

Enable security definitions

☑ Use API security definitions ☐ clientID (API Key)

15. In the side bar, click Definitions to display the Definitions panel. Add a Definition by clicking the Add Definition icon ⊕.
16. Expand your new definition by clicking new-definition-1. For the Name field, enter `address`, and a Description of `The format of the address object`.
17. Using the same Definitions panel, configure the Properties definition according to the following table. Edit the default property and then create new properties by clicking Add Property and editing the default values.

Table 2. 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`

This is an OpenAPI (Swagger 2.0) 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.

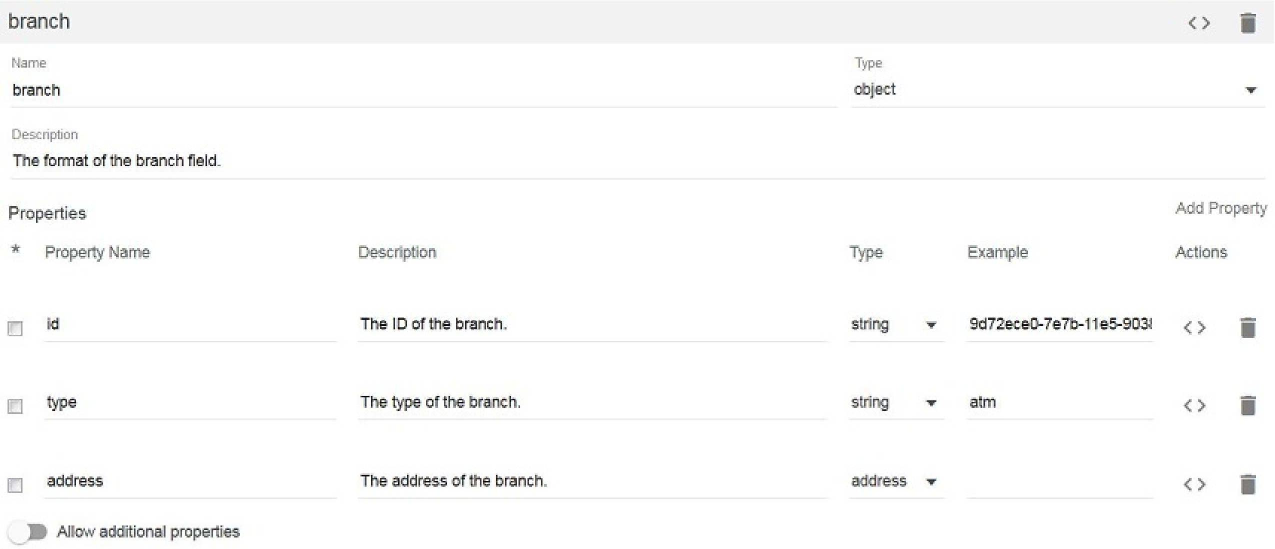
The Required The Required icon column indicates whether a property is required for success if a rest-validate policy uses the definition to perform validation. In this tutorial, no validation is performed and so none of your properties need to be marked as required.

18. Create a second definition by clicking the Add icon ⊕ in the Definitions panel.
19. Name the definition `branch` and, in the Description field, enter `The format of the branch field`.
20. 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 Property.

Table 3. 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`

Notice that for the address property, the type of the property references another definition within your API and the example is left blank. In this manner, you can create complex data structures.



branch

Name: branch

Type: object

Description: The format of the branch field.

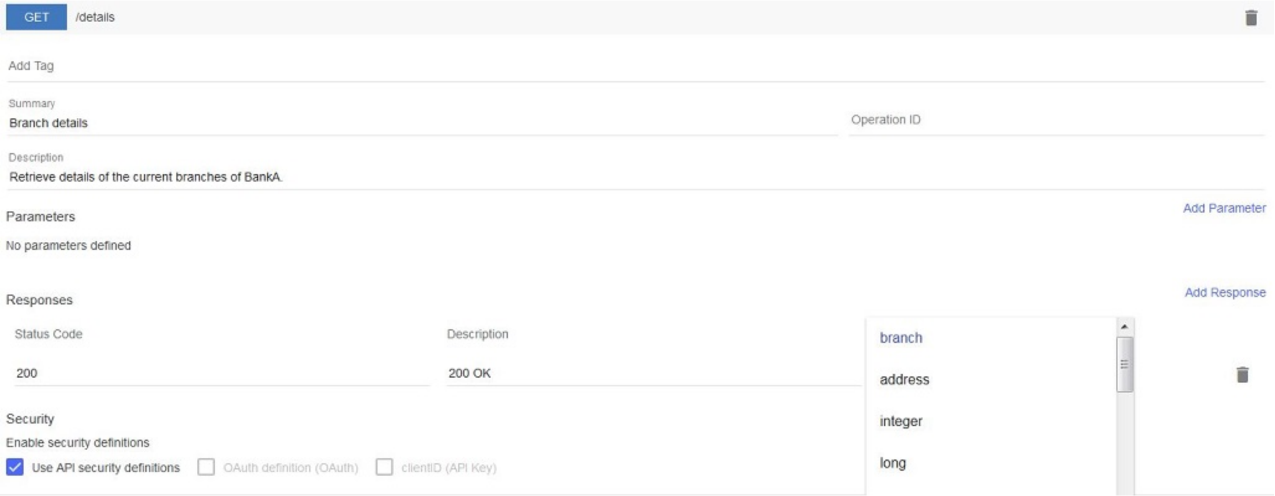
Properties

Add Property

*	Property Name	Description	Type	Example	Actions
□	id	The ID of the branch.	string	9d72ece0-7e7b-11e5-903	
□	type	The type of the branch.	string	atm	
□	address	The address of the branch.	address		

Allow additional properties

21. In the side bar, select Paths to display the Paths panel. For the /details Path, click GET to expand the available settings. Include the branch definition in the GET operation Status Code 200 response by clicking the Schema field and selecting branch from the drop-down list.



GET /details

Add Tag

Summary: Branch details

Operation ID

Description: Retrieve details of the current branches of BankA.

Parameters

Add Parameter

No parameters defined

Responses

Add Response

Status Code: 200

Description: 200 OK

Security

Enable security definitions

☑ Use API security definitions □ OAuth definition (OAuth) □ clientID (API Key)

22. In the submenu navigation bar, click the Assemble tab to open the assemble view.
23. Access the invoke policy property sheet by clicking the invoke label.
24. Populate the Title, Description, and URL fields according to the following table. When called, your API now invokes the existing Branches API and uses its response. In this tutorial, no transformations are applied to the response of this API and so the entirety of the response is returned to the caller. You can see this response at https://apictutorials.mybluemix.net/branches.

Table 4. invoke fields

Field	Value
Title	`Branches Invoke`
Description	`Invoke an API to retrieve the status of all branches in the BankA system`
URL	`https://apictutorials.mybluemix.net/branches`

Leave the remaining fields with their default values.

25. Click the Save icon 💾 to save your changes.
26. Click the Source tab to view the OpenAPI (Swagger 2.0) definition of your API. All the configuration you have performed is included in this definition, either as part of the standard OpenAPI (Swagger 2.0) schema, or as part of the **`x-ibm-configuration`** extension.

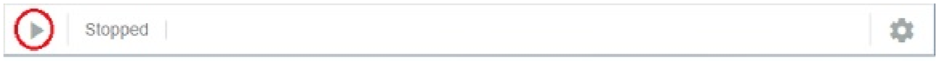
Your REST API is defined. This example helped you to configure the REST API invocation through the Assembly tool. No coding was required. The definitions help developers who are creating applications and integrating with the BankA Branches REST API for the first time.

Testing the REST API

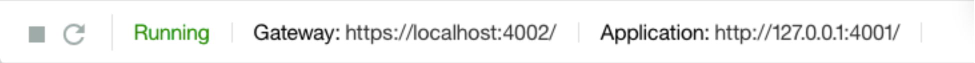
Test your REST API to ensure that it is defined and implemented correctly.

To test the REST API, complete the following steps:

1. V5.0.1+ Start the local test servers by completing the following steps:
   a. In the test console at the bottom of the screen, click the Start the servers icon:

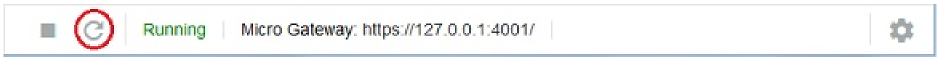


   b. Wait until the `Running` message is displayed:


Running
Gateway: https://localhost:4002/
Application: http://127.0.0.1:4001/

   Depending on your project configuration and whether other processes are running, a different port number might be displayed.

   Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking the Restart the servers icon



   .
2. Start the local test servers by completing the following steps:
   a. In the API Designer, click Run.
   b. Click Start to run your Micro Gateway locally and host your APIs on it. A node.exe window opens when your Micro Gateway is running; leave the window open until you have finished testing your API.
      Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking Restart.
   c. Wait until the `Running` message is displayed and then click the APIs tab.
3. V5.0.0 ONLY Click the Branches API to show the details of the Branches API.
4. Click the Assemble tab.
5. Click the Test icon ▶. The test tool opens, overlaying the palette.
6. In the Operation section, use the drop-down menu to select the get /details operation.
7. At the bottom of the section, click Invoke. The operation is called by the test tool. The response of your API is shown in the test tool.

In other tutorials you get a chance to test the API by using the Developer Portal and API Manager test tools, both of which run online.

Creating a Product and a Plan for the REST API

Create a Product and a Plan so that you can later stage or publish your APIs.

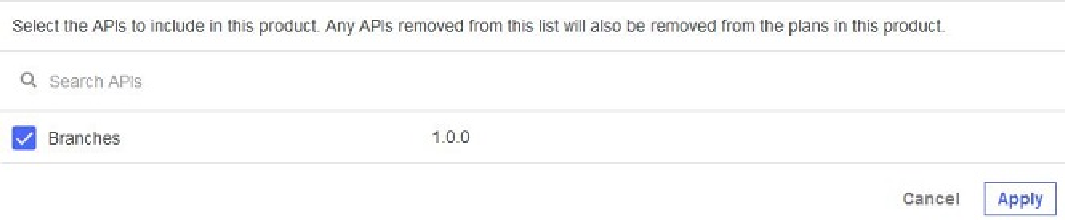
To create a Product, complete the following steps:

1. Click All APIs and then click the Products tab.
2. V5.0.4 + Click Add and then click New product from scratch.
   V5.0.3 and earlier Click Add and then click New Product. The "Add a new product" window opens.
3. Complete the fields as shown in the following table and then click V5.0.4 + Create product.

   Table 5. Product fields

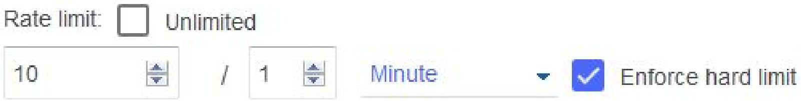
Field Name	Value
Title	`Banking Services`
Name	`banking-services`
Version	`1.0.0`

4. In the Visibility section you can control who the Product is visible to and who can subscribe to its Plans. The Product visibility is set to "Public" and so anybody will be able to see the Product when it is published to the Developer Portal. When published, the Plans can be subscribed to by "Authenticated users", which refers to users who have accounts in the Developer Portal.
5. In the APIs section, click the Add API icon ⊕. The Select APIs window opens.
6. Select the Branches API.


Select the APIs to include in this product. Any APIs removed from this list will also be removed from the plans in this product.

7. Click Apply.
8. Expand the Plan titled Default that has been automatically created. Because no APIs have been excluded, the only API in the Product is included in the Plan.
9. Enter `Basic` for the Title field and the Name field.
10. Add a rate limit to your /details operation by completing the following steps:
    V5.0.0 ONLY V5.0.1 ONLY
    a. Hover the cursor over your Branches API and click the Show operations icon ••• that appears.

b. Click the Edit rate limit icon The Rate limit icon beside the get /details operation.
c. Clear the Unlimited check box under Rate limit.
d. Use the controls to set the rate limit as 10 requests per 1 minute and select Enforce hard limit.

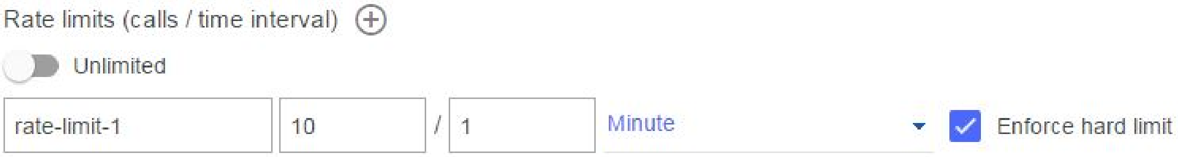


Rate limit: ☐ Unlimited

10 / 1 Minute ☑ Enforce hard limit

V5.0.2 +

a. Hover the cursor over your Branches API and click the Show operations icon .
b. Click Override rate limit beside the get /details operation.
c. Use the controls to set the rate limit as 10 requests per 1 minute against rate-limit-1, and select Enforce hard limit.



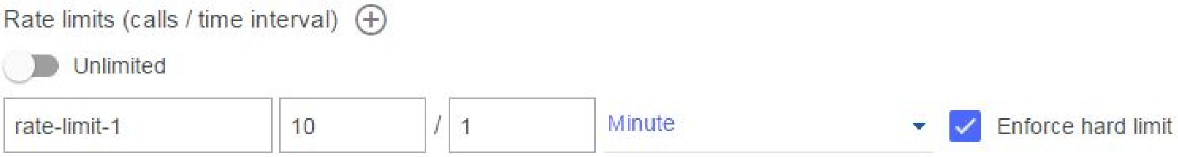
Rate limits (calls / time interval) ⊕

Unlimited

rate-limit-1 | 10 / 1 Minute ☑ Enforce hard limit

V5.0.4 +

a. Expand the Branches API.
b. Click Override rate limit beside the get /details operation.
c. Use the controls to set the rate limit as 10 requests per 1 minute against rate-limit-1, and select Enforce hard limit.



Rate limits (calls / time interval) ⊕

Unlimited

rate-limit-1 | 10 / 1 Minute ☑ Enforce hard limit

11. Click the Save icon to save your changes.

You have created the Banking Services Product with the Basic Plan within it, you added the Branches API to the Basic Plan, added a rate limit to the `rate-limit-1` operation, and staged your Banking Services Product to your development environment.
Note: These steps are not necessary to test your APIs offline, but a Product is needed when making your APIs externally available.

Publishing your Product

Publish your Product and the API it contains to make them externally available for later tutorials.

1. In the API Designer, click Publish and then click Add and Manage Targets.
2. Click Add a different target.
3. In the API Connect host address field, enter the address of your Management server, for example, `example.host.com`.
4. Provide a user name and password for an API Manager account on your server and then click Sign in.
5. In the Organization field, select the provider organization that you want to publish with.
6. Select Sandbox from the list of Catalogs. If you have a large number of Catalogs, use the Search field to refine the list of Catalogs.
7. Click Save.
8. Click Publish and then click your newly created target.
9. Select Select specific products and then select your Banking Services Product.
10. Click Publish. Your Product is now available through your gateway server and visible in both API Manager and the Developer Portal,

Your Product and the API it contains are published to your specified target.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a REST API definition.
- Tested a REST API.
- Created a Product that contains a Plan.
- Published a Product to a Catalog.

What to do next

- Discover and use your API in the Developer Portal.

- Secure your API with a client ID and secret.
- Secure your API by using an LDAP registry.
- Secure your API by using OAuth.

Related concepts

- Working with Products in the API Designer

Related tasks

- Creating API definitions
- Testing an API with the API Designer test tool

Related information

- IBM API Connect overview
- Creating APIs by using the API Designer
- Using the Developer Portal

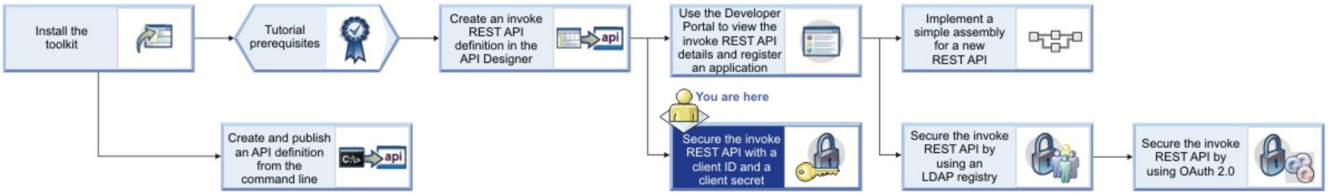
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Securing an API with a client ID and client secret

This tutorial shows you how to secure an API so that a calling application must supply a client ID and a client secret in IBM® API Connect Version 5.0.6 and earlier. This option is similar to requiring a user ID and password to be supplied.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
You are here
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

You will modify the security settings for the Branches API, which you created in the tutorial Tutorial: Creating an invoke REST API definition, so that a calling application must supply a client ID and a client secret, then you will attempt to call the Branches API with and without the client ID and client secret, to verify that the client ID and client secret are required.

You will complete the following lessons:

1. Setting the identification mechanism of an API
2. Calling an API by using a client ID and client secret

Setting the identification mechanism of an API

To modify the security settings for the Branches API so that a calling application must supply a client ID and client secret, complete the following steps:

1. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

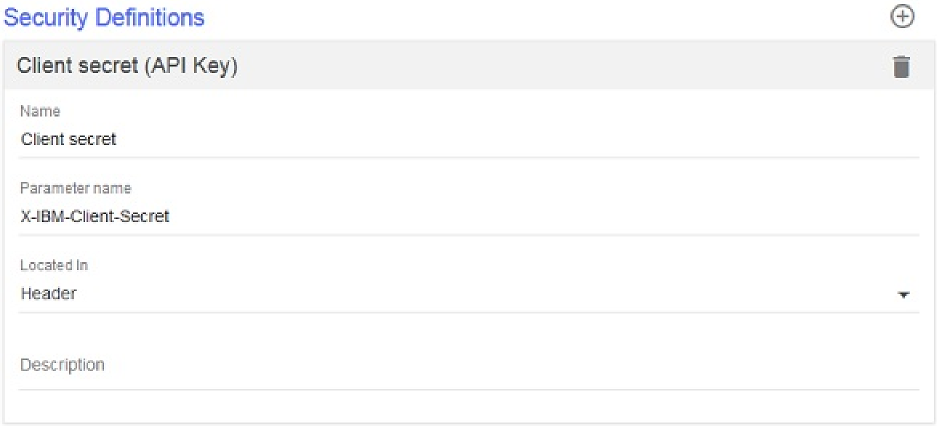
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
2. Click APIs, then click the Branches REST API that you created in the tutorial [Tutorial: Creating an invoke REST API definition](#).
3. Navigate to the Security Definitions section.
4. Note that, by default, a Client ID security definition already exists for your API.
5. Click the Add Security Definition icon ⊕ in the Security Definitions section, then select API Key. A new API Key security definition displays in the Security Definitions section.
6. Change the value of the Name field to `Client secret`.
7. Leave the value of the Parameter Name field as X-IBM-Client-Secret. You have defined a new security scheme.

   Security Definitions

   Client secret (API Key)

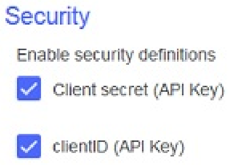
   Name
   Client secret

   Parameter name
   X-IBM-Client-Secret

   Located In
   Header

   Description

8. Navigate to the Security section.
9. In the Security section, select Client secret (API Key), in addition to Client ID (API Key) which should already be selected by default.

   Security

   Enable security definitions

   ☑ Client secret (API Key)

   ☑ clientID (API Key)

10. Click the Save icon 💾 to save your changes.

You have modified the operation so that a calling application must supply a client ID and client secret.

Calling an API by using a client ID and client secret

Now that you have determined the client ID and client secret for the Baggage Tracker application, you can supply them when calling the BankA API. For the purposes of this tutorial, you call the Branches API by testing it in the API Designer Explorer.

To call the Branches API by using a client ID and client secret, complete the following steps:

1. V5.0.0 ONLY Start the local test servers by completing the following steps:

a. Click Run.

b. Click Start.

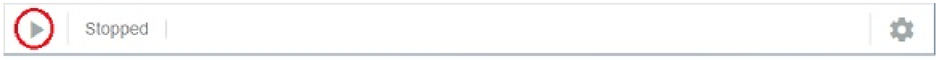
c. Wait until the **`Running`** message is displayed:

Depending on your project configuration and whether other processes are running, different port numbers might be displayed.

Note: If your Micro Gateway is already running, you must restart it before you can test your changes.

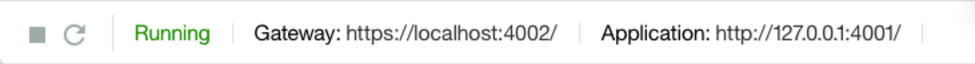
2. V5.0.1+ Start the local test servers by completing the following steps:

a. In the test console at the bottom of the screen, click the Start the servers icon:



Stopped

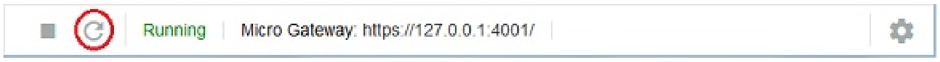
b. Wait until the **`Running`** message is displayed:



Running | Gateway: https://localhost:4002/ | Application: http://127.0.0.1:4001/

Depending on your project configuration and whether other processes are running, a different port number might be displayed.

Note: If your Micro Gateway is already running, you must restart it before you can test your changes, by clicking the Restart the servers icon

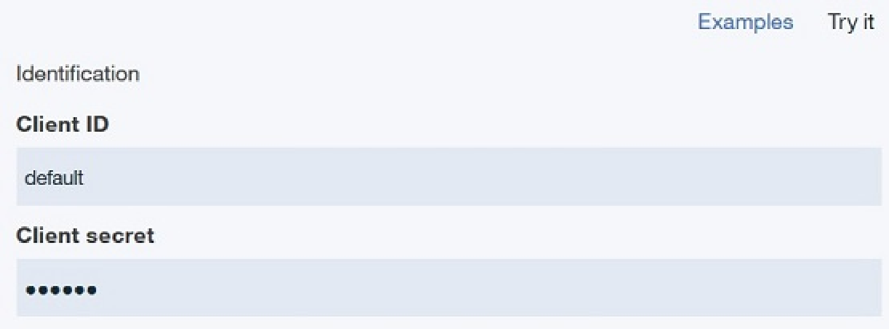


Running | Micro Gateway: https://127.0.0.1:4001/

.

3. Click Explore and scroll down to the Identification section in the pane on the right.

The Client ID field contains the value `default`, and the Client secret field contains the value `SECRET`; these are the default values that are used for testing in the API Designer Explorer.



Identification

Client ID

default

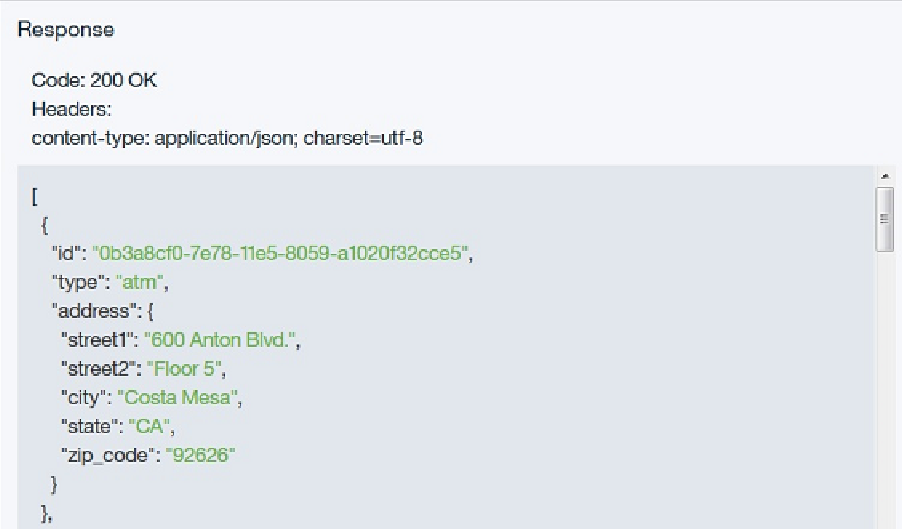
Client secret

••••••

When the API is published and becomes available to application developers through the Developer Portal, the API will be called by using application specific client ID and client secret values; for more information, see [Adding an application]().

4. Remove the client ID and client secret values and click Call operation to test the API. The call fails.

5. Restore the client ID and client secret value by entering `default` in the Client ID field and `SECRET` in the Client secret field, and click Call operation to test the API. The Branches response is returned correctly:



Response

```
Code: 200 OK
Headers:
content-type: application/json; charset=utf-8
```

```
[
 {
  "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5",
  "type": "atm",
  "address": {
   "street1": "600 Anton Blvd.",
   "street2": "Floor 5",
   "city": "Costa Mesa",
   "state": "CA",
   "zip_code": "92626"
  }
 },
```

What you did in this tutorial

In this tutorial, you completed the following activities:

- Set the identification mechanism of an API.
- Called an API by using a client ID and client secret.

Related concepts

- Working with Products

Related information

- Creating APIs by using the API Designer
- Tutorial: Creating an invoke REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

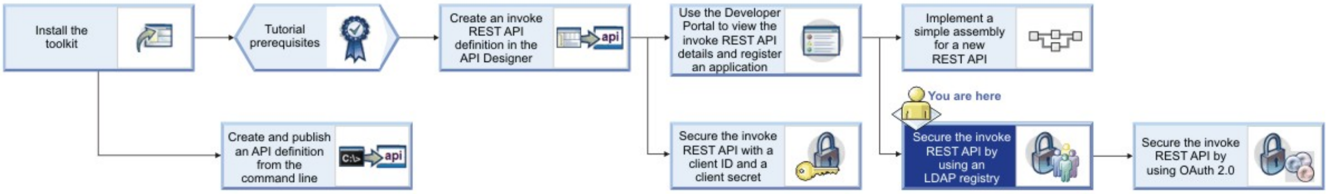
Tutorial: Securing APIs by using an LDAP user registry

This tutorial shows you how to secure an API with an LDAP registry so that LDAP credentials must be supplied when the API is called, in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial assumes that you have an available working LDAP registry in which you have a registered user ID, and to which you know the connection details.

The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
You are here
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

You will complete the following lessons:

1. Creating an LDAP user registry definition in API Manager
2. Securing an API with an LDAP registry
3. Republishing a Product
4. Resubscribing to the Basic Plan
5. Testing a secured API in the Developer Portal

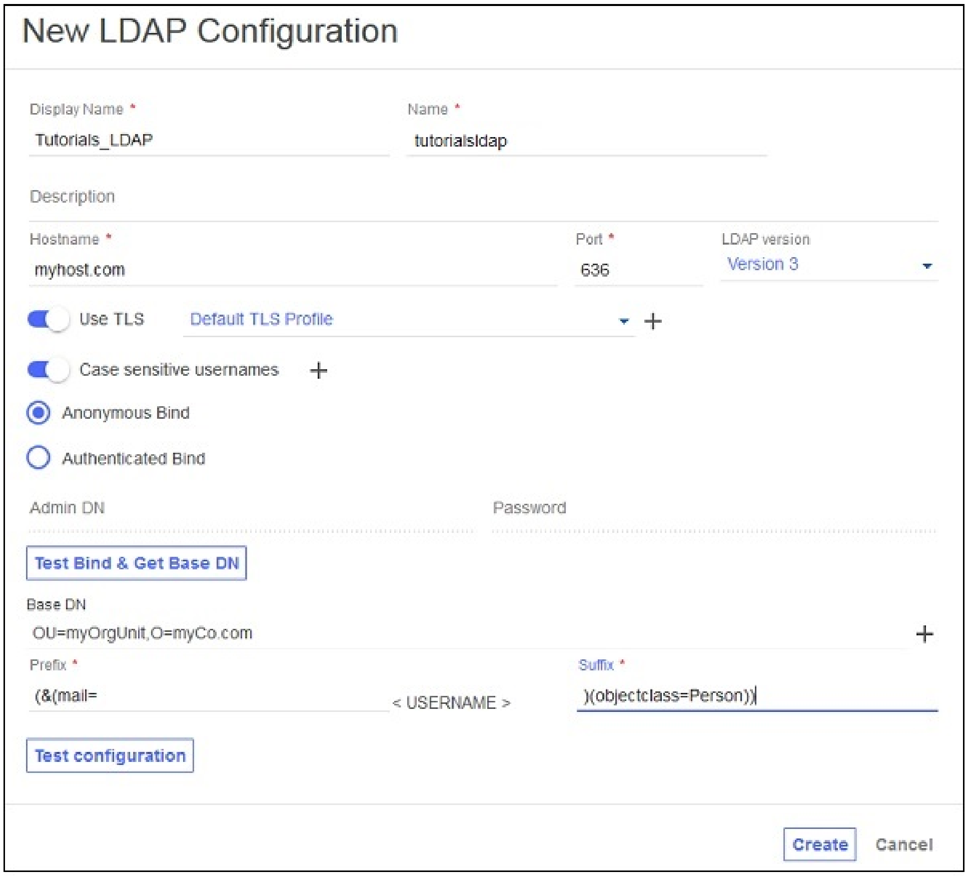
Creating an LDAP user registry definition in API Manager

To secure APIs with an LDAP registry, you must first define the connection details for the LDAP registry by creating an LDAP user registry definition in API Manager.

If an LDAP user registry definition has already been created, obtain the name of that LDAP user registry definition and proceed to Securing an API with an LDAP registry, otherwise complete the following steps.

1. Sign in to the API Manager user interface.

2. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
3. V5.0.4 and earlier Click Admin in the navigation pane, then click User Registries.
4. V5.0.5 + In the navigation pane, select Admin and select the Security tab.
5. V5.0.5 + Click User Registries.
6. Click Add > LDAP Registry. The New LDAP Configuration window opens.
7. In the New LDAP Configuration window, complete the following steps, as applicable for your LDAP registry:
   a. Enter `Tutorials_LDAP` in the Display Name field and `tutorialsldap` in the Name field and, optionally, enter a description.
   b. Enter the host name and port.
   c. Click the Version drop-down list and select the registry version.
   d. To protect user credentials, move the Use TLS slider to ON.
      Note:
      - If you want to use this option, TLS must be enabled on your LDAP server.
      - If the Use TLS slider is not set to ON, user credentials are not protected in transit to the LDAP user directory.
   e. Select Anonymous bind or Authenticated bind.
   f. If required, enter, in the Admin DN and Password text fields, the distinguished name and password of a user that has access to connect to the server and read all user and group data; for example, a built-in administrator account.
   g. In the Base DN text field, enter the Base DN information. If you are unsure of the correct value, click Test Bind & Get Base DN to display a drop-down menu of Base DNs available for the configured LDAP server.
   h. Enter the Prefix and Suffix information.
   i. When you are done, click Test configuration. If the test is successful, a confirmation message is displayed. If the test is not successful, an error message is displayed. Recheck your settings and run the test again.
   j. Click Create to create the registry. The following screen capture shows a sample configuration; you must ensure that you enter the details applicable to your LDAP registry:



New LDAP Configuration

Display Name *: Tutorials_LDAP

Name *: tutorialsldap

Description

Hostname *: myhost.com

Port *: 636

LDAP version: Version 3

Use TLS: Default TLS Profile

Case sensitive usernames

● Anonymous Bind

○ Authenticated Bind

Admin DN

Password

Base DN: OU=myOrgUnit,O=myCo.com

Prefix *: (&(mail= < USERNAME > Suffix *:)(objectclass=Person))

Note: You can view or edit the details of your LDAP registry configuration by completing the following steps:

1. V5.0.4 and earlier Click Admin in the navigation pane, then click User Registries.
2. V5.0.4 and earlier Select your LDAP registry in the side pane.
3. V5.0.5 + In the navigation pane, select Admin and select the Security tab.
4. V5.0.5 + Click User Registries.
5. V5.0.5 + Click the manage icon ⋮ for the user registry that you want to edit, then click Edit user registry. The details of the registry are displayed.
6. Modify the configuration details as required, then click Save to save your changes.

Securing an API with an LDAP registry

You will use the API Designer user interface to secure the Branches API with the Tutorials_LDAP registry that you defined previously. The Branches API was created and configured in the tutorial Tutorial: Creating an invoke REST API definition.

1. In a command window, change to the project folder that you created in the tutorial Tutorial: Creating an invoke REST API definition.
2. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

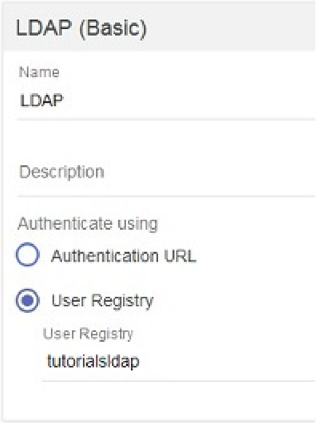
   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

   where *port_number* is the port number to use.
3. Click APIs, then click the Branches REST API that you created in the tutorial Tutorial: Creating an invoke REST API definition.
4. Navigate to the Security Definitions section.
5. In the Security Definitions section click the Add Security Definition icon ⊕.
6. Select Basic. A new Basic security definition displays in the Security Definitions section.
7. Change the value of Name to `LDAP`.
8. Select User Registry and, in the User Registry field, enter `tutorialsldap` to match the name that you gave to the LDAP registry you created in Creating an LDAP user registry definition in API Manager, or if an LDAP user registry definition has already been created enter the corresponding name.

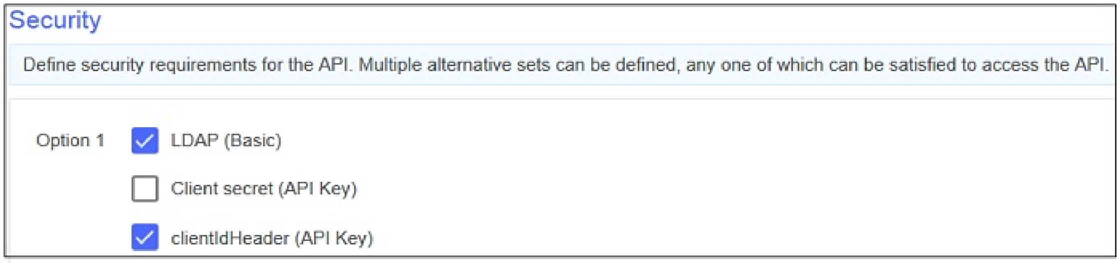
   LDAP (Basic)

   Name
   LDAP

   Description

   Authenticate using
   ○ Authentication URL
   ● User Registry
   User Registry
   tutorialsldap

9. If you have previously completed the tutorial Tutorial: Securing an API with a client ID and client secret, clear the selection for the Client secret security definition.
10. Navigate to the Security section.
11. Select LDAP to apply the LDAP security definition to the API.

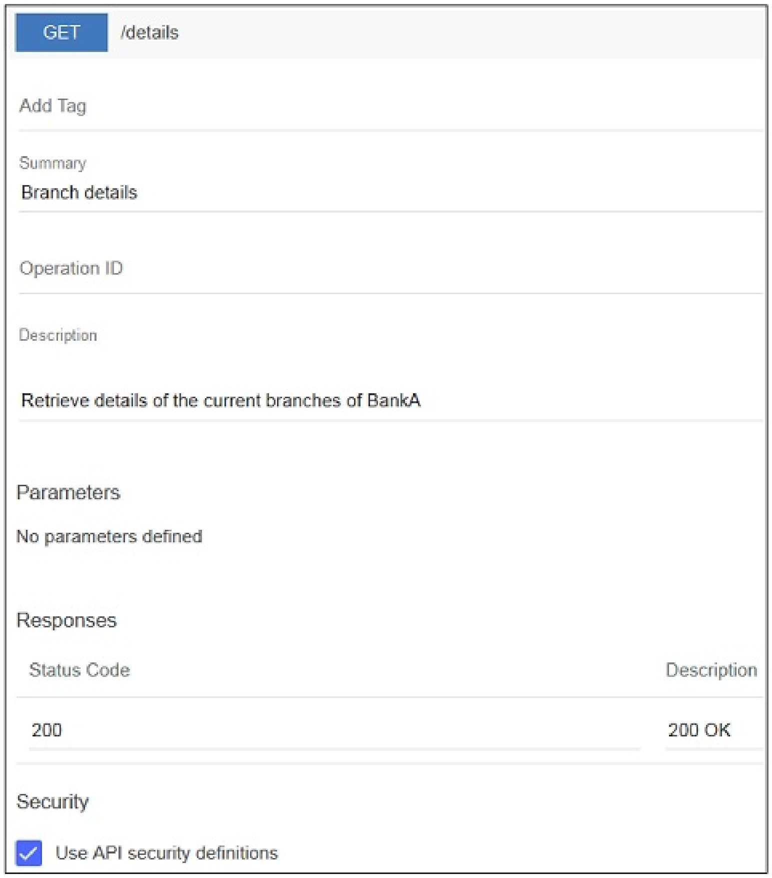
    

    Security

    Define security requirements for the API. Multiple alternative sets can be defined, any one of which can be satisfied to access the API.

    Option 1 ☑ LDAP (Basic)
    □ Client secret (API Key)
    ☑ clientIdHeader (API Key)

12. Navigate to the Paths section, click the GET method to expand its details, and ensure that Use API security definitions is selected; this setting means that all the security definitions that are defined for the API are inherited by the method.



GET /details

Add Tag

Summary
Branch details

Operation ID

Description
Retrieve details of the current branches of BankA

Parameters
No parameters defined

Responses

Status Code	Description
200	200 OK

Security

☑ Use API security definitions

13. Click the Save icon 💾 to save your changes.

You have now secured the Branches API so that the user ID and password of a user in the Tutorials_LDAP registry must be supplied when the API is called.

Republishing a Product

For the security changes you have made to take effect, you must republish the Banking Services Product that contains the Branches API and Basic Plan.

To republish the Banking Services Product, complete the following steps:

1. Click Publish and then click the target that you created in the tutorial Tutorial: Creating an invoke REST API definition, for example Sandbox.
2. Select Select specific products and then select your Banking Services Product.
3. Click Publish.

Your Product and the API it contains are published to your specified target.

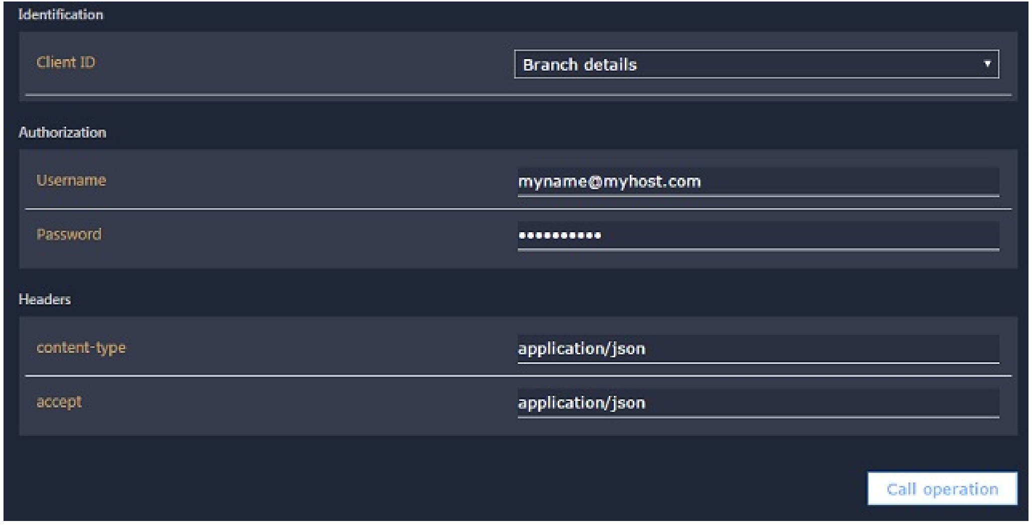
Resubscribing to the Basic Plan

When a Product is republished, all subscriptions to the Plans in the Product are removed. You must now resubscribe to the Basic Plan. Note: If your republish target was the Sandbox Catalog, you do not need to resubscribe to the Basic Plan, as subscriptions to a development Catalog automatically continue.

1. Sign in to the Developer Portal.
2. Click API Products in the Developer Portal dashboard.
3. Click the Banking Services Product. The details of the Plan named Basic are displayed.
4. Click Subscribe.
5. Under the Application heading, select the Branch details radio button.
6. Click Subscribe.

Testing a secured API in the Developer Portal

1. In the Developer Portal, click API Products > Banking Services > Branches > GET /details. The GET /details operation is displayed.
2. In the "Try this operation" section of the console, use the drop-down menu to select the Branch details application.
3. Because you have secured the Branches API with an LDAP registry, an Authorization section is displayed. Enter your LDAP user name and password, then click Call operation.



Identification

Client ID: Branch details

Authorization

Username: myname@myhost.com

Password: ••••••••••

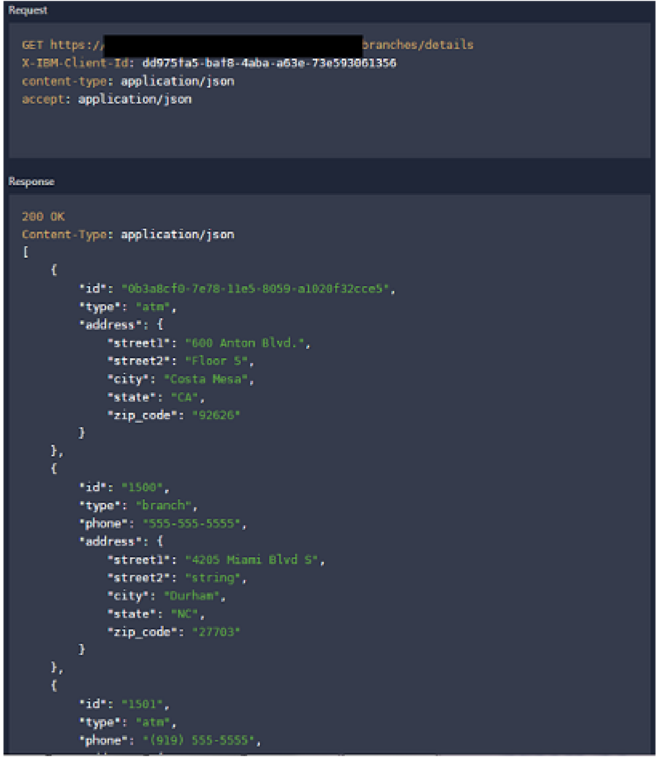
Headers

content-type: application/json

accept: application/json

Note: If you receive no response, 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.

4. A returned response of `200 OK` and the message body are displayed, indicating that the REST API operation call was successful.



Request

```
GET https:/[redacted]branches/details
X-IBM-Client-Id: dd975fa5-baf8-4aba-a63e-73e593061356
content-type: application/json
accept: application/json
```

Response

```
200 OK
Content-Type: application/json
[
    {
        "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": "1500",
        "type": "branch",
        "phone": "555-555-5555",
        "address": {
            "street1": "4205 Miami Blvd S",
            "street2": "string",
            "city": "Durham",
            "state": "NC",
            "zip_code": "27703"
        }
    },
    {
        "id": "1501",
        "type": "atm",
        "phone": "(919) 555-5555",
```

What you did in this tutorial

In this tutorial, you completed the following activities:

- Secured a Catalog with an LDAP registry.
- Secured an API with an LDAP registry.
- Republished a Product to a Catalog.
- Registered an application to call an API.
- Tested a secured API in the Developer Portal.

Related information

- IBM API Connect overview

- Using the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only

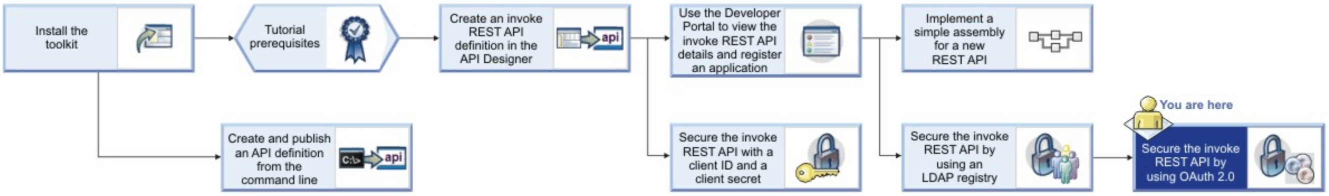
Tutorial: Securing an API by using OAuth 2.0

This tutorial shows you how to secure an API by using OAuth 2.0 so that an application can access the API on a user's behalf, in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
To complete this tutorial, you need an environment capable of sending HTTP requests and receiving HTTP responses. In these instructions, the **curl** command is used in a command line interface to demonstrate the OAuth flow without the need to write any application code. When you implement the OAuth flows for your application, make the same HTTP REST calls as used with **curl** in this tutorial.

Note: The commands and example commands in this tutorial might need to be adapted for the syntax of your own command line interface.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
You are here
Secure the invoke REST API by using OAuth 2.0

Note: You can secure your APIs with a third-party OAuth provider. For more information, see Integrating third party OAuth provider.

About this tutorial

You will modify the security settings for the Branches API, which you created in the tutorial Tutorial: Creating an invoke REST API definition, so that a calling application can make use of OAuth 2.0 to access the API on behalf of a user without requiring the user's password.

In this tutorial you will complete the following lessons:

- Choosing your OAuth Scheme
- Creating an OAuth 2.0 provider API
- Configuring the API Security Scheme
- Acquiring Access Tokens for Individual Schemes
- Using the Access Token

In OAuth 2.0, the following three parties are involved:

- The user, who possesses data that is accessed through the API and wants to allow the application to access it
- The application, which is to access the data through the API on the user's behalf
- The API, which controls and enables access to the user's data

Using OAuth 2.0, it is possible for the application to access the user's data without the disclosure of the user's credentials to the application.

The API will grant access only when it receives a valid access token from the application. How the application obtains an access token is dependent upon the OAuth scheme that is in use.

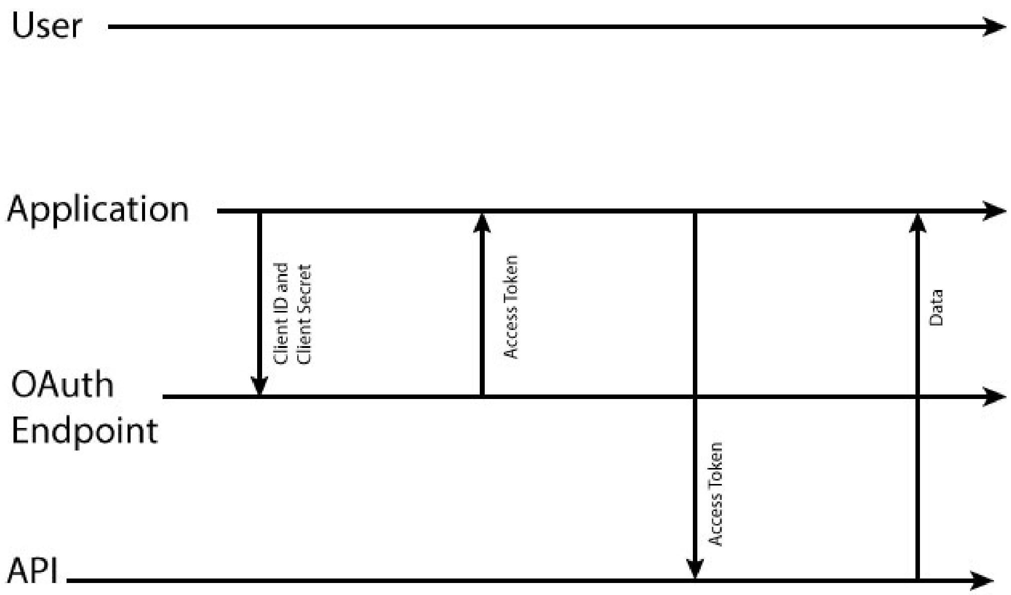
In this tutorial, you will be able to implement and test any of the following six OAuth schemes: implicit flow, application flow, confidential password flow, public password flow, confidential access code flow, and public access code flow. More information about these schemes is available in the following section.

Choosing your OAuth scheme

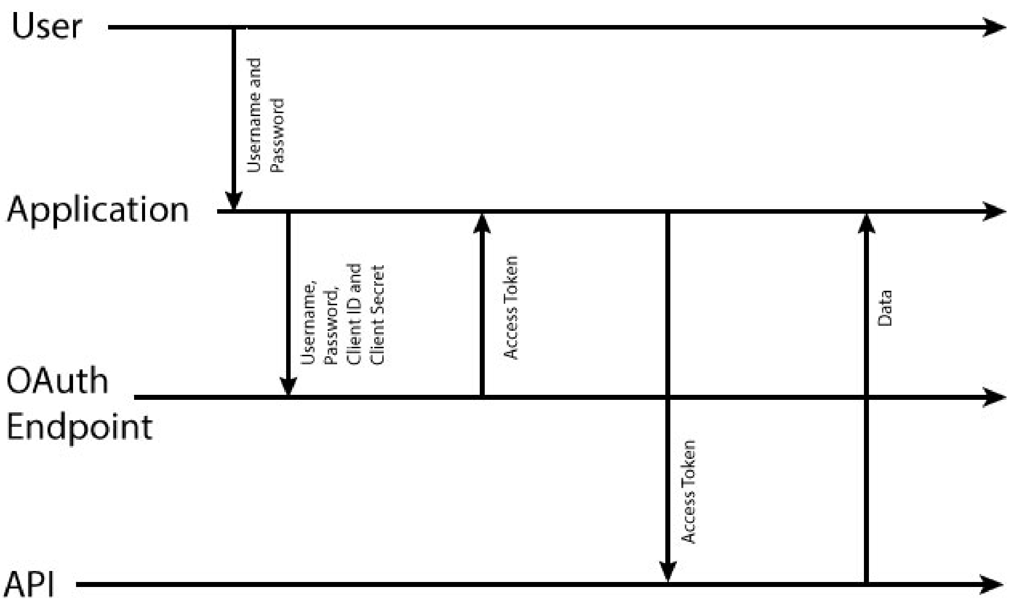
If you already know which OAuth scheme you intend to use, skip this section and proceed to Creating an OAuth 2.0 provider API.

To choose an OAuth scheme, you must first establish whether your implementation is considered public or confidential. This will narrow your choices to three schemes. A brief outline of each scheme and the characteristics of the three public and three confidential schemes follows:

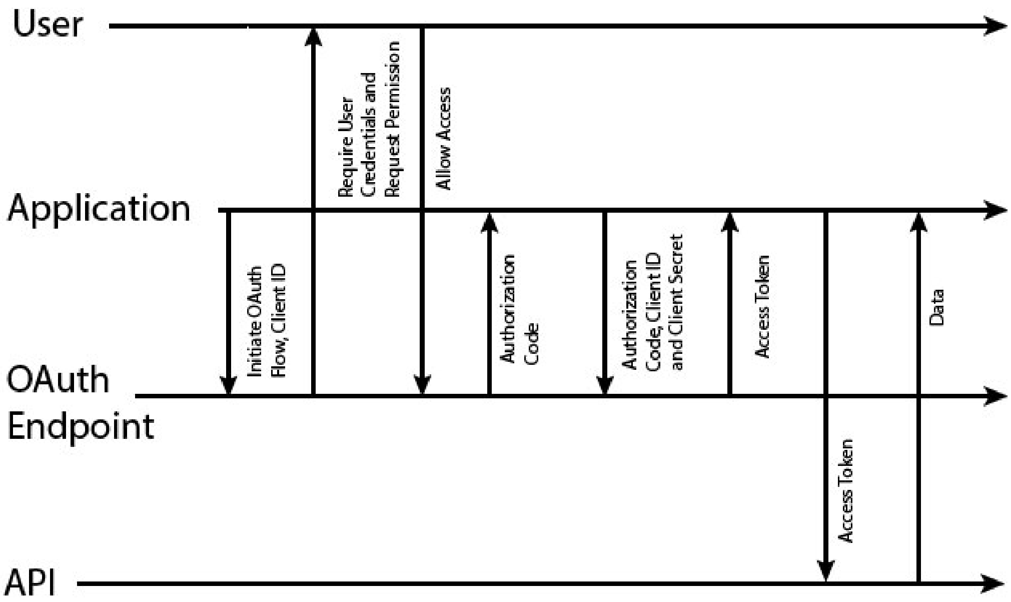
- Confidential
  A confidential scheme is suitable when an application is capable of maintaining the secrecy of the client secret. This is usually the case when an application runs in a browser and accesses its own server when obtaining OAuth access tokens. As such, these schemes make use of the client secret.
  - Application flow
    In the application flow scheme, the user is not required to provide authorization at any stage. Instead, the application uses its client secret to obtain an access token. In this case, it is critical that the client secret is kept safe.



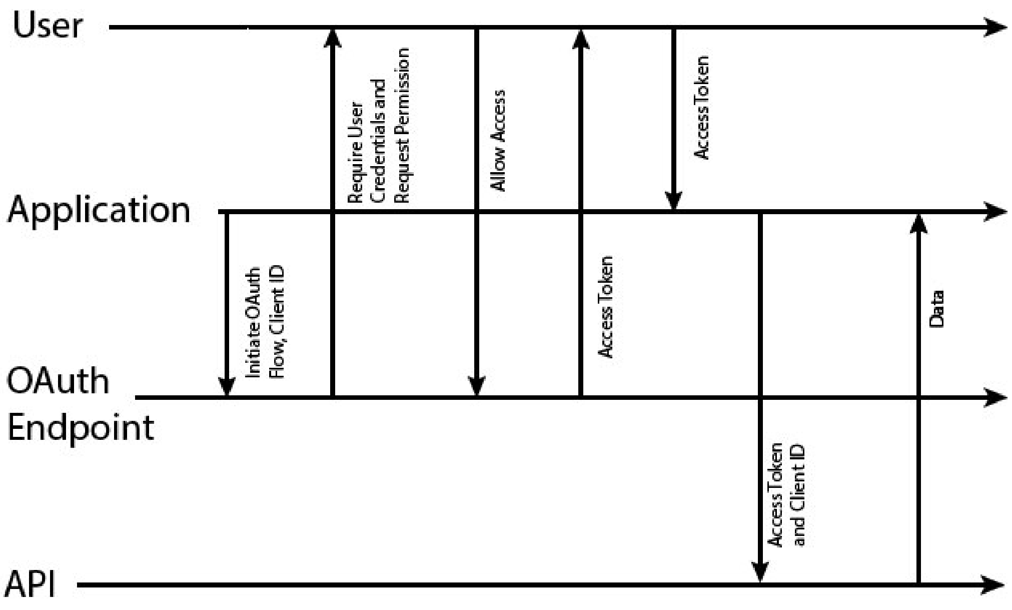
  - Password flow
    In the password flow scheme, the user provides the application with a user name and password that can be used to access the user's data. Following this, the client will directly contact the provider API to request an access token. In this case, trust must exist between user and application because the user's password is revealed to the application. However, this still has an advantage over the application using the password directly, because the validity of the access token or client ID can later be revoked without impacting other applications that do not need their access revoked. However, the application must be trusted to not store the user name and password.



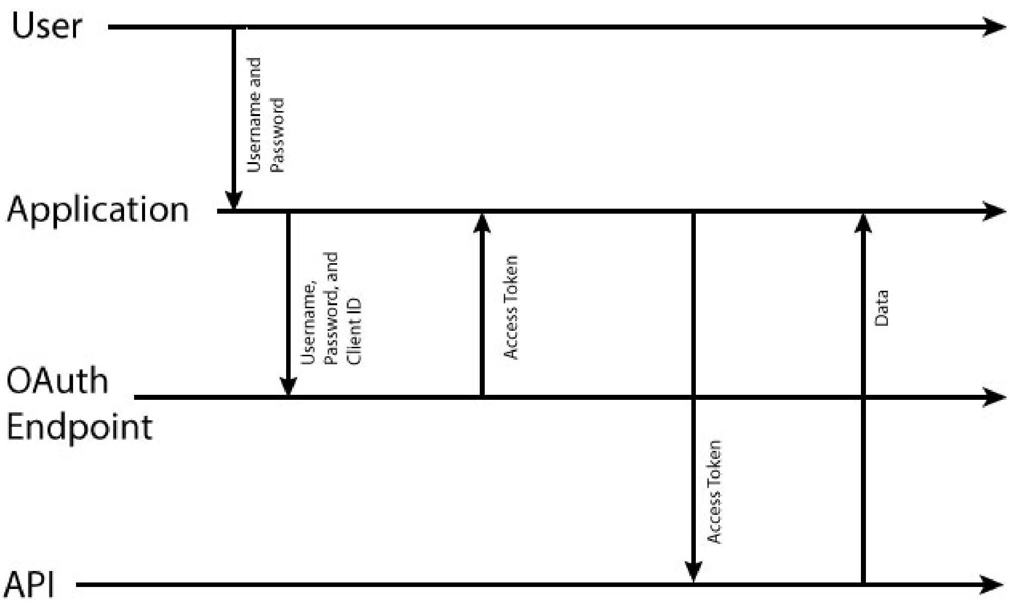
- Access code flow
  In the access code flow, the application has the user provide authorization through a form provided by the gateway server, which, if they grant authorization, provides an authorization code to the application. The application sends the authorization code to the provider API and is granted an access token in return.


User
Require User Credentials and Request Permission
Allow Access
Application
Initiate OAuth Flow, Client ID
Authorization Code
Authorization Code, Client ID and Client Secret
Access Token
Data
OAuth Endpoint
Access Token
API

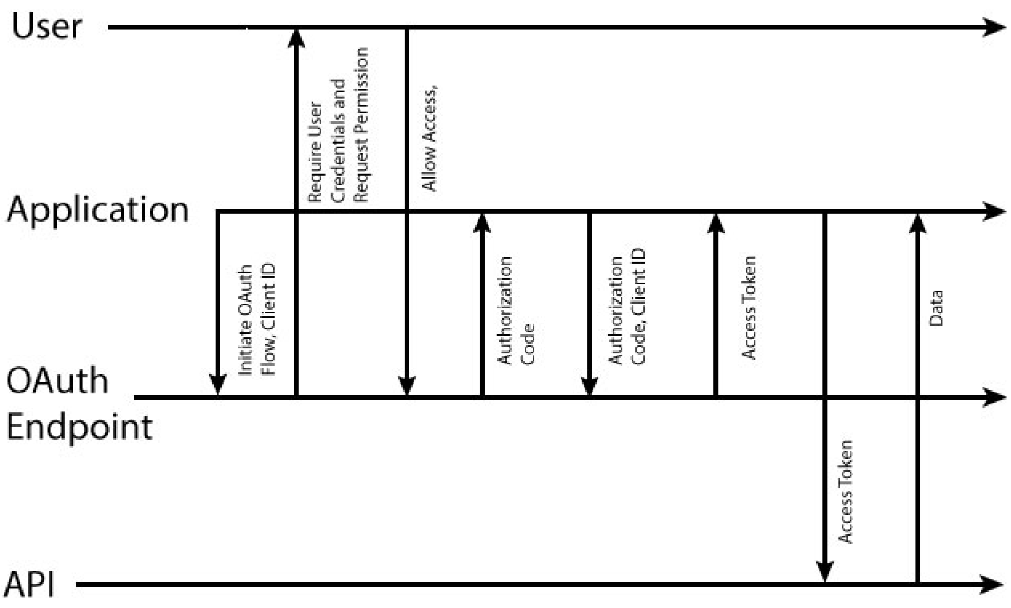
- Public
  A public scheme is suitable when an application is incapable of maintaining the secrecy of the client secret. This is usually the case when the application is native on a computer or mobile where the secret would have to be stored on the user's device, likely inside the source code of the application. As such, these schemes do not make use of the client secret.
  - Implicit flow
    In the implicit flow scheme, the application requests an access token from the gateway server and the user grants permission, at which point an access token is provided to the user, who must then pass the token to the application


User
Application
OAuth Endpoint
API
Initiate OAuth Flow, Client ID
Require User Credentials and Request Permission
Allow Access
Access Token
Access Token
Access Token and Client ID
Data

- Password flow
  In the password flow scheme, the user provides the application with a user name and password that can be used to access the user's data. Following this, the client will directly contact the server to request an access token. In this case, trust must exist between user and application because the user's password is revealed to the application. However, this still has an advantage over the application using the password directly, because the validity of the access token or client ID can later be revoked without impacting other applications that do not need their access revoked. However, the application must be trusted to not store the user name and password.



- Access code flow
  In the access code flow, the application has the user provide authorization through a form provided by the gateway server, which, if they grant authorization, provides an authorization code to the application. The application sends the authorization code to the provider API and is granted an access token in return.


User
Application
OAuth Endpoint
API
Require User Credentials and Request Permission
Allow Access,
Initiate OAuth Flow, Client ID
Authorization Code
Authorization Code, Client ID
Access Token
Data
Access Token

Creating an OAuth 2.0 provider API

To create an OAuth 2.0 provider API, complete the following steps:

1. In a command window, change to the project folder that you created in the tutorial Tutorial: Creating an invoke REST API definition.
2. Change directories to your LoopBack® project and enter the following command:

   ```
   apic edit
   ```

   After a brief pause, the console displays this message:

   ```
   Express server listening on http://127.0.0.1:9000
   ```

   API Designer opens in your web browser, initially displaying the login page if you haven't logged in recently.

   Note: The login page prompts you to Sign in with IBM Cloud. Enter your IBM Cloud credentials, which authenticates you on IBM Cloud and provides access to the API Manager features such as Publish, Explore, and Analytics. You will continue to work in API Designer locally to create APIs, models and data sources.
   Note: If you need to run the editor on a different port, use the following command:
   Linux Mac OS X

   ```
   PORT=port_number apic edit
   ```

   Windows

   ```
   set PORT=port_number && apic edit
   ```

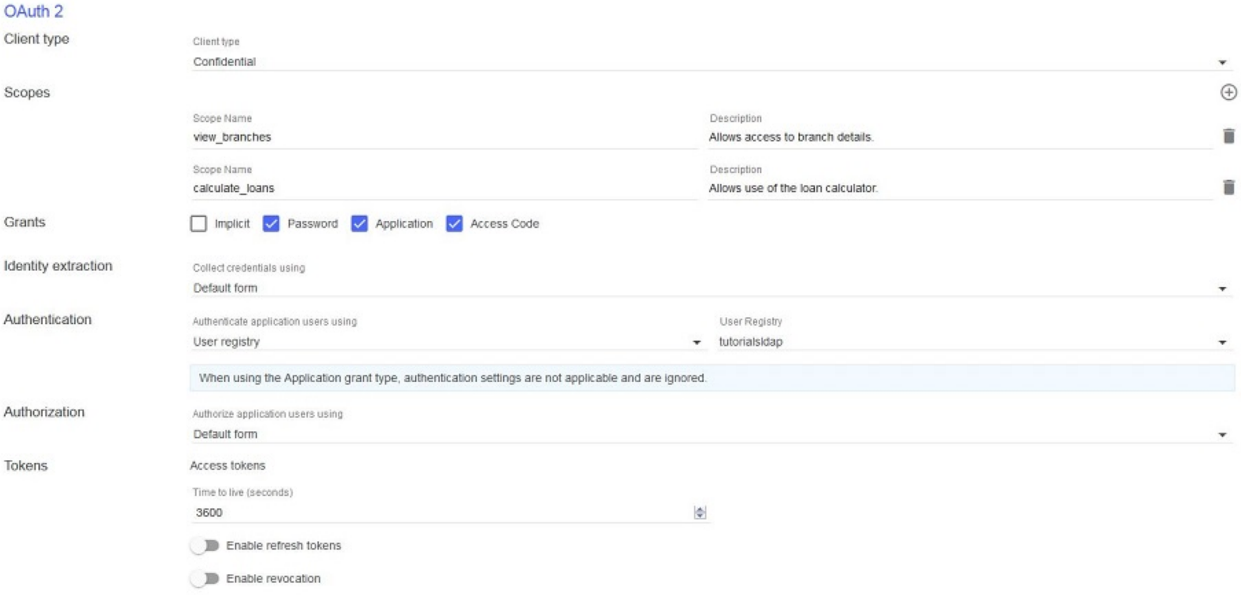
   where *port_number* is the port number to use.
3. In the API Designer, click the APIs tab.
4. Click Add > New OAuth 2.0 Provider API.
5. Complete the fields according to the following table:

   Table 1.

Field	Contents
Title	OAuth Endpoint API
Name	oauth-endpoint-api
Version	1.0.0
Base Path	/oauth-end
V5.0.0 ONLY Description	Provides operations for acquiring access tokens for other APIs.

6. V5.0.3 and earlier Click Next, ensure that Don't add to a product is selected and then click Add.
7. V5.0.4 + Click Create API.
8. In the OAuth 2 section, configure the OAuth settings of your provider API.
   a. Depending on your chosen scheme, select Public or Confidential in the Client type field.
   b. Rename Scope 1 to `view_branches` by using the text field.
   c. In the Description field for view_branches, enter `Allows access to branch details`.

d. Rename Scope 2 to `calculate_loans` by using the text field.
e. In the Description field for calculate_loans, enter `Allows use of the loan calculator`.
f. Delete Scope 3 by clicking the Remove scope icon .
g. In the Grants section, clear the check box of any grant types you do not want to use.
Note: You must clear the check box of Implicit if you selected Confidential in step 8.a and you must clear the check box of Application if you selected Public in step 8.a.
h. In the Collect credentials using field of Identity extraction, select Basic.
i. In the Authenticate application users using field of Authentication, select User registry and, in the User registry field, enter `tutorialsldap` to reference the LDAP registry that you created in Tutorial: Securing APIs by using an LDAP user registry.
j. In the Authorize application users using field of Authorization, select Default form.
k. Ensure that the Enable refresh tokens and Enable revocation switches of Tokens are in the Off position.



OAuth 2

Client type
Client type: Confidential

Scopes
Scope Name: view_branches
Description: Allows access to branch details.
Scope Name: calculate_loans
Description: Allows use of the loan calculator.

Grants
□ Implicit ☑ Password ☑ Application ☑ Access Code

Identity extraction
Collect credentials using: Default form

Authentication
Authenticate application users using: User registry
User Registry: tutorialsldap
When using the Application grant type, authentication settings are not applicable and are ignored.

Authorization
Authorize application users using: Default form

Tokens
Access tokens
Time to live (seconds): 3600
Enable refresh tokens
Enable revocation

9. Click the Save icon to save your changes.

Configuring the API security scheme

To enable your chosen authentication scheme in API Designer, complete the following steps:

1. In the API Designer, click the APIs tab.
2. Click your Branches API definition.
3. In the Security Definitions section, click the Add Security Definition icon and then click OAuth.
4. In the Name field, rename your security definition as `OAuth definition`.
5. In the Flow field, select the type of flow you want to use.
6. In the Scopes section click the Add scope icon .
7. In the Scope Name field, rename the scope to `view_branches`.
8. If you are using a flow that uses an authorization URL, in the Authorization URL field, enter the following URL:

```
https://Host_Address/Org_Segment/sb/oauth-end/oauth2/authorize
```

where:
- *Host_Address* is the address of your gateway host.
- *Org_Segment* is the path segment of the organization that you want to use.

Note: If you do not know your host address or organization segment, you can find them by completing the following steps:
a. Log in to the API Manager user interface.
b. From the Dashboard page, click the Sandbox Catalog to display its details.
c. Click the Settings tab, then click Endpoints. You can obtain the host address and organization segment from the displayed Base URL value.
9. If you are using a flow that uses an authentication URL, in the Authentication URL field, enter the following URL:

```
https://Host_Address/Org_Segment/sb/oauth-end/oauth2/token
```

where:
- *Host_Address* is the address of your gateway host.

- *Org_Segment* is the path segment of the organization that you want to use.

   Note: If you do not know your host address or organization segment, you can find them by completing the following steps:
   a. Log in to the API Manager user interface.
   b. From the Dashboard page, click the Sandbox Catalog to display its details.
   c. Click the Settings tab, then click Endpoints. You can obtain the host address and organization segment from the displayed Base URL value.

10. In the Security section, select the check box for your OAuth definition and clear the check box for your LDAP (Basic) security definition.
    Note: You should have only your OAuth definition and clientID security definitions select.
11. Click the Save icon 💾 to save your changes.
12. Click All APIs and then click the Products tab.
13. Click your Banking Services Product.
14. In the APIs section, click the Add API icon ⊕.
15. Then select Branches and OAuth Endpoint API and then click Apply.
16. Click your Basic Plan to expand it and ensure that Branches and OAuth Endpoint API are selected.
17. Click the Save icon 💾 to save your changes.
18. Publish the Banking Services Product.
    a. Click Publish and then click the Sandbox Catalog for the host address and organization that you want to use.
    b. Select Select specific products and then select your Banking Services Product.
    c. Click Publish.

Acquiring access tokens for individual schemes

In this section you will acquire an access token to access your API by using OAuth 2.0. The different schemes each use a different method to acquire an access token.

1. In your Developer Portal, sign in to your developer account.
   Note: An administrator account cannot register applications and therefore cannot be used in this tutorial.
2. Click Apps
3. V5.0.5 + Click Create new App.
   V5.0.4 and earlier Click Register new Application.
4. In the Title field, enter `OAuth Application`.
5. In the OAuth Redirect URI field, enter `https://example.com/redirect` and then click Submit.
6. Click Show Client Secret and click Show beside the Client ID field and then record your application's client secret and your application's client ID.
   Note: The Client Secret can be viewed only once; if you have viewed it before and not recorded it you will need to reset it, which will invalidate uses of the previous secret elsewhere.
7. Click API Products and then click your Banking Services Product.
8. V5.0.3 and earlier In the navigation bar, click Plans.
9. For your Basic plan, click Subscribe.
10. Select your OAuth Application and then click Subscribe.
11. Acquire the access token. The process for obtaining an access token differs for each scheme. Click the link for your chosen scheme:
    - Confidential
      - Application flow
      - Password flow
      - Access code
    - Public
      - Implicit flow
      - Password flow
      - Access code

Using the access token

This section is common to all the OAuth schemes. You will access the Branches API and be authenticated.

1. From the API page of your Branches API in the Developer Portal, expand the details of your GET /branches/details operation.
2. Record the operation URL that is displayed.
3. Enter the following command into your command line interface (as one line):

```
curl -k -v -H "X-IBM-Client-Id: Client_ID" -H "Authorization: Bearer Access_Token" -X GET
'Operation_URL'
```

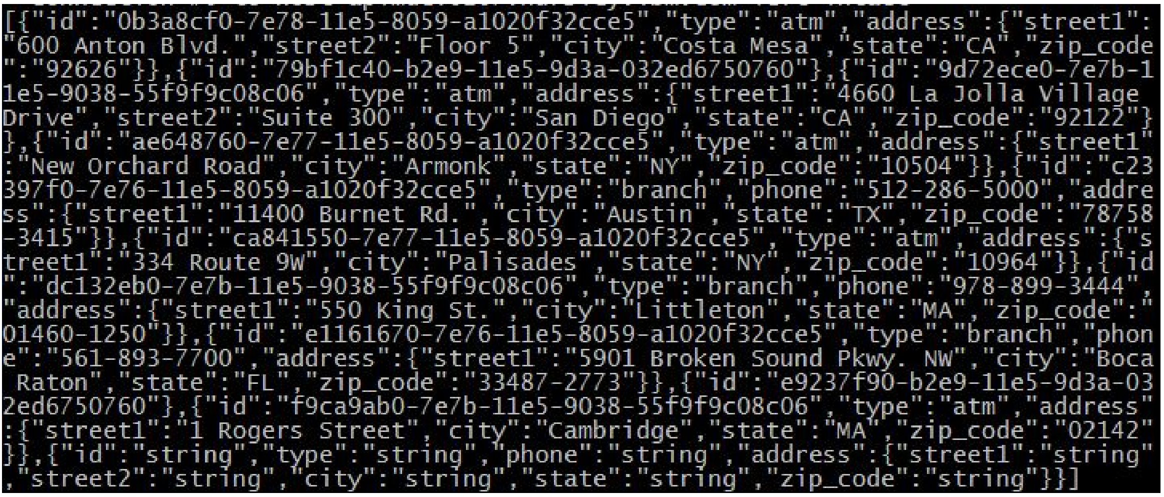
   where:
   - *Client_ID* is as recorded in step 6 of the Acquiring access tokens for individual schemes section of this tutorial.
   - *Access_Token* is the access token that you recorded at the end of the scheme-specific section of this tutorial.

- *Operation_URL* is the URL you would use to call the operation if it were to be unsecured

For example:

```
curl -k -v -H "X-IBM-Client-Id: 6e3f115d-5220-48bd-a019-3c9680e657b7"
 -H "Authorization: Bearer AAEkNmUzZjExNWQtNTIyMC00OGJkLWEwMTktM2M5Nj
gwZTY1N2I3_3QEMOL24OpXJTnuA4y0qvqM_7HgX7ImJ0Uyl_Jcq8Xx8CRHTBtPoBukueb
fgDK3BtZjId7_Yeyjd01vqx8Xl0ORdFxC0BF-grZLCG1BeaY"
-X GET 'https://host.com/myorg/sb/branches/details'
```

The response includes the data from the operation.



```
[{"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":"79bf1c40-b2e9-11e5-9d3a-032ed6750760"},{"id":"9d72ece0-7e7b-1
1e5-9038-55f9f9c08c06","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":"c23
397f0-7e76-11e5-8059-a1020f32cce5","type":"branch","phone":"512-286-5000","addre
ss":{"street1":"11400 Burnet Rd.","city":"Austin","state":"TX","zip_code":"78758
-3415"}},{"id":"ca841550-7e77-11e5-8059-a1020f32cce5","type":"atm","address":{"s
treet1":"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","phon
e":"561-893-7700","address":{"street1":"5901 Broken Sound Pkwy. NW","city":"Boca
 Raton","state":"FL","zip_code":"33487-2773"}},{"id":"e9237f90-b2e9-11e5-9d3a-03
2ed6750760"},{"id":"f9ca9ab0-7e7b-11e5-9038-55f9f9c08c06","type":"atm","address"
:{"street1":"1 Rogers Street","city":"Cambridge","state":"MA","zip_code":"02142"
}},{"id":"string","type":"string","phone":"string","address":{"street1":"string"
,"street2":"string","city":"string","state":"string","zip_code":"string"}}]
```

.

4. To verify that access is granted only for the correct user ID and the correct access token, alter one or both in the previous command.

What you did in this tutorial

In this tutorial you have:

- Chosen your OAuth Scheme.
- Configured the API Security Scheme.
- Acquired an access token for your chosen scheme.
- Used the access token.

What to do next

- When acquiring an access token, use the `calculate_loans` scope, instead of the `view_branches` scope, to verify that a code requested for only the scope specified by the secured API can be used to access the API.
- Repeat the tutorial for a different OAuth flow.

Related concepts

- OAuth user scenario
- Tokens

Related tasks

- Configuring API security by using the API Designer
- Protecting an API with OAuth security definition

Related reference

- OAuth revocation URL

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for application flow

This tutorial shows you how to acquire an access token for the OAuth scheme *application flow* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the application flow scheme, which is only valid when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. Expand the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

```
curl -v -u Client_ID:Client_Secret -k -X POST -d {}
'Token_Endpoint_URL?grant_type=client_credentials&scope=Scope'
```

Note: On Windows systems, you must modify the **curl** command to place quotation marks around the curly brace characters to change them from `{}` to `"{}"`.
Where:

- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
- *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
- *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
- *Token_Endpoint_URL* is as recorded in step 2.

Note: If you are adapting the command for use with your command line interface, you must send `application/x-www-form-urlencoded` content.
For example:

```
curl -v -u 097a4830-eeb7-4f6e-ad4b-e507313a771e:1Q8hT1vA1mJ4qL0eL7nW0xL7wL5gE4hF8aL
6wH5fW0rB8rW3iD -k -X POST -d {} 'https://host.com/myorg/sb/oauth-end/oauth2/token?
grant_type=client_credentials&scope=view_branches'
```

A message is returned, of the following form:

```
{"token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"/Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFYy1hbGwiOugk49sRso0D1yKfD
i4Uny_W589WWa_Ea3eBG6ZbWwh-BT2zawnHK8sKM1EBUuAGTcyYh-n7sATyWi3-ElH8QBWqeadLE0h5c
JcsUxMHa_65VM_tI8KnnNphi7CIxx0NJRuMbCE8uOHRIPCmNon3", "expires_in":3600, "scope":"/branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for password flow (confidential)

This tutorial shows you how to acquire an access token for the OAuth scheme *password flow (confidential)* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the password flow scheme when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. In the Developer Portal, select the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

```
curl -v -u Client_ID:Client_Secret -k -X POST -d
'grant_type=password&scope=Scope&username=User_Name&
password=Password' 'Token_Endpoint_URL'
```

Note: On Windows systems, you must modify the **curl** command to place quotation marks around the curly brace characters to change them from `{}` to `"{}"`.
where:

- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
- *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
- *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
- *Token_Endpoint_URL* is as recorded in step 2.
- *User_Name* is a valid user name from your LDAP registry.
- *Password* is the password that corresponds to *User_name* in your LDAP registry.

Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
For example:

```
curl -v -u 51dfaed9-9d07-46fc-83a6-52c09735a4a0:W3xB2tK0uP7vC8uX0rO8t
P5gE0yD7aV8eE3wN0jG1vY7aT1tL2 -k -X POST -d 'grant_type=password&scope=view_branches&
username=USER@example.com&password=PASSWORD'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

A message is returned, of the following form:

```
{ "token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"/Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFYy1hbGxug39ky5-MaadRBFgRB4Cn
hzClFNbidoP_UGs56vyTy6r-TbdqOLXuIaLQz8SKLdteXqN3VutqOnSkiB50fm9h6nvodXYrHkiLwGE9Sas
IzJImtTFuNNEdE03WL1BbLKnhNM_I0DxBoETUT0yIfNJToJJwnV4_yTaPuMwuL6ZjbQ", "expires_in":3600,
"scope":"/branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for access code flow (confidential)

This tutorial shows you how to acquire an access token for the OAuth scheme *access code flow* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the access code flow scheme when the client is suitable for confidential approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. Expand the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL and Authorization Endpoint URL values.
3. In your web browser, enter the following URL:

   ```
   Authorization_Endpoint_URL?response_type=code&
   redirect_uri=Redirect_URI&scope=Scope&
   client_id=Client_ID
   ```

   where:
   - *Authorization_Endpoint_URL* is as recorded in step 2.
   - *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
   - *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

   For example:

   ```
   https://host.com/myorg/sb/oauth-end/oauth2/authorize?
   response_type=code&redirect_uri=https://example.com/redirect&scope=view_branches&
   client_id=51dfaed9-9d07-46fc-83a6-52c09735a4a0
   ```

4. When prompted, authenticate using the LDAP registry you specified when creating your API.
   You are presented with a prompt asking you to confirm that you want to grant your OAuth Application access to your Branches API.

5. Click Allow Access.
   You are redirected to your Redirect URI, which will have additional information appended. This URL takes the following form:

   ```
   Redirect_URI?code=Authorization_Code
   ```

   For example:

   ```
   https://example.com/redirect?code=AAINXmK3Qp91jot1nqrqPDgrhpDJNOD
   1oywxOHeJ2sdYfOLLcwZZOa82tvUDpknWtfdQe1s59DYGOnR9XQn6jJg7D4HPf2HGsofwkvZPJH
   brMubPuJtGgjWYQa-Ra_bScJsc5LF9KznPEBXFG2QjqJNqlHrtziyZAqJfWFBuIqYwQg
   ```

   Record the *Authorization_Code* value.
6. From your command line interface, enter the following command (as one line):

   ```
   curl -v -k -u Client_ID:Client_Secret -X POST -d
   'grant_type=authorization_code&redirect_uri=Redirect_URL&code=Authorization_Code'
   'Token_Endpoint_URL'
   ```

   where:
   - *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
   - *Client_Secret* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

- *Redirect_URL* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
- *Authorization_Code* is as recorded in step 5.
- *Token_Endpoint_URL* is as recorded in step 2.

Note: If you are adapting the command for use with your command line interface, you must send `application/x-www-form-urlencoded` content.
For example:

```
curl -v -k -u 094d3694-5c35-43b4-bea6-13a9ab0fdb89:nR7yC0dS4fH0tL8dQ2aV5m
S2aF7lC5uT7vJ6vP6fI6gS6dH5pE -X POST -d 'grant_type=authorization_code&
redirect_uri=https://example.com/redirect&code=AAIdK4UzxGPkKVYGcfViaTui8I1OpJs4nrv
U4xuuD4uNlMuR-lR8XBM6zfk6aPBU5NeTHQUlE2Z-OFKYmXasCexqocPMAV-GLxf539m6x0-FT3LBTh14s_
DBgsyNpSxAC7q3HhkxKbquGhGH2paD4ogUIAvwJDpXZaV-ekfqVLAZzA'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

The response takes the following form:

```
{ "token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"/Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFYy1hbGy8uvozvrfdHMbA1kw5BTt1yuaoO2B
QYaInzRX7cSI5HB9OqsM2LxVs7mJEfDGRHgSqlWyXZ1zMFusHqRwBgur_fEi0k7xYzoTG1IKWaL990HHpZwuGYclf
wPaOcq29AWaUY21x1E7XYVaG-k5hJrEwiBa62IeMk_inl6nDnqfgAA", "expires_in":3600, "scope":"/branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for implicit authorization

This tutorial shows you how to acquire an access token for the OAuth scheme *implicit authorization flow* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the implicit authorization flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. Expand the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Authorization Endpoint URL value.
3. Open a new browser tab or window and enter the following URL (as one line):

```
Authorization_Endpoint_URL?response_type=token&
redirect_uri=Redirect_URL&scope=Scope&
client_id=Client_ID
```

where:

- *Authorization_Endpoint_URL* is as recorded in step 2.
- *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
- *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

For example:

```
https://host.com/myorg/sb/oauth-end/oauth2/authorize?
response_type=token&redirect_uri=https://example.com/redirect&scope=view_branches&
client_id=95de3c66-5c52-486b-9229-7985b16bcd8a
```

4. Authenticate with your LDAP registry when prompted to do so. You are presented with a page requesting permission for your OAuth Application to access your Branches API.
5. Click Allow Access. You are redirected to your redirection URL with additional information appended to the URL. This has the following form:

```
Redirect_URL#access_token=Access_Token&expires_in=3600&
scope=Scope&token_type=bearer
```

For example:

```
https://example.com/redirect#access_token=AAEFcC1hbGx0zz3KhAki_
vD-PSaAIMx6tbsxBJ858oYRWRUwP6aO7BQPySCURVj0e2FJ49pIvooAZYUZfNCz2cfqHMhiN
9IR6ID71c563oBiCV_f_o7lvTC68Wad2CTTn5Ys7hrq0Z-hBbxBe-m3hI7bZBf44reE821Oq
rSX2EqwvXVaalkqag&expires_in=3600&scope=view_branches&token_type=bearer
```

Record the *Access_Token* value that is displayed in the query parameters.

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for password flow (public)

This tutorial shows you how to acquire an access token for the OAuth scheme *password flow (public)* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the password flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. Expand the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL value.
3. Using your command line interface, enter the following command (as one line):

```
curl -v -k -X POST -d 'grant_type=password&scope=Scope&
username=User_Name&password=Password&client_id=Client_ID'
'Token_Endpoint_URL'
```

where:

- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.
- *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
- *Token_Endpoint_URL* is as recorded in step 2.
- *User_Name* is a valid user name from your LDAP registry.
- *Password* is the password that corresponds to *User_name* in your LDAP registry.

Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
For example:

```
curl -v -k -X POST -d 'grant_type=password&scope=view_branches&
username=USER@example.com&password=PASSWORD&client_id=552b44f4-d533-4593-bec7-24dacf129847'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

A message is returned, of the following form:

```
{ "token_type":"bearer", "access_token":"Access_Token",
"expires_in":3600, "scope":"/Scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFcC1hbGxDUs-Pg6-O_
6N85VSN5uCNrk83vDCqK6ckswsO86DsKWY-zfr0ANPD9CtcY4PgSq5bC_mPnSCUyhSlEj1an
qOj1YXEWJEW-iN2xNlm4Uwm1l2x_yJrEEud_uBcX5m4L1Jev7iM082Ozb6j4aJCGsN5CL69-
7JqZSLMxCgb-Jwbkg", "expires_in":3600, "scope":"/branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Acquiring an access token for access code (public)

This tutorial shows you how to acquire an access token for the OAuth public scheme *access code flow* in IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
This tutorial is a subsection of Tutorial: Securing an API by using OAuth 2.0 and cannot be completed independently. All prerequisites from the parent tutorial apply.

About this tutorial

You will acquire an access token for your application that will allow it to access an API. This tutorial is for the Password Flow scheme when the client is suitable for public approaches. You will use the application and API from the Tutorial: Securing an API by using OAuth 2.0 tutorial.

Requesting an access token

1. Expand the GET /details operation of your Branches API.
2. In the Authorization section of Try this operation, in the console, record your Token URL and Authorization Endpoint URL values.
3. Open a new browser tab or window and enter the following URL (as one line):

```
Authorization_Endpoint_URL?response_type=code&
redirect_uri=Redirect_URL&scope=Scope&client_id=Client_ID
```

where:

- *Authorization_Endpoint_URL* is as recorded in step 2.
- *Scope* is `view_branches`, as specified for the secured API in step 7 of the Securing an API by using OAuth 2.0 tutorial.
- *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

For example:

```
https://host.com/myorg/sb/oauth-end/oauth2/authorize?
response_type=code&redirect_uri=https://example.com/redirect&scope=view_branches&client_id=
801a94e9-556f-4034-b333-f0f2c680f5f2
```

4. When prompted, authenticate using the LDAP registry you specified when creating your API. You are presented with a prompt asking you to confirm that you want to grant your OAuth Application access to your Branches API.
5. Click Allow Access. You are redirected to your Redirect URI which will have additional information appended. It will take the following form:

```
Redirect_URI?code=Authorization_Code
```

Record the *Authorization_Code*.
For example:

```
https://example.com/redirect?code=AAKjnhCDCUsQszr5osEEOo2ZDH7WN80znOq0nwEVzzUEub
HWu7etbQzXEOIB5EjFfYQcQHegAv5BYNufBK2_HG-cXvi4eW_3a8wNga4NbDsuGGnQNBdNcWkR1U7XKUXrhOVZ10hN
klII4WjCuWpjDtsYD3U6ihdxcITUY1IhMjFxqQ
```

6. From your command line interface, enter the following command (as one line):

```
curl -v -k -X POST -d 'grant_type=authorization_code&code=
Authorization_Code&redirect_uri=Redirect_URI&
client_id=Client_ID' 'Token_Endpoint_URL'
```

where:

- *Authorization_Code* is as you have previously recorded.
- *Token_Endpoint_URL* is as recorded in step 2
- *Redirect_URI* is as set in step 5 of the Securing an API by using OAuth 2.0 tutorial.
- *Client_ID* is as recorded in step 6 of the Securing an API by using OAuth 2.0 tutorial.

Note: If you are adapting the command for use with your command line interface, you must send **`application/x-www-form-urlencoded`** content.
For example:

```
curl -v -k -X POST -d 'grant_type=authorization_code&redirect_uri=
https://example.com/redirect&code=AAI3hfmo9HroOLzONkJuCUkZVUlUzSZXC4LK6xGC2Ta
5bODaDPBBZBeRnV2wNWSuibINAl7mJ65RWZJ_IQ6wqG26KW8ipdlj2KUDnjq1E_hjIE5GxvWk0avj
FbBPqm9sc-IKi27l36Ps01pPAQpTweB6LNxyvlqyOcHH4pwTjqFwxA&
client_id=9a131485-75f4-4e70-8657-a844b4a279af'
'https://host.com/myorg/sb/oauth-end/oauth2/token'
```

The response takes the following form:

```
{"token_type":"bearer", "access_token":"Access Token",
"expires_in":3600, "scope":"/scope}
```

Record the *Access_Token* value that is displayed in the message.
For example:

```
{ "token_type":"bearer", "access_token":"AAEFcC1hbGyoPkKTr_0_HQ3e7Q_u1Gx
_aejgznGs7im6H_jNR3xVSZrTcTCuuyvvj9u0K1xqpLXDQOjfCFhGfWOMql-LQ9F9-NjStVeIIpWpJ5Wjx
yCW1oU459k_Uvmm03SdHV4eS1PSiG_YroxqCgWDkC1b2UKFM5_72HYglbCsvBEstSuqJQ",
"expires_in":3600, "scope":"/branches" }
```

Using the access token

This section is common to all OAuth schemes and so is addressed in the Using the access token section of the Tutorial: Securing an API by using OAuth 2.0 tutorial.

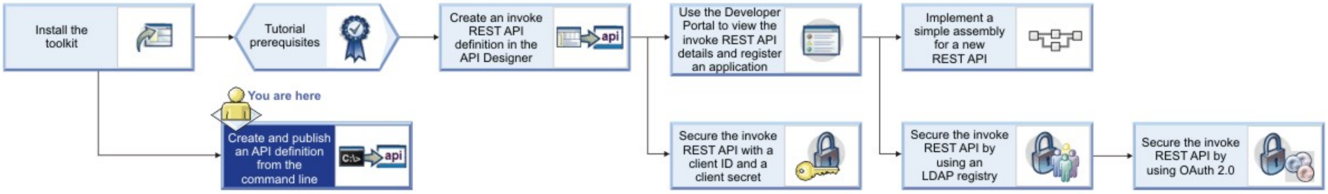
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Creating and publishing an API definition from the command line

This tutorial shows you how to use the developer toolkit to create an API definition, include it in a Product, and publish the Product to a Catalog in API Manager by using IBM® API Connect Version 5.0.6 and earlier.

Before you begin

Note: For tutorials about working with the developer toolkit in IBM API Connect Version 5.0.7 and later, see Developer toolkit tutorials for V5.0.7 and later.
The following diagram shows the sequential flow through the IBM API Connect Developer toolkit tutorials for working with API definitions that call an existing endpoint. Before beginning a tutorial, ensure that you have completed the previous tutorials in the sequence. You can click a tutorial in the diagram to open the instructions for that tutorial.


Install the toolkit
Tutorial prerequisites
Create an invoke REST API definition in the API Designer
api
Use the Developer Portal to view the invoke REST API details and register an application
Implement a simple assembly for a new REST API
You are here
Create and publish an API definition from the command line
C:\>
api
Secure the invoke REST API with a client ID and a client secret
Secure the invoke REST API by using an LDAP registry
Secure the invoke REST API by using OAuth 2.0

About this tutorial

With the developer toolkit, you can create API definition files locally and develop them by using your chosen editor. When you are ready to make them available to application developers, you publish them to a Catalog in API Manager, where they can be managed through their lifecycle. In this tutorial, you will create and publish an API.

To make an API available, it must be included in a Plan, and the Plan must be included in a Product; you then stage the Product to API Manager. In this tutorial, you first create an API, then you create a Product that includes the API. A default Plan, that contains the API, is created automatically in the Product.

Instructions are provided to create and publish an API that represents a banking service for obtaining loan quotations; however, if you want to create and publish your own API, you can adapt the instructions accordingly.

In this tutorial, you will complete the following activities:

- Create a local API definition YAML file.
- Validate the API definition.
- Create a local Product definition.
- Validate the Product definition.
- Log in to API Manager.
- Publish the Product to API Manager.

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`**.

Create a local API definition YAML file

To create a local API definition YAML file, you use the **`apim create`** command.

Complete the following steps:

1. Create a folder to hold your API and Product definitions, and change to that folder in a command window.
2. Enter the following command:

   ```
   apic create --type api --title Loans
   ```

   This command creates a file called loans.yaml, and assumes the following defaults:

version	1.0.0
basepath	/loans

   However, you can supply further command options to override these values.
3. Examine the contents of the loans.yaml file by opening it in your chosen editor.

Although not part of this tutorial, the API can be further developed by modifying the OpenAPI (Swagger 2.0) definition in the YAML file.

Validate the API definition

To validate the syntactical correctness of the API definition YAML file, you use the **apim validate** command.

Enter the following command:

```
apic validate loans.yaml
```

The validation should complete successfully.

Create a local Product definition

To create a local Product definition YAML file, you use the **apim create** command.

Complete the following steps:

1. Enter the following command:

   ```
   apic create --type product --title "Banking Services" --apis loans.yaml
   ```

   This command creates a product definition file called banking-services.yaml that includes a reference to the loans.yaml API definition file that you defined previously. The following defaults are assumed:

name	banking-services
version	1.0.0

   However, you can supply further command options to override these values. The value of the name property determines the file name.
2. Examine the contents of the banking-services.yaml file by opening it in your chosen editor. Note that a reference to the loans API is included.
   Although not part of this tutorial, you can modify the Product definition in the YAML file; for example, you can add further Plans and APIs.

Validate the Product definition

To validate the syntactical correctness of the Product definition YAML file, you use the **apim validate** command.

Enter the following command:

```
apic validate banking-services.yaml --product-only
```

The validation should complete successfully.
By default, the **apic validate** command validates the Product definition YAML file and all the API definition files that it references. If you supply the **--product-only** option, only the Product definition is validated.

Log in to API Manager

Complete the following steps:

1. Enter the following command:

   ```
   apic login
   ```

2. At the prompts, enter the following information:
   - Server: The virtual host name or virtual IP address of your Management cluster.
   - Username: The user name with which you are registered in API Manager.
   - Password: Your API Manager password.

   Note: You can also supply your credentials as command parameters. However, if you use the command interactively, your password is hidden.

Stage the Product to API Manager

For application developers to be able to use an API, it must be published to a Catalog. A Catalog has an associated Developer Portal and runtime capability. For example, a simple provider organization might consist of a development Catalog and a production Catalog.

Complete the following steps:

1. To list the Catalogs that are available in API Manager, enter the following command:

   ```
   apic catalogs --all-organizations --server management_cluster_hostname_or_address
   ```

   You should see the identifier for a Catalog called `sb`, which corresponds to the development Catalog that is created by default when IBM API Connect is installed; the identifier includes the organization that contains it. For example:

   ```
   apic-catalog://myserver.mydomain.com/orgs/myorg/catalogs/sb
   ```

2. Publish the Product to the development Catalog by entering the following command (all on one line):

   ```
   apic publish banking-services.yaml --catalog sb --organization
   URL_path_segment_of_your_provider_organization
   --server management_cluster_hostname_or_address
   ```

   Note: You can choose only to stage the Product, by supplying the `--stage` option. If a Product is staged, its APIs are not yet available to application developers, and the Product must be subsequently published by using the API Manager user interface. For more information, see Managing your Products.
3. Confirm that your Product and its referenced APIs have been published to the development Catalog by entering the following commands:

   ```
   apic products --catalog sb --organization URL_path_segment_of_your_provider_organization --server
   management_cluster_hostname_or_address
   apic apis --catalog sb --organization URL_path_segment_of_your_provider_organization --server
   management_cluster_hostname_or_address
   ```

   You should see the `banking-services` Product and the `loans` API listed.

Note: You can define default values for your management server, Catalog and provider organization so that you do not have to supply them as command options; enter the following command (all on one line):

```
apic config:set catalog=apic-catalog://management_cluster_hostname_or_address
/orgs/URL_path_segment_of_your_provider_organization
/catalogs/URL_path_segment_of_your_catalog
```

For example:

```
apic config:set catalog=apic-catalog://myhost.com/orgs/myorg/catalogs/sb
```

You can obtain the required value of the `catalog` configuration property for a specific Catalog from the API Manager user interface; for more information, see Obtaining the publish target URL for a Catalog.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Created a local API definition YAML file.
- Validated the API definition.
- Created a local Product definition.
- Validated the Product definition.
- Logged in to API Manager.
- Defined your Management cluster and your organization.
- Staged the Product to API Manager.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reference

Reference information for the API Designer component in API Connect.

- **API Connect context variables**
  List of 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.
- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Connect context variables

List of 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:

- API Connect context variables.
- OAuth context variables.
- Application certificate 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 IBM API Connect see Variable references in API Connect.

For more information about creating a user-defined policy, see Authoring policies.

Context variables

Note:

- Although some context variables can be used with both the DataPower® Gateway and the Micro Gateway, some context variables are restricted to a particular Gateway or return results in a different format depending on the Gateway. Restrictions or differences are marked with the following icons:
  - DataPower Gateway only Indicates that the context variable is available only on the DataPower Gateway.
  - Micro Gateway only Indicates that the context variable is available only on the Micro Gateway.
- 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.
- With the exception of client ID and client secret, the passing of form input as a parameter into an API is not supported.
- IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

Table 1. API Connect context variables

Name	Description	Permissions
`api.name`	The name of the API; this corresponds to the `x-ibm-name` field in the OpenAPI (Swagger 2.0) definition for the API.	Read/write
`api.document`	The OpenAPI (Swagger 2.0) document.	Read/write
DataPower Gateway only `api.root`	DataPower Gateway only The API basePath. Micro Gateway only Use `api.document.basepath` for the API basePath.	Read/write
`api.version`	The version string of the API.	Read/write
`api.endpoint.address`	The address of the API Gateway endpoint.	Read/write

Name	Description	Permissions
`api.endpoint.hostname`	The host name of the API Gateway endpoint, as requested by the application.	Read/write
V5.0.1+ `api.operation.id`	V5.0.1+The ID of the operation.	V5.0.1+ Read/write
V5.0.1+ `api.operation.path`	V5.0.1+The path of the operation.	V5.0.1+ Read/write
`api.org.id`	The organization ID of the API provider.	Read/write
`api.org.name`	The organization short name of the API provider.	Read/write
`api.type`	The API type; REST or SOAP.	Read/write
`api.properties.`*`propertyname`*	The name of a custom API property. Property values are Catalog specific. Note: - You have write permission to a custom property **only** from the user interface, **not** from GatewayScript. - To access a Catalog specific property value from GatewayScript, you must refer to the property by using the following syntax: `apim-`*`catalog`*`-`*`name`* where *catalog* is the name of the Catalog, and *name* is the property name. For example: `var mypropertyvalue = $(apim-mycatalog-mypropertyname)`	Read/write
`client.app.name`	The name of the application that is identified as having issued the request.	Read/write
`client.app.id`	The client ID or application key that is received on the request.	Read/write
`client.app.secret`	The client secret that is received in the request.	Read/write
V5.0.7+ `client.app.type`	V5.0.7+The status of the calling client application. The possible values are as follows: - Development - Production (default) For more information, see Managing the application lifecycle.	V5.0.7+ Read/write
`client.org.id`	The unique identification key of the organization that owns this application.	Read/write
`client.org.name`	The name of the organization that owns this application.	Read/write
`env.path`	The path segment that represents this Catalog.	Read/write
`message.body`	The payload of the request or response message. DataPower Gateway only Note: The `message.body` context variable is not supported with `getContext()` function. Use the `getvariable()` function instead.	Read/write
`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 *name* segment is case-insensitive.	Read/write
`message.status.code`	The HTTP status code of the response.	Read/write
`message.status.reason`	The HTTP reason phrase of the response.	Read/write
`plan.name`	The name of the plan.	Read/write
`plan.id`	The unique identifier of the plan.	Read/write
`plan.version`	The version number of the plan.	Read/write
`plan.rate-limit`	The rate limit (the number of API calls per time interval) of the plan.	Read/write
`request.authorization`	The parsed HTTP `authorization` header.	Read-only
`request.body`	The payload from the incoming request.	Read-only
`request.content-type`	Normalized content-type value.	Read-only
`request.date`	A date object that represents approximately when the request was received by the Gateway.	Read-only
`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 *headername* segment is case-insensitive.	Read-only
`request.parameters`	You can obtain your incoming parameters from path and query parameters.	Read-only
`request.path`	The path section of the `request.uri` that starts with the base path of the API, including the '/' character that begins the base path.	Read-only

Name	Description	Permissions
`request.querystring`	The request query string without the leading question mark.	Read-only
`request.search`	The request query string with the leading question mark.	Read-only
`request.uri`	The full HTTP request URI from the application.	Read-only
`request.verb`	The HTTP verb of this request.	Read-only
`system.datetime`	Returns a string that represents the current date and time in the system time zone of the gateway.	Read-only
`system.time`	DataPower Gateway only Returns a string that represents the current time in the system time zone of the gateway. Micro Gateway only Returns the current time in the system time zone of the gateway in a JSON object.	Read-only
`system.time.hour`	Returns a number 0 - 23 inclusive, representing the hour of the current time in the system time zone of the gateway.	Read-only
`system.time.minute`	Returns a number 0 - 59 inclusive representing the minute of the current time in the system time zone of the gateway.	Read-only
`system.time.seconds`	Returns a number 0 - 59 inclusive representing the seconds of the current time in the system time zone of the gateway.	Read-only
`system.date`	DataPower Gateway only Returns a string that represents the current date in the system time zone of the gateway. Micro Gateway only Returns a JSON object that represents the current date in the system time zone of the gateway.	Read-only
`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.	Read-only
`system.date.day-of-month`	Returns a number 1 - 31 representing the day of the month in the system time zone of the gateway.	Read-only
`system.date.month`	Returns a number 1 - 12 representing the month in the system time zone of the gateway.	Read-only
`system.date.year`	Returns a four-digit number that represents the year in the system time zone of the gateway.	Read-only
`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, `-04:00`.	Read-only

Table 2. OAuth context variables.

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	Permissions
DataPower Gateway only `oauth.access-token`	If the request is authenticated with OAuth, this variable contains the access token string.	Read/write
V5.0.6 + DataPower Gateway only `oauth.miscinfo`	V5.0.6 + This variable contains information explicitly included in the following headers: Authenticate URL Metadata URL	V5.0.6 + Read/write
DataPower Gateway only `oauth.not-after`	If the request is authenticated with OAuth, this variable contains the date when the token expires.	Read/write
DataPower Gateway only `oauth.not-before`	If the request is authenticated with OAuth, this variable contains the date when the token was issued.	Read/write
DataPower Gateway only `oauth.resource-owner`	If the request is authenticated with OAuth, this variable contains the name of the resource owner.	Read/write
DataPower Gateway only `oauth.scope`	If the request is authenticated with OAuth, this variable contains the scope of this access token.	Read/write
`oauth.introspect.active`	Always available. Boolean value.	Read/write
`oauth.introspect.response`	Always available. Shows the complete current response payload. Example payload value: `{"active":true, "client_id", "xxx-xxx", "token_type", "bearer", "scope":"neon"}`	Read/write

Name	Description	Permissions

Other variables might be available from the third party, in the form of: `oauth.introspect.<variable>`	Decoding the example above, the following variables are made available for further processing.
	`oauth.introspect.client_id: xxx-xxx`
	`oauth.introspect.token_type: bearer`
	`oauth.introspect.scope: neon`
		Read/write

V5.0.8 + 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	Permissions
`application.certificate.Base64`	Base64 format.	Read-only
`application.certificate.fingerprint`	Fingerprint	Read-only
`application.certificate.Version`	Version	Read-only
`application.certificate.SerialNumber`	Serial number	Read-only
`application.certificate.SignatureAlgorithm`	Signing algorithm	Read-only
`application.certificate.Issuer`	The issuer of the certificate	Read-only
`application.certificate.Subject`	Subject	Read-only
`application.certificate.NotBefore`	Not valid before this date	Read-only
`application.certificate.NotAfter`	Not valid after this date	Read-only
`application.certificate.SubjectPublicKeyAlgorithm`	Algorithm for the subject public key	Read-only
`application.certificate.SubjectPublicKeyBitLength`	Length for the subject public key	Read-only
`application.certificate.KeyValue.type`	Various context variables that depend on the algorithm and key. The following are possible context variables: • `application.certificate.KeyValue.RSAKeyValue.Modulus` • `application.certificate.KeyValue.RSAKeyValue.Exponent`	Read-only

Related concepts

- Variable references in API Connect
- The behavior of an assembly

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

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
  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 --type 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. 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 Implicity 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

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 `apis.routes[n].$ref` fields, for example:
		`apis:`
		`  'routes':`
		`    $ref: apidef.yaml`
		`    ...`

{{name}}	String	Value of `info.x-ibm-name` field.
{{title}}	String	Value of `info.title` field.
{{version}}	String	Value of `info.version` 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 `host`.
{{definitions}}	OpenAPI definitions objectconverted 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}}	OpenAPI definitions objectconverted 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 `host` field.
{{name}}	String	Value of `info.x-ibm-name` field.
{{paths}}	OpenAPI paths object	For a LoopBack API, contains the relative paths to the individual endpoints. The path is appended to `{{basePath}}` to construct the full URL.

Variable	Type	Description
{{pathsObj}}	OpenAPI paths object	For a LoopBack API, contains the relative paths to the individual endpoints. The path is appended to `{{basePath}}` to construct the full URL.
{{schemes}}	Array of string	Transfer protocol of the API. Values must be one of: `"http"`,`"https"`, `"ws"`, or `"wss"`.
{{targeturl}}	String	Value of `x-ibm-configuration.assembly.execute[invoke]`. Default is `$(runtime-url)$(request.path)`.
{{title}}	String	Value of `info.title` field.
{{version}}	String	Value of `info.version` field.

Related tasks

- Creating and using API and Product definitions templates

Related reference

- API and Product definition template examples

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing your V5 APIs

You manage your V5 APIs by using the API Manager user interface. 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, monetize, 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.
Note: The API Manager UI also includes the ability to create and edit APIs, however, the preferred method for these tasks is by using the API Designer UI in the developer toolkit. For more information about creating and editing APIs, see Creating API definitions by using the API Designer.
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, see Detailed System Requirements, then click the Supported Software tab (the provided link shows the requirements for the latest version; if you want to see the requirements for an earlier version, 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, you must activate the user account that your Cloud Manager administrator invited you to join.
- **API Manager user interface**
  The API Manager user interface provides a range of options for working with your APIs and Products, and managing security.

- **Working with Catalogs**
  Products must be staged to a Catalog and then published to Developer organizations to become available to application developers. 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 Developer organizations. V5.0.5+ The syndication feature in IBM API Connect means that you can also publish a Product to a Space in a Catalog.
- V5.0.7+ **Managing the application lifecycle**
  By using the application lifecycle capability, you can have separate Development and Production endpoints for the same API. Applications that are subscribed to use the API initially have Development status, and can call the API only through Development endpoints. When application testing is complete, the application developer can request to upgrade the application to Production status; when the request is approved, the application is upgraded and can call the API through Production endpoints.
- V5.0.5+ **Using syndication in IBM 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 Developer organizations**
  Manage the Developer organizations that access your APIs and Plans when their users sign up to use the Developer Portal.
- **Working with Apps**
  An App is the publish target for an application. An App references an API Connect collective. When you publish an application by using the API Designer or the CLI **apic publish** command, the application is deployed to the collective referenced by the App.
- **Creating APIs**
  An API is a set of functions that provides some business or technical capability and can be called by applications by using a defined protocol. In the context of API Manager, applications are typically mobile or web applications, and they use the HTTP protocol.
- **Security and authentication**
  In API Manager, you can use TLS profiles to secure the transmission of data through websites, and also configure user registries to securely authenticate your Catalogs and APIs. V5.0.5+ You can also use LTPA keys to secure the transmission of data across WebSphere® Application Server domains.
- **Working with Products in the API Manager**
  In API Connect, Plans and APIs are grouped together in Products. You use the API Designer to create, edit, and stage your Product, and the API Manager to manage the Product lifecycle, and the availability and visibility of APIs and Plans.
- **API Analytics**
  You can filter, sort, and aggregate your API event data, and then 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, 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.
- **Reference**
  Reference information for the API Manager component in API Connect.
- **API Manager tutorials**
  Tutorials for using API Manager.

Related information

- IBM API Connect overview
- Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Activating your API Manager user account

Before you can access the API Manager user interface, you must activate the user account that your Cloud Manager administrator invited you to join.

Before you begin

The operator of the Cloud Manager invited you to join IBM® API Connect .

Procedure

1. Complete the following steps to create your API Manager account:

   If the Identity Provider uses LDAP
   : An invitation email is sent. Click the link in the email to log in directly with your user credentials. No password is required.

   If the Identity Provider uses a local registry
   : An invitation email and an activation email are sent. In the activation email, click the link or paste it in a browser. The activation link takes you to a page where you enter your first name, last name and password. The username is the email address. 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: ! # $ %)

   If the Identity Provider uses an authentication URL
   : An invitation email and an activation email are sent. In the activation email, click the link or paste it in a browser. The activation link takes you to a page where you enter your credentials, which then become your username and password.

2. Click Sign up.
   Your API Connect home page is displayed.

Results

You have created your IBM API Connect account.

What to do next

- To access the API Manager UI in subsequent sessions, enter the URL

  ```
  https://host/apim
  ```

  where *host* is the fully qualified host name or IP address of the Management server.
- If your Management server is part of a cluster, you can use the fully qualified host name or IP address of any Management server in the cluster. However, doing this causes high load on that server so it is best practice to use the Management cluster host name that resolves to a load balancer that spreads users across all of the Management servers. For more information, see Load balancing in IBM API Connect.
- Depending on your role, you can now start to create Catalogs, Developer organizations, APIs and Products.

Related concepts

- Creating and configuring Catalogs
- Working with Products
- Load balancing in IBM API Connect

Related tasks

- Administering Developer organizations
- Creating APIs

Related reference

- API Manager user interface

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Manager user interface

The API Manager user interface provides a range of options for working with your APIs and Products, and managing security.

Primary banner

When you click the Navigate to ≡ icon, the **IBM API Connect** primary banner is displayed with a list of options. To pin the menu to your window, click the pin icon 📌 on the right side of the primary banner.

Use the menu options displayed in the navigation pane to select the activity you want to perform, as described in the following sections.

Favorites

The Favorites view enables you to mark Catalogs, APIs, and Products as favorites so that you can locate them more quickly in the API Manager user interface.

Dashboard

The dashboard displays your Catalogs. A Catalog is a staging target and behaves as a logical partition of the gateway and Developer Portal. The URL for API calls and the URL for 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 Catalog can be configured so that approval is enabled for each state transition.

For each Catalog, an Administrator can use settings to control whether an application developer can invite other developers to collaborate in the Catalog, and whether self-service onboarding is enabled.

Drafts

You use the Drafts view to design and configure your APIs and Products. An API is made visible in the Developer Portal by publishing one or more Products. A Product makes available a collection of APIs contained in one or more Plans that comprise a Product. In the Developer Portal, application developers gain access to APIs by registering applications to access the Plans contained in published Products.

Admin

Use the Admin view to manage users, assign roles to users, and create TLS profiles and user registries. A user is represented in the system as a top-level entity, and can belong to more than one organization. Within an organization, a user can have one or more roles that confer permissions. For example, you can specify that only users in a specified role have permission to stage Products to a Catalog. A TLS profile specifies a security certificate and the allowed protocol versions for the SSL/TLS services. A User Registry is a source or directory of users for API Connect. The User Registry tracks users and their authentication. You can use one of the following authentication mechanisms for your user registries:

- LDAP directory.
- Local user registry, which uses an internal database.
- Authentication URL.
- A user registry that implements the System for Cross-domain Identity Management (SCIM) standard.
  Important: Support for the SCIM registry type is deprecated and the feature will not be available after the IBM® API Connect Version 5.0 release.

Related concepts

- API Analytics
- Security and authentication

Related tasks

- Administering Developer organizations
- Administering user access

Related reference

- API Connect user roles

Related information

- Managing your V5 APIs
- Working with Products

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with Catalogs

Products must be staged to a Catalog and then published to Developer organizations to become available to application developers. 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 Developer organizations. V5.0.5 + The syndication feature in IBM 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.

V5.0.5 + 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.

Types of Catalog

You can apply the following type settings to a Catalog:

V5.0.2 and earlier Sandbox

V5.0.2 and earlier

Use the Sandbox setting for a development Catalog. In a development Catalog, staging and publishing actions are forced, meaning that if you republish a previously published Product it is overwritten without warning. If conflicts are found, they are automatically resolved by the system. Unpublish actions happen automatically. Furthermore, approvals are bypassed for publishing and lifecycle actions; you cannot configure the Catalog to require approval. Pending approvals are canceled when a non-development Catalog is converted to a development Catalog.

By default, a development Catalog is provided for you. A development Catalog must be used only for test purposes. When you use the test tool in a development Catalog, any Product that you test is forced through and overwrites staged and published Products even if the operations are being used on the Developer Portal. A Developer Portal created from a development Catalog must be used in the same way, that is, for testing purposes only and not for real cases.

V5.0.3 + Development

V5.0.3 +

In a development Catalog, staging and publishing actions are forced, meaning that if you republish a previously published Product it is overwritten without warning. If conflicts are found, they are automatically resolved by the system. Unpublish actions happen automatically.

Note:

- V5.0.4 and earlier In a development Catalog, approvals are bypassed for publishing and lifecycle actions; you cannot configure the Catalog to require approval. Pending approvals are canceled when a non-development Catalog is converted to a development Catalog.
- V5.0.5 + 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.

By default, a development Catalog is provided for you. A development Catalog must be used only for test purposes. A Developer Portal created from a development Catalog must be used in the same way, that is, for testing purposes only and not for real cases.

V5.0.4 and earlier When you use the test tool in a development Catalog, any Product that you test is forced through and overwrites staged and published Products even if the operations are being used on the Developer Portal.

V5.0.3 + Automatic subscription

V5.0.3 +

If you enable automatic subscription for a Catalog, testing of your APIs in the API Manager user interface is made easier because a test application is used, with a pre-supplied client ID and client secret, which is automatically subscribed to all the Plans in the Catalog, so you don't have to specify a plan or application when testing. The test application is not subject to rate limits. Automatic subscription is available only for a development Catalog.
V5.0.3 and earlier Note: Automatic subscription is supported only with the DataPower® Gateway.

Default
You can set one of your Catalogs to be the default Catalog. Then, calls to APIs that are published to that Catalog can use a shorter URL that does not include the Catalog name.

For more information on using the Developer Portal, see Discovering and using APIs.

- **Configuring the default Catalog permissions template**
  Use the default Catalog permissions template to pre-configure the Catalog permissions that are assigned to each role by default when a new Catalog is created in a provider organization.
- **Creating and configuring Catalogs**
  You create and configure your Catalogs by using the API Manager user interface. A development Catalog called Sandbox is provided by default, and you can create additional Catalogs as required. For example, you may require separate catalogs for APIs under development and for APIs released to production.
- V5.0.7+ **Using multiple DataPower Gateway services with a Catalog**
  You can configure a Catalog to use two or more DataPower Gateway services. Then by modifying the Gateway service endpoints, and configuring your DNS appropriately, you can route API calls to the required Gateway service.
- **Configuring sub paths for Developer Portal sites**
  You can increase the specificity of a Developer Portal site URL by using sub paths.
- V5.0.5+ **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.
- **Obtaining the publish target URL for a Catalog**
  Every Catalog has a unique URL containing all the information required to publish an API to the Catalog from the developer toolkit command line.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the default Catalog permissions template

Use the default Catalog permissions template to pre-configure the Catalog permissions that are assigned to each role by default when a new Catalog is created in a provider organization.

About this task

Each provider organization has its own default Catalog permissions template. The changes that you make to the default permissions affect any new Catalogs that are created in the provider organization thereafter. The permissions that are defined in existing Catalogs are unaffected. You can change the default Catalog permissions at any time before creating a new Catalog.
Note: You cannot change the default permissions for the organization owner.
After you have created a Catalog, you can change the permission settings for that Catalog to override the default permissions. For details, see Creating and configuring Catalogs.

Procedure

To configure the default Catalog permissions template, complete the following steps:

1. In the upper-right drop-down menu of the API Manager user interface, select the provider organization that you want to work with.
2. In the navigation pane, click Admin, then click the Roles tab.
3. Select the role for which you want to configure the default Catalog permissions.
4. Expand the Default Catalog Permissions section.
5. For each permission in the Default Catalog Permissions list, select or clear the check boxes to enable or disable the actions that the user can perform.
6. Click the Save icon 💾.

Creating and configuring Catalogs

You create and configure your Catalogs by using the API Manager user interface. A development Catalog called Sandbox is provided by default, and you can create additional Catalogs as required. For example, you may require separate catalogs for APIs under development and for APIs released to production.

For details on how to create and configure Catalogs, refer to one or other of the following subtopics according to your IBM® API Connect version:

- V5.0.7 + **Creating and configuring Catalogs (V5.0.7 and later)**
  These instructions describe how to create and configure Catalogs with IBM API Connect V5.0.7 and later.
- V5.0.6 and earlier **Creating and configuring Catalogs (V5.0.6 and earlier)**
  These instructions describe how to create and configure Catalogs with IBM API Connect V5.0.6 and earlier.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 +

Creating and configuring Catalogs (V5.0.7 and later)

These instructions describe how to create and configure Catalogs with IBM API Connect V5.0.7 and later.

Before you begin

If you are using IBM API Connect V5.0.6 or earlier, see Creating and configuring Catalogs (V5.0.6 and earlier).

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.

Procedure

To create your Catalog:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Add a Catalog.
   a. Click Dashboard.
   b. Click Add > Catalog.
      The Add Catalog window is displayed.
   c. Enter the name of your new Catalog in the Display Name field.
      The name that you provide is displayed on your dashboard.
   d. Enter the text that you want to form the Catalog segment of the URL, in the Name field.
      Note:
      - The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), or hyphen characters (-). A hyphen cannot be used as the first or last character in the name. For more information, see Sub paths for Developer Portal sites.
      - The following names are reserved and cannot be used:
        - apis
        - plans
        - products
   e. Click Add. Your Catalog is created, and is displayed on your dashboard.
   f. To configure your Catalog, click the name of the Catalog, then click Settings.

3. To configure the general settings for the Catalog, click Overview, then proceed with the following steps:
    a. If the new Catalog is a development Catalog, enable development mode by selecting Development mode.
    Development and Production catalogs behave differently:
        - Development catalog
          In a development Catalog, staging and publishing actions are forced, meaning that if you republish a previously published Product it is overwritten without warning. If conflicts are found, they are automatically resolved by the system. Unpublish actions happen automatically.

          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.

        - Production catalog
          You will be prevented from publishing a Product to a Production 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. 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 you create a new Product that contains one or more modified APIs, you must create new versions of those APIs for inclusion in the Product. If the Product contains a modified API and there is already a published API with the same name and version, your changes will not be published.

    b. If you want to enable automatic subscription for the Catalog, select Automatic subscription.
    Enabling automatic subscription makes testing of your APIs easier because a test application is used, with a pre-supplied client ID and client secret, which is automatically subscribed to all the Plans in the Catalog, so you don't have to specify a plan or application when testing.
    Note: Automatic subscription is available only for a development Catalog.
    c. If the Catalog is the default staging Catalog, select Default. Then, calls to APIs that are published to the Catalog can use a shorter URL that does not include the Catalog name.
    Note: You can only select Default for one Catalog.
    d. To enable the partition of this Catalog into Spaces, select the Spaces toggle.
    For more information, see Using syndication in IBM API Connect.
    e. Optional: In the API endpoint for unenforced APIs section, enter a custom API URL.
    For any API, there are two possible endpoints to consider:
        - The gateway endpoint at which the API is invoked. The URL has the following format:

          ```
          https://gateway_cluster_hostname/organization_name/catalog_name
          ```

        - The endpoint that is visible to the consumer in the Developer Portal.

    If you do not configure a custom API URL, 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 custom API URL. The custom API URL represents the endpoint by which the API is known externally; that is, the endpoint that is published to the Developer Portal and used by an application developer to invoke or advertise the API. The URL represents only the server endpoint and does not include the path to the API itself; for example:

    ```
    https://custom_host.domain
    ```

    The ability to call an API without supplying the provider organization or Catalog name in the URL is called *host to catalog mapping*.
    If you are using a third-party gateway or external load balancer in this Catalog, supply the URL in this field. Any API endpoints that are displayed in the developer portal will then reflect the specified URL. These endpoints exist on the third-party gateway or load balancer and project a virtual address, exposed to API consumers, that is mapped to the API proxy or API assembly endpoints on the gateway. The endpoints that are derived from the custom API URL are typically published in production developer portals to advertise the address of the API.

    If you specify a custom API URL for a Catalog, it takes precedence over any host name that you specify when configuring the API; for more information, see specifying an alternative host for an API.

4. To configure the Gateway settings for the Catalog, click Gateways > Configure, then proceed with the following steps:
    a. Choose one of the following options depending on the gateway that is to be used with APIs that are published to this Catalog.
        i. If you are using the DataPower® gateway, select the DataPower Gateway services that you want to use with this Catalog.
        If you select two or more Datapower Gateway services, you can modify the Gateway service endpoint URLs, and configure your DNS appropriately, so that API calls are routed to the required Gateway service; for more information, see Using multiple DataPower Gateway services with a Catalog.

        Gateway services are configured in the Cloud Manager user interface; for more information, see Configuring the initial Gateway service and Adding more Gateway services.

        ii. If you are using the Micro Gateway with the IBM API Connect Professional or Enterprise offering then, under Collectives, select the collective to which the Micro Gateway has been published, and click Next; for more information on

configuring the Micro Gateway, see Adding a Micro Gateway to a Catalog.
Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

   iii. If you are using the standalone Micro Gateway with the IBM API Connect Essentials offering, select API Connect Standalone Micro Gateway and click Next; for more information on configuring the standalone Micro Gateway, see Adding a Micro Gateway to a Catalog.
   Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

  b. Click Done to save your changes.
  When you modify gateway settings, the change takes effect almost immediately for all APIs in the catalog

5. To configure the Developer Portal, click Portal and proceed with the following steps:
   a. Select IBM Developer Portal or Other, then enter the appropriate URL in the related text field.
   If you select IBM Developer Portal, enter the URL in 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.
   For more information on using the Developer Portal, see Discovering and using APIs.

   b. To enable development application workflow for applications that are created in the Developer Portal that is associated with this Catalog, select Development Application Workflow.
   When development application workflow is enabled, any application that an application developer creates in the Developer Portal has Development status by default, and can call APIs in the Catalog only through dedicated DataPower Gateway service development endpoints. When ready, the application developer can request to upgrade the application to Production status. After the upgrade request is approved, the application can call APIs through the DataPower Gateway service production endpoints. For more information, see Managing the application lifecycle.

   c. In the User Registry drop-down menu, select an existing User Registry or create a new one to be used by the Developer Portal. The user registry that you select is used to authenticate login to the Developer Portal. For more information, see Working with user registries. If you select Portal Delegated User Registry, see Portal Delegated User Registry for more information.
   Note: After the User Registry is specified for a Catalog, you can change it only if no portal users have been created, and it's not set to be a Portal Delegated User Registry.
   d. Specify your user registry options; for more information, see Working with user registries.
   e. Optional: In the Portal API Endpoint section, enter a host name for Developer Portal API calls. The host name that you enter can be the host name of your Management service.
   To access the Developer Portal API within the context of a Developer Portal, you must configure the base host name for the Developer Portal API calls.

   This action allows API Manager to map your host name to the provider organization and Catalog of the Developer Portal API calls instead of requiring you to look it up and include them in your calls.

6. To configure the Permissions that each role in API Manager has, click Roles.
   To add permissions to a role, or to remove permissions from a role, complete the following steps:
   a. Select a role from the list of available roles. You can configure the actions that permissions have for the following default roles that are available:
      - Administrator
      - Product Manager
      - API Developer
      - API Administrator
      - Catalog Owner

   You can configure the actions that permissions have for any custom roles that you have created, and are in the list of available roles. For details of how to configure a custom role, see Creating custom roles.
   Note: The Administrator and Catalog Owner role have all permissions. You cannot alter the permissions for the Catalog Owner.
   b. After you have selected the role that you want to configure the permissions for, select or clear any check boxes for any possible actions that you want to have enabled or disabled for a permission.
   c. To ensure a role can manage user-defined policies, select Manage for the Catalogs Settings permission.

    d. To grant a role the permission to approve a specific lifecycle state change, select the state change in the Product Lifecycle Approvals section.
    e. To specify the Product lifecycle state changes for which you want to enforce approval, click Approvals in the Catalog settings navigation pane, then select the required state changes.
    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.

   Note: The permissions that are assigned to each role by default when you create a new Catalog are determined by the settings in the default Catalog permissions template. For details, see Configuring the default Catalog permissions template.

7. To define the policies in API Manager, follow the procedure that is outlined in Importing a user-defined policy into a Catalog.
   DataPower Gateway only Note: When you import a policy, its implementation is imported into all Gateway devices associated with the Gateway Cluster 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.
   Micro Gateway only Note: Policy implementation is a manual import for the Micro Gateway. For more information, see Packaging and importing your policies into IBM API Connect.
   For more information about how to create and manage user-defined policies, see User defined policies.

   Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), support for implementing your own policies is *not* available.
8. To import an extension schema into your Catalog so that you can extend the OpenAPI (Swagger 2.0) specification, click Extensions; for full details, see Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Manager UI).
9. Click the Save icon 💾.

Results

A message is displayed indicating that the save is successful.

Related concepts

- Developer Portal REST APIs
- Authoring policies

Related tasks

- Specifying the identity provider
- Working with user registries

Related information

- Working with Products

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 and earlier

Creating and configuring Catalogs (V5.0.6 and earlier)

These instructions describe how to create and configure Catalogs with IBM API Connect V5.0.6 and earlier.

Before you begin

If you are using IBM API Connect V5.0.7 or later, see Creating and configuring Catalogs (V5.0.7 and later).

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.

Procedure

To create your Catalog:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Add a Catalog.
   a. Click Dashboard.
   b. Click Add > Catalog.
      The Add Catalog window is displayed.
   c. Enter the name of your new Catalog in the Display Name field.
      The name that you provide is displayed on your dashboard.
   d. Enter the text that you want to form the Catalog segment of the URL, in the Name field.
      Note:
      - The Name field can contain only lowercase alphanumeric characters (a-z and 0-9), or hyphen characters (-). A hyphen cannot be used as the first or last character in the name. For more information, see Sub paths for Developer Portal sites.
      - The following names are reserved and cannot be used:
        - apis
        - plans
        - products
   e. Click Add. Your Catalog is created, and is displayed on your dashboard.
3. V5.0.2 and earlier To configure the Catalog, click the name of the Catalog, then click Settings > Configuration, and proceed with the following steps:
   a. Click Configure Gateway and choose one of the following options depending on the gateway that is to be used when APIs are published to this Catalog, then click Done when finished.
      i. If you are using the DataPower® gateway, select the required gateway service under Datapower Services. Gateway services are configured in the Cloud Manager user interface; for more information, see Configuring the initial Gateway service and Adding more Gateway services.
      ii. If you are using the Micro Gateway with the IBM API Connect Professional or Enterprise offering then, under Collectives, select the collective to which the Micro Gateway has been published, and click Next; for more information on configuring the Micro Gateway, see Adding a Micro Gateway to a Catalog.
      Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
      iii. If you are using the standalone Micro Gateway with the IBM API Connect Essentials offering, select API Connect Standalone Micro Gateway and click Next; for more information on configuring the standalone Micro Gateway, see Adding a Micro Gateway to a Catalog.
      Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
   b. If the new Catalog is a development Catalog, select Sandbox.
      Development and Production catalogs behave differently:
      - Development catalog
        In a development Catalog, staging and publishing actions are forced, meaning that if you republish a previously published Product it is overwritten without warning. If conflicts are found, they are automatically resolved by the system. Unpublish actions happen automatically.

        Note: Approval boxes are not displayed for development Catalogs. You cannot enable the approval process for lifecycles.
      - Production catalog
        You will be prevented from publishing a Product to a Production 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. 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 you create a new Product that contains one or more modified APIs, you must create new versions of those APIs for inclusion in the Product. If the Product contains a modified API and there is already a published API with the same name and version, your changes will not be published.
   c. If the Catalog is the default staging Catalog, select Default. Then, calls to APIs that are published to that Catalog can use a shorter URL that does not include the Catalog name.
      Note: You can only select Default for one Catalog.
   d. Optional: Populate the following fields:

Custom Gateway URL

In the Custom Gateway URL text field, enter a URL. You use the Custom Gateway URL if you want to achieve custom branding of URLs for APIs that are deployed to IBM API Connect, rather than using the default URL that API Manager generates.

By default, the IBM API Connect gateway URL has the following format:

```
https://gateway_cluster_hostname/organization_name/catalog_name
```

However, you can override the default by specifying a URL that is more appropriate for your enterprise; for example, `https://api.mycompany.com`. Any API endpoints that are displayed in the Developer Portal will then reflect the specified URL.

Note:

- You must configure your DNS entry to send requests for your custom domain `https://api.mycompany.com` to your gateway cluster load balancer endpoint so that API calls can be received by the gateway.
- V5.0.3 and earlier You must also configure transformation of the path of the HTTP request so that it reaches the gateway as the internal format `/organization_name/catalog_name/api_root`, and not just `api_root`. This step is not required in V5.0.4 or later.
- For the endpoints of an API to reflect your custom gateway URL, you must configure the API to be enforced by the IBM API Connect gateway; for details, see specifying an alternative host for an API.
- Ensure that the same custom gateway URL is not applied to multiple Catalogs as the behavior in that scenario is undefined.

Tip: When you call the API, you can also set the HTTP host header on the API request to the value that you specified in the Custom Gateway URL field.

Custom API URL

In the Custom API URL text field, enter the URL. You use the Custom API URL to specify the URL for APIs that are deployed to a third party gateway.

The Custom API URL represents the endpoint by which the API is known externally; that is, the endpoint that is published to the developer portal and used by an application developer to invoke or advertise the API.

If you are using a third party gateway or external load balancer in this Catalog, supply the URL in this field. Any API endpoints that are displayed in the developer portal will then reflect the specified URL. These endpoints exist on the third party gateway or load balancer and project a virtual address, exposed to API consumers, that is mapped to the API proxy or API assembly endpoints on the gateway. The endpoints that are derived from the custom API URL are typically published in production developer portals to advertise the address of the API.

Note: If you specify a custom API URL for a Catalog, it takes precedence over any host name that you specify when configuring the API; for more information, see specifying an alternative host for an API.

Hostname for Developer Portal API Calls

In the Portal API Endpoint window area, enter a host name for Developer Portal API calls. The host name that you enter can be the host name of your Management service.
To access the Developer Portal API within the context of a Developer Portal, you must configure the base host name for the Developer Portal API calls.

This action allows API Manager to map your host name to the provider organization and Catalog of the Developer Portal API calls instead of requiring you to look it up and include them in your calls.

4. V5.0.3+ To configure the Catalog, click the name of the Catalog, click Settings, then proceed with the following steps:
   a. Click Gateway > Configure Gateway and choose one of the following options depending on the gateway that is to be used when APIs published to this Catalog, and click Done when finished.
      i. If you are using the DataPower gateway, select the required gateway service under Datapower Services. Gateway services are configured in the Cloud Manager user interface; for more information, see Configuring the initial Gateway service and Adding more Gateway services.
      ii. If you are using the Micro Gateway with the IBM API Connect Professional or Enterprise offering then, under Collectives, select the collective to which the Micro Gateway has been published, and click Next; for more information on configuring the Micro Gateway, see Adding a Micro Gateway to a Catalog.
      Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
      iii. If you are using the standalone Micro Gateway with the IBM API Connect Essentials offering, select API Connect Standalone Micro Gateway and click Next; for more information on configuring the standalone Micro Gateway, see Adding a Micro Gateway to a Catalog.
      Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

b. If the new Catalog is a development Catalog, enable development mode:
   - V5.0.4 and earlier Click Info and select the Development mode toggle.
   - V5.0.5 + Click Overview and select the Development mode toggle.

   In a development Catalog, staging and publishing actions are forced, meaning that if you republish a previously published Product it is overwritten without warning. If conflicts are found, they are automatically resolved by the system. Unpublish actions happen automatically.

   Note:
   - V5.0.4 and earlier In a development Catalog, approvals are bypassed for publishing and lifecycle actions; you cannot configure the Catalog to require approval. Pending approvals are canceled when a non-development Catalog is converted to a development Catalog.
   - V5.0.5 + 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.

c. If you want to enable automatic subscription for the Catalog, select Automatic subscription.
   Enabling automatic subscription makes testing of your APIs easier because a test application is used, with a pre-supplied client ID and client secret, which is automatically subscribed to all the Plans in the Catalog, so you don't have to specify a plan or application when testing.
   Note:
   - Automatic subscription is available only for a development Catalog.
   - V5.0.3 and earlier Automatic subscription is supported only with the DataPower Gateway.

d. If the Catalog is the default staging Catalog, select Default. Then, calls to APIs that are published to the Catalog can use a shorter URL that does not include the Catalog name.
   Note: You can only select Default for one Catalog.

e. V5.0.5 + To enable the partition of this Catalog into Spaces, select the Spaces toggle.
   For more information, see Using syndication in IBM API Connect.

f. Optional: Click Endpoints, and populate the following fields:

   Custom Gateway URL

   In the Custom Gateway URL text field, enter a URL. You use the Custom Gateway URL if you want to achieve custom branding of URLs for APIs that are deployed to IBM API Connect, rather than using the default URL that API Manager generates.

   By default, the IBM API Connect gateway URL has the following format:

   ```
   https://gateway_cluster_hostname/organization_name/catalog_name
   ```

   However, you can override the default by specifying a URL that is more appropriate for your enterprise; for example, `https://api.mycompany.com`. Any API endpoints that are displayed in the Developer Portal will then reflect the specified URL.
   Note:
   - You must configure a DNS entry that maps your custom hostname and domain to the default gateway URL.
   - For the endpoints of an API to reflect your custom gateway URL, you must configure the API to be enforced by the IBM API Connect gateway; for details, see specifying an alternative host for an API.
   - Ensure that the same custom gateway URL is not applied to multiple Catalogs as the behavior in that scenario is undefined.

   Custom API URL

   In the Custom API URL text field, enter the URL. You use the Custom API URL to specify the URL for APIs that are deployed to a third party gateway.

   The Custom API URL represents the endpoint by which the API is known externally; that is, the endpoint that is published to the developer portal and used by an application developer to invoke or advertise the API.

   If you are using a third party gateway or external load balancer in this Catalog, supply the URL in this field. Any API endpoints that are displayed in the developer portal will then reflect the specified URL. These endpoints exist on the third party gateway or load balancer and project a virtual address, exposed to API consumers, that is mapped to the API proxy or API assembly endpoints on the gateway. The endpoints that are derived from the custom API URL are typically published in production developer portals to advertise the address of the API.

   Note: If you specify a custom API URL for a Catalog, it takes precedence over any host name that you specify when configuring the API; for more information, see specifying an alternative host for an API.

   Hostname for Developer Portal API Calls

   In the Portal API Endpoint window area, enter a host name for Developer Portal API calls. The host name that you enter can be the host name of your Management service.
   To access the Developer Portal API within the context of a Developer Portal, you must configure the base host name for the Developer Portal API calls.

   This action allows API Manager to map your host name to the provider organization and Catalog of the Developer Portal API calls instead of requiring you to look it up and include them in your calls.

5. To configure the Developer Portal, click Portal and proceed with the following steps:
    a. Select IBM Developer Portal or Other, then enter the appropriate URL in the related text field.
    If you select IBM Developer Portal, enter the URL in 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.
    For more information on using the Developer Portal, see Discovering and using APIs.

    b. In the User Registry drop-down menu, select an existing User Registry or create a new one to be used by the Developer Portal. The user registry that you select is used to authenticate login to the Developer Portal. For more information, see Working with user registries. If you select Portal Delegated User Registry, see Portal Delegated User Registry for more information.
6. V5.0.4 and earlier To configure the Permissions for the roles in API Manager, click Permissions.
The Product management permissions to which you can assign roles are displayed:

Stage
    Allow staging of Products to the Catalog.
View
    Allow viewing and listing of Products in the Catalog.
Manage
    Allow management of Product lifecycle (publishing, deprecating, retiring and archiving).
Approve
    Enable approvals for Product lifecycle state changes.

    a. To add a Role to the Stage, View or Manage permissions, click the corresponding + icon. The following list shows the default available roles:
    - Administrator
    - Product Manager
    - API Developer
    - Publisher

    The Owner role has all permissions.

    b. Enable approvals for Product lifecycle state changes by selecting the required check boxes, then clicking the corresponding + icon to assign the roles that have permission to approve a lifecycle state change.
    The lifecycle state changes that you select are those for which you want to enforce approval. For example, if you select Publish but leave the others 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 assign the roles that have permission to manage user-defined policies, click the + icon in the Policy Management section. The policy management permissions are displayed, enabling you to assign roles as required.
7. V5.0.5 + To configure the Permissions that each role in API Manager has, click Roles.
To add permissions to a role, or to remove permissions from a role, complete the following steps:
    a. Select a role from the list of available roles. You can configure the actions that permissions have for the following default roles that are available:
    - Administrator
    - Product Manager
    - API Developer
    - API Administrator
    - Catalog Owner

    You can configure the actions that permissions have for any custom roles that you have created, and are in the list of available roles. For details of how to configure a custom role, see Creating custom roles.
    Note: The Administrator and Catalog Owner role have all permissions. You cannot alter the permissions for the Catalog Owner.
    b. After you have selected the role that you want to configure the permissions for, select or clear any check boxes for any possible actions that you want to have enabled or disabled for a permission.
    c. To ensure a role can manage user-defined policies, select Manage for the Catalogs Settings permission.
    d. To grant a role the permission to approve a specific lifecycle state change, select the state change in the Product Lifecycle Approvals section.
    e. To specify the Product lifecycle state changes for which you want to enforce approval, click Approvals in the Catalog settings navigation pane, then select the required state changes.
    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.

Note: The permissions that are assigned to each role by default when you create a new Catalog are determined by the settings in the default Catalog permissions template. For details, see Configuring the default Catalog permissions template.

8. To define the policies in API Manager, follow the procedure that is outlined in Importing a user-defined policy into a Catalog.

   DataPower Gateway only Note: When you import a policy, its implementation is imported into all Gateway devices associated with the Gateway Cluster 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.

   Micro Gateway only Note: Policy implementation is a manual import for the Micro Gateway. For more information, see Packaging and importing your policies into IBM API Connect.

   For more information about how to create and manage user-defined policies, see User defined policies.

   Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), support for implementing your own policies is *not* available.
9. To import an extension schema into your Catalog so that you can extend the OpenAPI (Swagger 2.0) specification, click Extensions; for full details, see Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Manager UI).
10. V5.0.2 and earlier Click Save.
11. V5.0.3+ Click the Save icon 💾.

Results

A message is displayed indicating that the save is successful.

Related concepts

- Developer Portal REST APIs
- Authoring policies

Related tasks

- Specifying the identity provider
- Working with user registries

Related information

- Working with Products

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Using multiple DataPower Gateway services with a Catalog

You can configure a Catalog to use two or more DataPower® Gateway services. Then by modifying the Gateway service endpoints, and configuring your DNS appropriately, you can route API calls to the required Gateway service.

The following examples describe possible scenarios when you use multiple Gateway services with a Catalog:

- To route API calls according to geography, so that calls are routed to the Gateway service that is most local to the calling application.
- To control API calls according to application type, so that calls from development applications are made to different Gateway services than calls from production applications. For more information, see Managing the application lifecycle.

Note: Rate limits are not calculated across multiple gateway services. They are calculated only for gateway servers within a gateway service.

Configuring a Catalog to use multiple DataPower Gateway services

To configure a Catalog to use multiple Gateway services, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. Click Settings > Gateways > Configure
3. Select the DataPower Gateway services that you want to use, then click Done. The selected Gateway services are listed.
   Note: Gateway services are defined by using the Cloud Manager UI; for more information, see Configuring the initial Gateway service and Adding more Gateway services.
4. To modify the endpoint for a Gateway service, click the Endpoint field alongside the Gateway service and enter the required value. For example, if you want to load balance across all Gateway services, enter the **same** virtual endpoint URL in each of the Endpoint fields, then configure your DNS appropriately.
5. (Optional) Enter a description for a Gateway service endpoint in the Endpoint description field.

For more information on Catalog configuration, see Creating and configuring Catalogs.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring sub paths for Developer Portal sites

You can increase the specificity of a 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 have created 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

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 Catalog Members > Manage permission. For more information on assigning Catalog permissions to a role, see Creating and configuring Catalogs.

Procedure

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. Click the Members tab.
   The Members tab lists all the members in the Catalog and shows the roles assigned to each. If Spaces are enabled in the Catalog, the Members tab lists the members of all Spaces in the Catalog. To view the roles that are assigned to a user in a Space, click the down arrow in the Scope column. A user might be a member of more than one Space.

   For information on adding users to a Space, see Managing Space membership.

   For more information on Spaces, see Using syndication in IBM API Connect.

3. Add a new user to the Catalog by completing the following steps:
   a. Click Add.
      The Add User window opens.
   b. Specify the user; you can search for, and select, an existing user, or you can enter the email address of a new user.
   c. 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.
   d. Click Add.
      The user is added to the Members page, and an activation email is sent to the user.

   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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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.

Before you begin

You must possess Catalog edit permissions to complete this task.

Procedure

To define the policies in API Manager, proceed with the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Dashboard and click the Catalog that you want to import the user-defined policy into.
3. Click Settings, then click Policies.
   The Policies configuration page is displayed.
4. Click the Import policy icon ⬆ and click Browse to select the user-defined policy you require.
   DataPower Gateway only Note: When you import a policy, its implementation is imported into all Gateway devices associated with the Gateway Cluster 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.
   Micro Gateway only Note: Policy implementation is a manual import for the Micro Gateway. For more information, see Packaging and importing your policies into IBM API Connect.
   IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

   For more information about how to create and manage user-defined policies, see User-defined policies.
5. After you have selected the user-defined policies, click Import.

Results

The new user-defined policy is shown in the list of available policies. 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

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the publish target URL for a Catalog

Every Catalog has a unique URL containing all the information required to publish an API to the Catalog from the developer toolkit command line.

About this task

When you publish an API from the developer toolkit command line, you can create a `catalog` configuration property that defines a URL that contains the target server, provider organization, and Catalog, so that you do not have to supply these values as command options. You can use the API Manager user interface to obtain the required command that sets the publish target URL for a specific Catalog.

For information on publishing an API from the developer toolkit command line, see Publishing APIs and applications and the following tutorial Creating and publishing an API definition from the command line.

V5.0.5 + Note: If Spaces are enabled for a Catalog, Products and their associated APIs can be published only to a Space within that Catalog. For more information about Spaces, see Using syndication in IBM API Connect.

Procedure

To obtain the command that sets the publish target URL for a Catalog, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. On the Dashboard page, click the Show catalog identifier icon 🔗 for the required Catalog.
3. Copy the command from the Catalog identifier window, then click OK.
   The command has the following format:

```
apic config:set catalog=publish_target_URL
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Managing the application lifecycle

By using the application lifecycle capability, you can have separate Development and Production endpoints for the same API. Applications that are subscribed to use the API initially have Development status, and can call the API only through Development endpoints. When application testing is complete, the application developer can request to upgrade the application to Production status; when the request is approved, the application is upgraded and can call the API through Production endpoints.

Configuring IBM API Connect to use the application lifecycle

To configure the IBM® API Connect to use the application lifecyle, you must complete the following steps:

1. Configure separate Development and Production DataPower® Gateway services in the Cloud Manager user interface. You configure a DataPower Gateway service to be either a Development or Production gateway service as follows:
   a. Click Services.
   b. Either configure a new DataPower Gateway service, or modify an existing service, by completing one or other of the following actions:
      - To create a new service, click Add > Add DataPower Service.
      - To modify an existing service, click the Service Settings icon ⚙ alongside the service that you want to modify.
   c. In the Supported Application Types field, select the type of application traffic that can flow through this DataPower Gateway service. The options are as follows:
      - Production: this Gateway service is for use with production applications only.
      - Development: this Gateway service is for use with development applications only.
      - Both (default): this Gateway service supports both development and production applications.
   d. When you are finished, click Create or Save depending on whether you are creating a new service or modifying an existing one.
2. Configure the Catalog to use the Development and Production DataPower Gateway services; for details, see Using multiple DataPower Gateway services with a Catalog.
3. Enable development application workflow in the Catalog; complete the following steps:
   a. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   b. Click Settings > Portal, then select Development Application Workflow.
   c. Click the Save icon 💾.

Using application lifecycle in the Developer Portal

If development application workflow is enabled for the Catalog associated with your Developer Portal, then when you register a new application it initially has Development status. When you then subscribe the application to use an API, it can make API calls only to Development endpoints. When you have completed your application testing, you can request to upgrade your application to Production status; for details, see Upgrading a Development application to Production status.

After the application upgrade request has been approved, the application can make calls to Production endpoints.

Approving an application upgrade request

A request to upgrade a Development application to Production status is approved or declined in the API Manager UI; for details, see Approving application upgrade requests.

Related information

- Configuring the initial Gateway service
- Adding more Gateway services
- Registering an application
- Signing up to use an API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Approving application upgrade requests

To approve or decline requests by application developers to upgrade a Development application to Production status, use the Approvals tab in the Catalog that is associated with the Developer Portal from which the application upgrade request was submitted.

Before you begin

To approve application upgrade requests in a Catalog, you must either be the owner of the associated API provider organization, or you must be assigned a user role that has Subscription and Application Approvals > Manage permission in the Catalog. For information on configuring permissions for a Catalog, see Creating and configuring Catalogs (V5.0.7 and later). For information on assigning user roles, see Managing Catalog membership.

About this task

If development application workflow is enabled for a Catalog, any application that an application developer registers in the associated Developer Portal initially has Development status. A development application can call APIs in the Catalog only through dedicated development endpoints. When application testing is complete, the application developer can request to upgrade the application to Production status; the request is submitted for approval. After the upgrade request is approved, the application can call APIs through production endpoints.

Procedure

To approve application upgrade requests, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. Select the Approvals tab, then locate the application upgrade request that you want to deal with.
3. Click the Approve ✓, Decline ⊗, or Cancel Request ✕ icon as required. Depending on the action you selected, either the Approve this request, Decline this request, or Cancel this request window opens.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Using syndication in IBM 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.

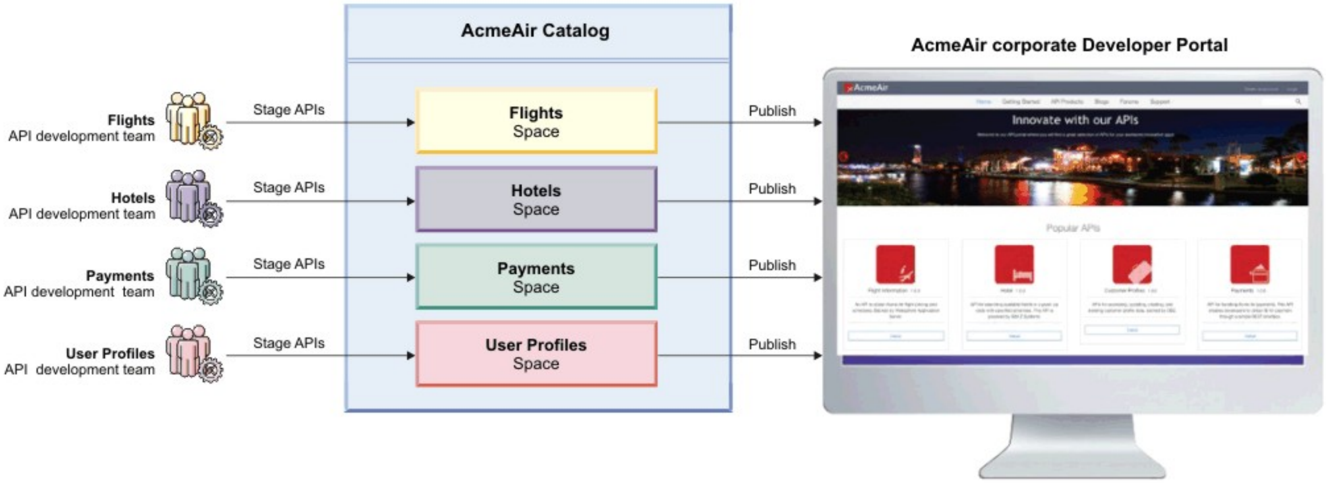
Each Space has its own Product lifecycle management, subscription approvals, and analytics data. You use Space specific access control to restrict user access to each Space; for example, a developer in the Flights team is able to stage APIs only to the Flights Space.

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 owner of a Space, and delete a Space from a Catalog.
- **Working with Spaces**
  If Spaces are enabled in a 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

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.

About this task

By default, Spaces are disabled in a Catalog. You enable Spaces by modifying the Catalog settings.

Note: You can also enable Spaces by using the developer toolkit CLI; for details, see Toolkit command summary.

Procedure

To enable Spaces in a Catalog, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. Click Settings > Overview, and click the Spaces slider control.
3. In the Enable Spaces window, click Enable, then click the Save icon 💾.
   A Manage spaces link is displayed beneath the Spaces slider control, and a Spaces link is added to the navigation pane. You can manage your Spaces by clicking either of these links.

Results

Spaces are enabled for your Catalog, and a default Space, called New Space, is created. The Catalog owner is also the owner of the default Space, but you can change the owner; for details, see Creating, modifying, and deleting Spaces.
Note: 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 Designer, and see Publishing APIs for publishing by using the developer toolkit.

What to do next

You can change the name of default Space, and you can create and configure additional Spaces in your Catalog. For more information, see Creating, modifying, and deleting Spaces and Working with Spaces.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

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 owner of a Space, 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. Click Settings > Spaces.
   The Spaces settings option is available only if Spaces are enabled in the Catalog; see Enabling Spaces in a Catalog.

Note: You can also create, modify, and delete Spaces by using the developer toolkit CLI; for details, see Toolkit command summary.

Procedure

- To create a new Space, complete the following steps:
  1. Click Add.
  2. In the Create Space window, enter the Display Name and Name for the Space and, optionally, a Description.
     Note:
     - The value you specify in the Name field is a single string that is used to identify the Space in developer toolkit CLI commands.
     - The following names are reserved and cannot be used:
       - apis
       - plans
       - products
  3. Click Save to create the Space.
  4. Click the Save icon 💾 to save the Catalog.

  Note: As the user who creates the Space, you are automatically the Space owner and have all Space permissions.

- To modify the details of a Space, complete the following steps:
  1. Click the Manage icon ⋮ alongside the Space whose details you want to modify.
  2. Click Edit, modify the details as required, then click Save.
  3. Click the Save icon 💾 to save the Catalog.
- To change the owner of a Space, complete the following steps:
  1. Click the Manage icon ⋮ alongside the Space whose owner you want to change.
  2. Click Change Owner, specify the new owner, then click Add.
  3. Click the Save icon 💾 to save the Catalog.
- To delete a Space, complete the following steps:
  1. Click the Manage icon ⋮ alongside the Space that you want to delete.
  2. Click Delete, then click OK to confirm deletion.
  3. Click the Save icon 💾 to save the Catalog.

  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. However, if a Catalog contains only one Space then you can disable Spaces for the Catalog. Any Products that have been staged or published to the Space remain in the Catalog. To disable Spaces for a Catalog, complete the following steps:
    1. Click Settings > Overview, then move the Spaces slider to the Off position.
    2. Click the Save icon 💾 to save the Catalog.
  - If an API is used in a single Space within a Catalog, the Space name is included in the analytics for that API. However, if the API is used in multiple Spaces within a Catalog, the API invoke must be associated with a Product plan to prevent ambiguity and ensure that the API can be included in analytics.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Working with Spaces

If Spaces are enabled in a 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
   The side bar of the Settings tab contains the following links:
   - Overview
   - Approvals
   - Gateway
   - Endpoints
   - Portal
   - Roles
   - Policies
   - Extensions

   Use the Roles link to manage Space security; for more information, see Managing user access in a Space.

   The remaining links in the side bar display read-only pages that show the corresponding settings for the Catalog that contains the Space.

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 a 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 a 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 a 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 a 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 a 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.
- **Configuring the default Space permissions template**
  Use the default Space permissions template to pre-configure the Space permissions that are assigned to each role by default when a new Space is created in a Catalog.
- V5.0.7+ **Managing Gateways in a Space**
  If Spaces are enabled in a Catalog, and the Catalog is configured to use one or more DataPower® Gateway services, you can separately control which of those Gateway services is used by each of the Spaces in the Catalog.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Managing Products in a Space

If Spaces are enabled in a 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 API 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
   The Products tab lists only the Products in the selected Space.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Managing subscription requests in a Space

If Spaces are enabled in a 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
3. Click the Approvals tab
   The Approvals tab 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Managing application subscriptions in a Space

If Spaces are enabled in a 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
3. Click Community > Subscriptions.
   The Subscriptions page lists the application subscriptions that are specific to the selected Space.

What to do next

For details of the management tasks that you can perform on application subscriptions in a Space, see Managing application subscriptions.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Managing Space membership

If Spaces are enabled in a 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 Space Members > Manage permission. 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
3. Click the Members tab.
   The Members tab lists the users that have been added to the selected Space, together with the users that have been added to the Catalog that contains the Space. For details of adding users to a Catalog, see Managing Catalog membership.
4. Add a new user to the Space by completing the following steps:
   a. Click Add.
      The Add User window opens.
   b. Specify the user; you can search for, and select, an existing user, or you can enter the email address of a new user.
   c. 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.
   d. Click Add.
      The user is added to the Members page, and an activation email is sent to the user.

   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 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: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Managing user access in a Space

If Spaces are enabled in a 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 Space Settings > Manage permission. For information on adding users to a Space and assigning user roles, see Managing Space membership.

About this task

The permissions that are assigned to each role by default you create a new Space are determined by the settings in the default Space permissions template. For details, see Configuring the default Space permissions template

Procedure

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
3. Click Settings > Roles, then select the role for which you want to configure the permissions.
4. For each permission in the Space Permissions list, select or clear the check boxes to enable or disable the actions that the user can perform.
5. Click the Save icon 💾.

Related tasks

- Creating custom roles

Related information

- API Connect user roles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the default Space permissions template

Use the default Space permissions template to pre-configure the Space permissions that are assigned to each role by default when a new Space is created in a Catalog.

About this task

Each Catalog has its own default Space permissions template. The changes that you make to the default permissions affect any new Spaces that are created in the Catalog thereafter. The permissions that are defined in existing Spaces are unaffected. You can change the default Space permissions at any time before creating a new Space.
Note: You cannot change the default permissions for the Space owner.
After you have created a Space, you can change the permission settings for that Space to override the default permissions. For details, see Managing user access in a Space.

Procedure

To configure the default Space permissions template, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. Ensure that the Catalog is selected in the upper-left drop-down menu of the submenu navigation banner.
3. Click Settings > Roles.
4. Select the role for which you want to configure the default Space permissions.
5. Expand the Default Space Permissions section.
6. For each permission in the Default Space Permissions list, select or clear the check boxes to enable or disable the actions that the user can perform.
7. Click the Save icon 💾.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Managing Gateways in a Space

If Spaces are enabled in a Catalog, and the Catalog is configured to use one or more DataPower® Gateway services, you can separately control which of those Gateway services is used by each of the Spaces in the Catalog.

Before you begin

To manage Space Gateway settings in the API Manager UI, a user must be assigned a role that has the Space Settings > Manage permission. For information on adding users to a Space and assigning user roles, see Managing Space membership.

About this task

When you create a Space in a Catalog, the Space is automatically configured to use the same DataPower Gateway services that are used by the Catalog. Therefore, you need to change the Space Gateway settings only in the following situations:

- You want the Space to use a subset of the Gateway services that are used by the containing Catalog.
- You want to restore to the Space Gateway configuration a Gateway service that has previously been removed.

Note: You can use a Gateway service in a Space only if it is used by the containing Catalog.

Procedure

To manage the Gateway settings for a Space, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. In the submenu navigation banner, click the Spaces menu drop-down arrow to select the Space that you want to work with. The Products tab for the Space opens.
3. Click Settings > Gateways.
   The DataPower Gateway services that are used by the Catalog that contains the Space are listed. By default, the Space uses the same Gateway services that are configured for containing Catalog.
4. To change the Gateway service configuration for the Space, select Customize gateway services used by this space, then use the slider controls alongside each Gateway service to select which Gateway services you want the Space to use.

Related concepts

- Using multiple DataPower Gateway services with a Catalog

Related information

- Configuring the initial Gateway service
- Adding more Gateway services

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Administering Developer organizations

Manage the Developer organizations that access your APIs and Plans when their users sign up to use the Developer Portal.

About this task

Users in Developer 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 Developer organization and each organization has one owner. Users are added through the Developer Portal UI.

You can also create a community of Developers. Creating a community is an efficient way of grouping Developers together to publish Products to. This function makes it easier to control who has access to the Plans in the Products, and can be used as a way to map Plan usage.

Use the following tasks to administer your Developer organizations:

- **Creating a Developer organization**
  If you have permission to manage developers, you can create new developer organizations to share in your cloud.
- **Editing a Developer organization**
  You can edit key information that is related to a Developer organization, including display name, owner and community categorization.
- **Deleting a Developer organization**
  When you delete a Developer organization, all of the resources in that organization are removed from the API Connect cloud. However, the account of the organization owner remains in API Connect.
- **Deleting a Developer organization user using a REST API**
  With proper permissions, you can delete user accounts from a Developer organization using a REST API.
- **Exporting developer data**
  You can export data that summarizes the API calling activity for all developer organizations. The exported data contains information about API usage, which is broken down by developer organizations, and can be used to generate billing information.
- **Sending messages to developers**
  You can send email messages to one or more developer organizations.
- **Managing developer applications**
  If an application is behaving suspiciously, exceeding rate limits, or has been compromised, you can block it from accessing your APIs.
- **Managing application subscriptions**
  You can use the API Manager user interface to manage the Plan subscriptions that have been created by application developers in the Developer Portal. You can migrate a subscription to another Plan, and view the developer organization and application associated with the subscription.

Related tasks

- Publishing a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a Developer organization

If you have permission to manage developers, you can create new developer organizations to share in your cloud.

Before you begin

Your IBM Developer Portal must be enabled to perform this task. For more information, see Creating and configuring Catalogs.
V5.0.5 + 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 Developer Organizations and Developers > Manage permission. For more information, see Adding provider organization users and assigning roles and Managing Catalog membership.

About this task

By creating a Developer organization, you can specify an owner for that Developer organization.
V5.0.5 + If Spaces are enabled in your Catalog, when you create a Developer organization, it is added to the Catalog and its Spaces. For more information about enabling Spaces, see Using syndication in IBM API Connect.

Procedure

1. Open the API Manager Dashboard if it is not currently on display:

    a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
    b. To pin the UI navigation pane, click the Pin menu icon to change its state to "pinned" .
    c. Click Dashboard in the navigation pane.
2. Click the Catalog for which you want to create the Developer organization.
3. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab for the Space opens.
4. V5.0.4 + Click Community, then click Developer Organizations.
5. V5.0.4 + Click Add.
6. V5.0.3 and earlier Click the Developers tab , then click Add Organization.
7. In the Developer organization field, enter the developer organization name.
8. Complete one of the following actions to specify the owner in the organization.
    a. To specify that the owner is an existing user, complete the following steps:
        i. Click the Existing User tab in the Specify Owner section.
           Note: If you are using an authentication URL user registry, you do not have the option to select Existing User because it doesn't allow querying for existing users.
        ii. Enter a search string in the Search users field, and click the search icon.
        iii. Select the required email address in the search results list, and click Add. The user is added to the organization, in the Developers section.
    b. To specify a new user for the owner, complete the following steps:
        i. Click New User in the Specify Owner section.
        ii. Enter the required email address, and click Add. The user is added to the organization, in the Developers section.
9. If the user registry is an LDAP directory, complete the following steps to define the owner:
    a. Enter a search string, and click the search icon
    b. Select the required user name in the results list.
    c. Click Add. The user is added to the organization in the Developers section.
   Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.
10. If the user registry is an authentication URL, enter an email address. The user is added to the organization, in the Developers section.

Results

The new Developer organization is added to the list, and an email invitation is sent to the owner. If the user registry is a local user registry or an authentication URL, the status is shown as `Invitation Pending` until the recipient of the email clicks the link in the email to complete the creation of their Developer Portal account. If the user registry is an LDAP directory, the status is shown as `Active`.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Editing a Developer organization

You can edit key information that is related to a Developer organization, including display name, owner and community categorization.

Before you begin

V5.0.5 + 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 Developer Organizations and Developers > Manage permission. For more information, see Adding provider organization users and assigning roles and Managing Catalog membership.

About this task

V5.0.5 + If Spaces are enabled in your Catalog, when you edit the details of a Developer organization, the updates are applied to the Catalog and its Spaces. For more information about enabling Spaces, see Using syndication in IBM API Connect.

Procedure

To edit a Developer organization, complete the following steps:

1. Open the API Manager Dashboard if it is not currently on display:
    a. If you have not previously pinned the UI navigation pane, click the Navigate to icon to open the API Manager UI navigation pane.
    b. To pin the UI navigation pane, click the Pin menu icon to change its state to "pinned".
    c. Click Dashboard in the navigation pane.
2. Click the Catalog that contains the Developer organization that you want to edit.
3. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab for the Space opens.
4. V5.0.4 + Click Community, then click Developer Organizations.
5. V5.0.3 and earlier Click the Developers tab .
6. To change the Developer organization display name, complete the following steps:
   V5.0.4 +
    a. Click the Manage icon for that organization, then click Edit developer organization.
    b. Update the Display Name field, then click Save.

   V5.0.3 and earlier
    a. Click the Developer organization name that you want to edit.
    b. To change the organization display name, enter the new display name in the Developer organization field, then click Save.
       You have changed the display name of your organization.
7. To change the organization owner, complete the following steps:
    a. V5.0.4 + Click the Manage icon for that organization, and then click Change organization owner.
       The Add Owner window is displayed.
    b. V5.0.3 and earlier Click the Developers tab, click the Manage icon for that organization, and then click Change Organization Owner.
       The Add Owner window is displayed.
    c. If the user registry is IBMid, complete the following steps to define the owner:
        i. Enter the full IBMid
        ii. Click Add. The IBMid is added to the Owner column.
    d. If the user registry is a local user registry, complete one of the following actions to define the owner:
        - To specify that the owner is an existing user, complete the following steps:
            - Click Existing User.
            - Enter a search string in the search field, and click the Search icon .
            - Select the required email address in the search results list, and click Add. The email address is added to the Owner column.
        - To specify a new user for the owner, complete the following steps:
            - Click New User.
            - Enter the required email address, and click Add. The email address is added to the Owner column.
    e. If the user registry is an LDAP directory, complete the following steps to define the owner:
        - Enter a search string, and click the Search icon .
        - Select the required user name in the results list, and click Add. The user name is added to the Owner column.

       Important: If you are using IBM API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.
    f. If the user registry is an authentication URL, enter an email address. When the activation email is received, click the activation link, then enter a user name and password.
8. V5.0.4 + To assign a community to a Developer organization, complete the following steps:
    a. Click the Assign Community icon for that organization.
    b. Complete one of the following steps:
        - Enter the name for your community in the New community field, then press the Enter key on your keyboard. Your community is added to the list of communities for the organization.
        - Select an existing community name from the list.
9. V5.0.3 and earlier To assign a community to a Developer organization, complete the following steps:
    a. Click the Assign Community icon adjacent to the organization that you want to work with.
    b. Complete one of the following steps:
        - Enter the name for your community in the Create new community field, then press Enter. Your community is added to the list of communities for the organization.
        - Select an existing community name from the list above the Create new community.

Results

The Developer organization information is edited. If you changed the owner name, an email invitation is sent; if the owner is a new local registry user or authentication URL user, `Pending` is displayed in the Status column. If the new owner is an existing LDAP user, the status is displayed as `Active`. If you assigned or changed the Community, the result is also displayed.

Related tasks

- Publishing a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deleting a Developer organization

When you delete a Developer organization, all of the resources in that organization are removed from the API Connect cloud. However, the account of the organization owner remains in API Connect.

Before you begin

V5.0.5 + 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 Developer Organizations and Developers > Manage permission. For more information, see Adding provider organization users and assigning roles and Managing Catalog membership.

Important: If you delete a Developer organization, client IDs and secrets associated with applications that were registered by users in the Developer 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 Developer organization, it is removed permanently and users can no longer access that organization.
V5.0.5 + If Spaces are enabled in your Catalog, when you delete a Developer organization, it is deleted from the Catalog and its Spaces. For more information about enabling Spaces, see Using syndication in IBM API Connect.

Procedure

To delete a Developer organization, complete the following steps:

1. Open the API Manager Dashboard if it is not currently on display:
   a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
   b. To pin the UI navigation pane, click the Pin menu icon to change its state to "pinned" .
   c. Click Dashboard in the navigation pane.
2. Click the Catalog that contains the Developer organization that you want to delete.
3. Optional: If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab for the Space opens.
4. V5.0.4 + Click Community, then click Developer Organizations.
5. V5.0.4 + Click the Manage icon ⋮, then click Delete developer organization.
6. V5.0.3 and earlier Click the Developers tab , then click the Manage icon adjacent to the Developer organization that you want to delete.
7. V5.0.4 and earlier Click Delete.
   The Confirm Delete window is displayed.
8. Click OK to delete the Developer organization.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deleting a Developer organization user using a REST API

With proper permissions, you can delete user accounts from a Developer organization using a REST API.

About this task

This feature requires the 5085-ifix, available as an Identity Fix from Fix Central (Fix Central > Select Fixes > WebSphere, IBM API Connect (5.0.8.4, All platforms)).

User accounts in a Developer organization can be deleted from IBM® API Connect using a REST API call. (Note that this API cannot be used to delete users from API Manager or Cloud Manager.) A role with one of the following catalog permissions is required to use this API:

- Developer Organizations and Developers
- View and manage developer organizations and developers

Procedure

To delete a user from a Developer organization, issue the following call:

```
DELETE /users/{Id}
```

where

- *{Id}* is the user's id. The user id is required.

Users cannot be deleted if they are currently an owner of a Developer organization. Transfer ownership of the Developer organization prior to deleting the user. Optionally, you can also delete the Developer organization and then delete the users, including the owner.

To find the user id, search using the provider organization environment id. Use the `searchFilter` query parameter with a search string to filter results:

```
curl -X GET 'https://APICmanagementServer.myCompany.com/v1/users/context/4c22194ee2b0cdc2dfd2a2f2?
searchFilter=Alice' -H 'Accept: application/json' -H 'Content-Type: application/json' -u
apimanager/Sally.Manager@myCompany.com:sally_password -k -i
```

where:

- `https://APICmanagementServer.myCompany.com` is the URL of the management server
- `4c22194ee2b0cdc2dfd2a2f2` is the provider organization environment id
- `Alice` is the search string (name of the user)
- `apimanager/Sally.Manager@myCompany.com:sally_password` is the username and password for a user with a role that has the required permissions to use the API, for example, API Manager administrator, product manager, or catalog owner

Following are sample results. The user id appears in bold text:

```
[{"name":"Alice.Smith@myCompany.com","validationLink":null,"id":"4c35144ae4b0cdc4dfd3a324","context":"4c
22194ee2b0cdc2dfd2a2f2","idpId":"4c22194ee2b0cdc2dfd2a2f2","username":"Alice.Smith@myCompany.com","email
":"Alice.Smith@myCompany.com","firstName":"Alice","lastName":"Smith","lastLoginTime":null,"phoneNumber":
null,"status":"active","url":"https://apimdev0068.hursley.ibm.com/v1/users/4c35144ae4b0cdc4dfd3a324"}]
```

You can also find the user id by searching using the provider organization name and environment, for example, `ProvOrgName.sb`:

```
curl -X GET 'https://APICmanagementServer.myCompany.com/v1/users/context/ProvOrgName.sb?
searchFilter=Alice' -H 'Accept: application/json' -H 'Content-Type: application/json' -u
apimanager/Sally.Manager@myCompany.com:sally_password -k -i
```

Following are the sample results for this call. The user id appears in bold text:

```
"name":"myEmail@uk.ibm.com","validationLink":null,"id":"5c877e910cf29fd6e6c6e80b","context":"5c2cce2d0cf
2a6345071d93a","idpId":"5c2cce290cf2a6345071d926","username":"myEmail@uk.ibm.com","email":"myEmail@uk.ib
m.com","firstName":null,"lastName":null,"lastLoginTime":null,"phoneNumber":null,"status":"pending","url"
:"https://158.175.102.115/v1/users/5c877e910cf29fd6e6c6e80b"
```

Note: The Portal Administrator must remove the user from the Portal database.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Exporting developer data

You can export data that summarizes the API calling activity for all developer organizations. The exported data contains information about API usage, which is broken down by developer organizations, and can be used to generate billing information.

The export operation does not export API Connect configuration, and cannot be used as a backup mechanism. For information about backing up or restoring API Connect configuration data, see Preserve your cloud data.

About this task

To obtain data about the API calling activity for all developer organizations, you can configure a visualization by using metric and bucket aggregations to define the type of information to be retrieved. For information about visualizations, see API Analytics.

A sample visualization definition, which retrieves API calling activity for developer organizations, is provided for your use in the following procedure. You can copy and save the sample definition as a .json file, and then import the file into the API Manager UI as a visualization that can be added to your Analytics dashboards. You must import the .json file into the Catalog from which you want to export data. After you add the visualization to a dashboard, you can then export the developer data to a CSV file.

The sample definition describes a data table with columns that display the Product name, Plan name, application name, developer organization name, developer organization ID, API name, URI path on the inbound request, and activity count.

Procedure

To export the data for developer organizations, complete the following steps:

1. Copy and paste the following code definition for a sample visualization into a text editor:

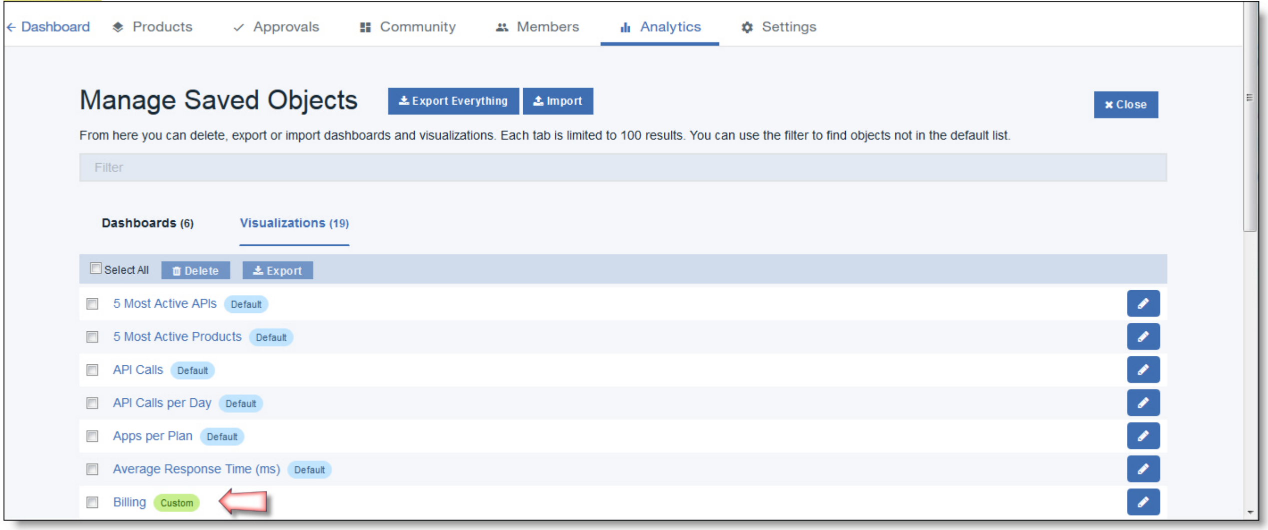
```
[
 {
   "_id": "Billing",
   "_type": "visualization",
   "_source": {
     "title": "Billing",
     "visState": "{\"type\":\"table\",\"params\":
{\"perPage\":25,\"showPartialRows\":false,\"showMeticsAtAllLevels\":false},\"aggs\":
[{\"id\":\"1\",\"type\":\"count\",\"schema\":\"metric\",\"params\":{}},
{\"id\":\"8\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"product_name\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"9\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"plan_name\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"2\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"app_name\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"10\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"developer_org_name\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"11\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"developer_org_id\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"5\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"api_name\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}},
{\"id\":\"6\",\"type\":\"terms\",\"schema\":\"bucket\",\"params\":
{\"field\":\"uri_path\",\"size\":0,\"order\":\"desc\",\"orderBy\":\"1\"}}],\"listeners\":{}}",
     "uiStateJSON": "{}",
     "description": "",
     "savedSearchId": "",
     "version": 1,
     "kibanaSavedObjectMeta": {
       "searchSourceJSON": "{\"filter\":[]}"
     }
   }
 }
]
```

   Note: The title of the sample visualization, as defined in the code definition, is `Billing`. If you already have a visualization with the same title within the provider organization and Catalog into which the .json file will be imported, amend the code to specify a different title that is unique. You can do so by changing the value of the title parameter. To ensure that the visualization ID is also unique for the provider organization and Catalog, you can change the value for the _id parameter; you can use the same value as the title parameter, but must replace any spaces with an underscore (_) or hyphen (-). For example:

```
   "_id": "API_activity_for_developer_organizations",
...
     "title": "API activity for developer organizations",
```

2. Save the text file with a .json extension.
3. From the API Manager UI, access the Analytics tab for the Catalog from which you want to export data. If Spaces are enabled in the Catalog, and you want to export data from a Space, access the Analytics tab for that Space. For more information, see Accessing analytics.
4. Import the .json file for the visualization into the Catalog or Space. For more information, see Importing visualizations.
   The visualization is added to the Visualizations tab on the Manage Saved Objects page, and displays the value of the title parameter that was specified in the .json file. The following image shows an example of an imported visualization (titled **`Billing`**) for the sample code definition in step 1.

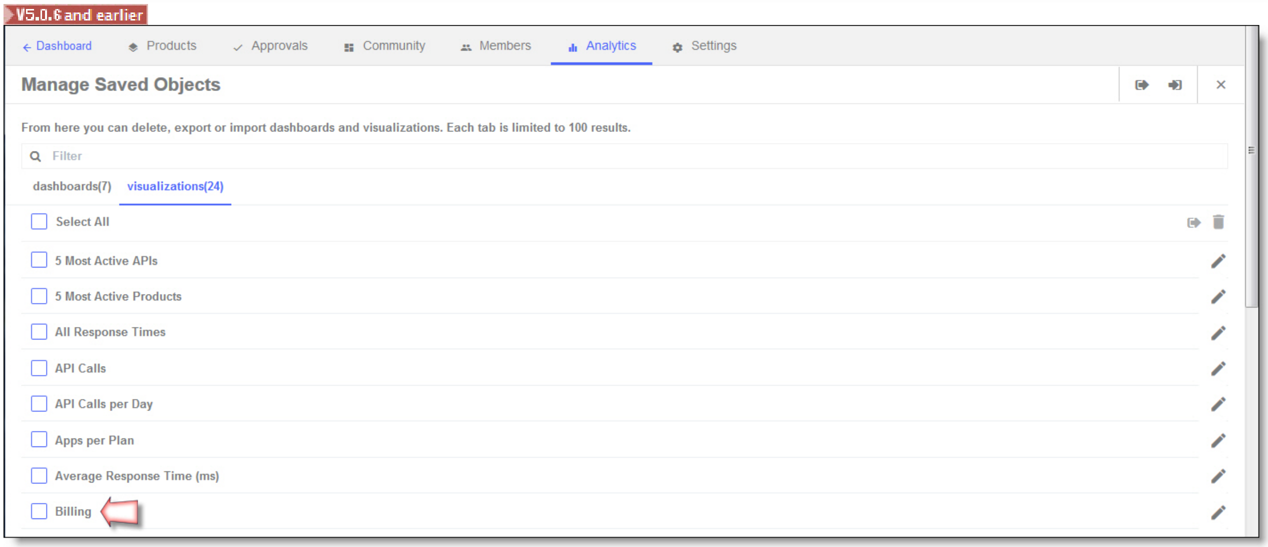
V5.0.7+



From here you can delete, export or import dashboards and visualizations. Each tab is limited to 100 results. You can use the filter to find objects not in the default list.

- □ 5 Most Active APIs Default
- □ 5 Most Active Products Default
- □ API Calls Default
- □ API Calls per Day Default
- □ Apps per Plan Default
- □ Average Response Time (ms) Default
- □ Billing Custom

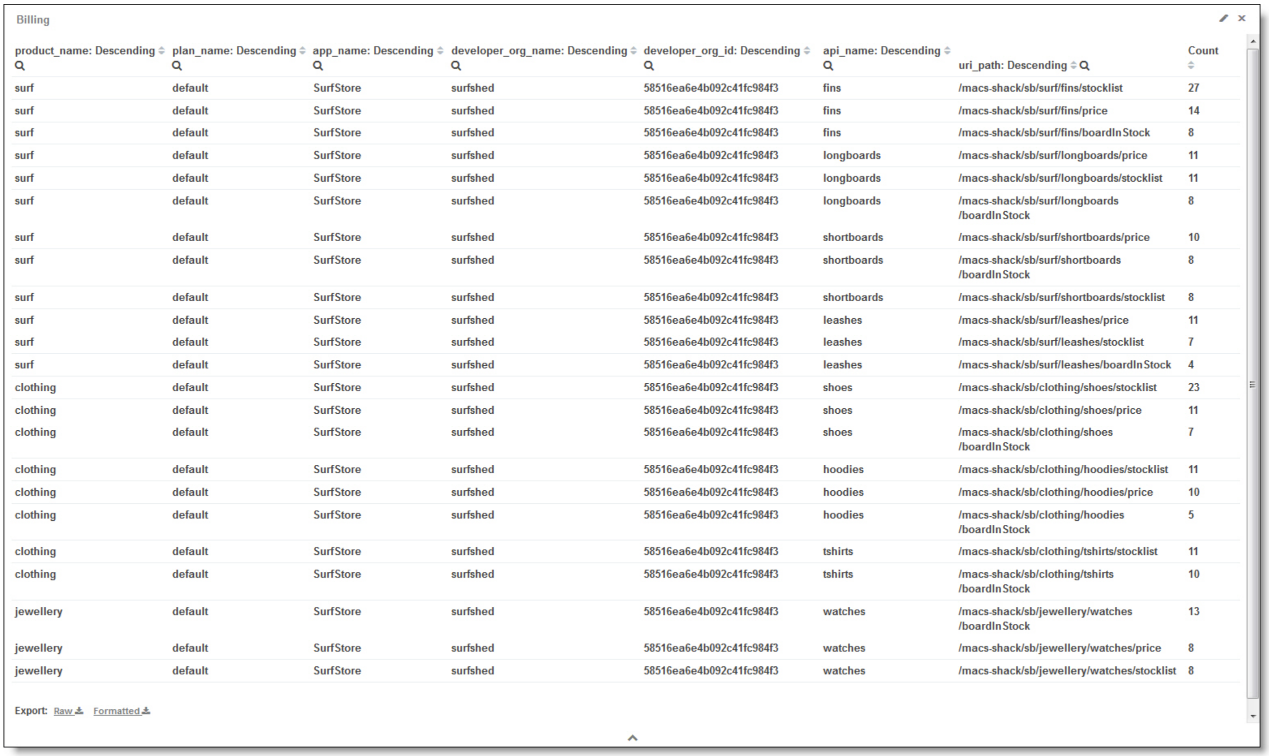
V5.0.6 and earlier



From here you can delete, export or import dashboards and visualizations. Each tab is limited to 100 results.

- □ Select All
- □ 5 Most Active APIs
- □ 5 Most Active Products
- □ All Response Times
- □ API Calls
- □ API Calls per Day
- □ Apps per Plan
- □ Average Response Time (ms)
- □ Billing

5. Add the visualization to a dashboard, as described in Creating custom dashboards or Editing dashboards.
   A data table of the API calling activity for your developer organizations is displayed in the visualization container. The following image shows an example of the visualization with sample data.

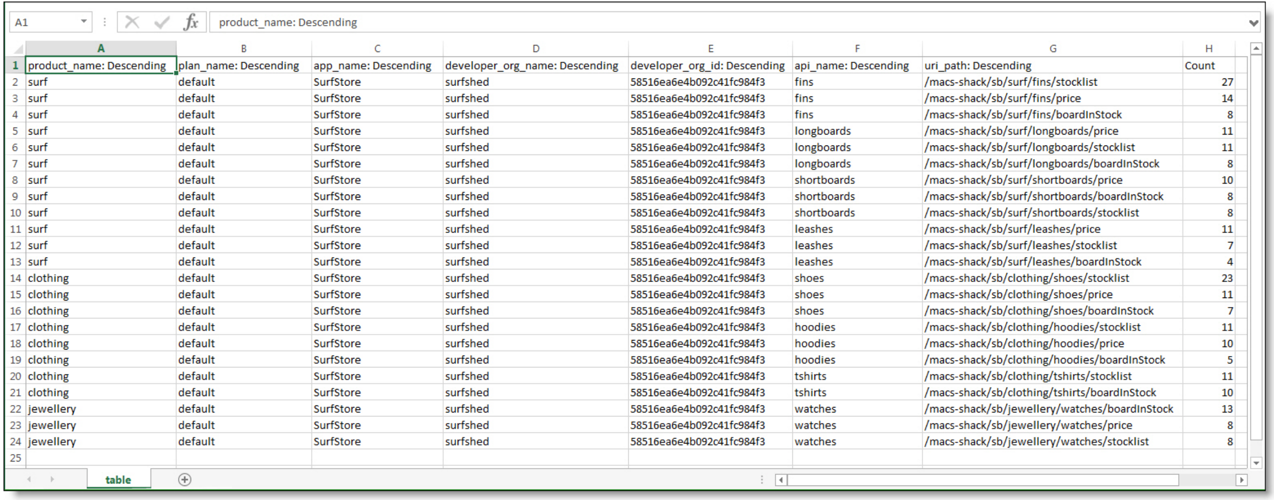


Billing

product_name: Descending	plan_name: Descending	app_name: Descending	developer_org_name: Descending	developer_org_id: Descending	api_name: Descending	uri_path: Descending	Count
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/stocklist	27
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/price	14
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/boardInStock	8
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards/price	11
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards/stocklist	11
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards /boardInStock	8
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards/price	10
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards /boardInStock	8
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards/stocklist	8
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/price	11
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/stocklist	7
surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/boardInStock	4
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes/stocklist	23
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes/price	11
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes /boardInStock	7
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies/stocklist	11
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies/price	10
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies /boardInStock	5
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	tshirts	/macs-shack/sb/clothing/tshirts/stocklist	11
clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	tshirts	/macs-shack/sb/clothing/tshirts /boardInStock	10
jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches /boardInStock	13
jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches/price	8
jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches/stocklist	8

Export: Raw Formatted

6. Export the analytics data from the visualization into a CSV file, as described in Exporting analytics data from a visualization.
   The following image shows an example of the CSV file contents for the sample visualization that is displayed in the previous step.



A1 fx product_name: Descending

	A	B	C	D	E	F	G	H
1	product_name: Descending	plan_name: Descending	app_name: Descending	developer_org_name: Descending	developer_org_id: Descending	api_name: Descending	uri_path: Descending	Count
2	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/stocklist	27
3	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/price	14
4	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	fins	/macs-shack/sb/surf/fins/boardInStock	8
5	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards/price	11
6	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards/stocklist	11
7	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	longboards	/macs-shack/sb/surf/longboards/boardInStock	8
8	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards/price	10
9	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards/boardInStock	8
10	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shortboards	/macs-shack/sb/surf/shortboards/stocklist	8
11	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/price	11
12	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/stocklist	7
13	surf	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	leashes	/macs-shack/sb/surf/leashes/boardInStock	4
14	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes/stocklist	23
15	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes/price	11
16	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	shoes	/macs-shack/sb/clothing/shoes/boardInStock	7
17	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies/stocklist	11
18	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies/price	10
19	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	hoodies	/macs-shack/sb/clothing/hoodies/boardInStock	5
20	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	tshirts	/macs-shack/sb/clothing/tshirts/stocklist	11
21	clothing	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	tshirts	/macs-shack/sb/clothing/tshirts/boardInStock	10
22	jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches/boardInStock	13
23	jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches/price	8
24	jewellery	default	SurfStore	surfshed	58516ea6e4b092c41fc984f3	watches	/macs-shack/sb/jewellery/watches/stocklist	8
25								

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Sending messages to developers

You can send email messages to one or more developer organizations.

Before you begin

V5.0.5 + To complete this task, you must be assigned a role that has the Developer Organizations and Developers > Manage permission. For more information, see Adding provider organization users and assigning roles and Managing Catalog membership.

About this task

The recipient of the message is the owner of the Developer organization.

Procedure

To send a message, complete the following steps:

1. Open the API Manager Dashboard if it is not currently on display:
    a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
    b. To pin the UI navigation pane, click the Pin menu icon 📌 to change its state to "pinned" 📌.
    c. Click Dashboard in the navigation pane.
2. Click the Catalog that contains the organization that you want to send a message to.
3. Optional: V5.0.5 + If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   For more information about enabling Spaces, see Using syndication in IBM API Connect.
4. V5.0.4 + Click Community, then click Developer Organizations.
5. V5.0.4 + Click the Manage icon.
6. V5.0.3 and earlier Click the Developers tab 👥, then click the Manage icon ⌄ alongside the required Developer organization.
7. Click Send message.
8. Enter the subject of the message in the Subject field.
9. Enter the content of the message in the Message field, then click Send.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing developer applications

If an application is behaving suspiciously, exceeding rate limits, or has been compromised, you can block it from accessing your APIs.

Before you begin

V5.0.5 + 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 suspend an application for some 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 bill for API usage and the developer is in arrears or has not paid for previous usage.
- You become aware that the client secret has been compromised and that the developer has not reset it.

V5.0.8 + You can suspend an application temporarily when there is a billing issue with the credit card service provider. An example of a billing issue is an expired credit card. When the issue is resolved, you can resume the application.

Note: If a suspended application calls an API, the API gateway returns error code `403`.

Procedure

To block an application from accessing your APIs, complete the following steps.

1. Open the API Manager Dashboard if it is not currently on display:
    a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
    b. To pin the UI navigation pane, click the Pin menu icon 📌 to change its state to "pinned" 📌.
    c. Click Dashboard in the navigation pane.

2. Click the Catalog that contains the application you want to work with.
3. Optional: V5.0.5+ If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   For more information about enabling Spaces, see Using syndication in IBM API Connect.
4. V5.0.4+ Click Community, then click Applications.
   All applications that have been registered in the Developer Portal associated with this Catalog are listed.
   V5.0.7+ If Development Application Workflow is enabled for the Catalog, an additional TYPE column shows whether each application is in Development or Production status; for more information, see Managing the application lifecycle.

5. V5.0.4+ Click the Manage icon ⋮ for the application that you want to suspend, then click Suspend application.
6. V5.0.4+ Click OK.
   V5.0.8+ Remember: The payment processing continues while the application is suspended when the suspended application uses APIs that have billing subscriptions. Customers must unsubscribe to the Plan in their IBM API Connect account to stop the payment processing for that subscription.
   V5.0.5+ If Spaces are enabled in the Catalog, the application is suspended in the Catalog and its Spaces.

7. Optional: V5.0.4+ To unsuspend the application, click the Manage icon ⋮ for the application that you want to unsuspend, then click Resume application.
   V5.0.5+ If Spaces are enabled in the Catalog, the application is reactivated in the Catalog and its Spaces.

8. V5.0.3 and earlier Click the Developers tab 👥, then click the name of the Developer organization that you want to work with.
9. V5.0.3 and earlier In the side pane, select the application that you want to suspend.
10. V5.0.3 and earlier Click Suspend.
    The application is suspended and the client ID is unable to make further API calls.
11. Optional: V5.0.3 and earlier To unsuspend the application, click Resume.

Results

When you suspend or resume an application, the application developer is notified by email and in their Developer Portal activity feed.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing application subscriptions

You can use the API Manager user interface to manage the Plan subscriptions that have been created by application developers in the Developer Portal. You can migrate a subscription to another Plan, and view the developer organization and application associated with the subscription.

Before you begin

V5.0.5+ To complete this task, you must be assigned a role that has the Subscriptions > Manage permission. For more information, see Adding provider organization users and assigning roles and Managing Catalog membership.

Open the list of application subscriptions by completing the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
2. V5.0.5+ If Spaces are enabled in the Catalog, select the Space that you want to work with by clicking the (Catalog) drop-down menu. For more information about enabling Spaces, see Using syndication in IBM API Connect.
3. Click Community > Subscriptions.

Procedure

- To migrate a subscription to another Plan, complete the following steps:
  1. Click the Manage icon ⋮ alongside the subscription that you want to migrate.
  2. V5.0.7+ Click Manage.
     V5.0.6 and earlier Click Migrate subscription.
     The "Migrate subscription to another plan" window opens. Plans are listed, grouped by containing Product.
  3. Select the Plan that you want to migrate the subscription to, then click Migrate.

The subscription is migrated and the Product column for the subscriptions updates to show the selected Plan.

- To show the developer organization that created the subscription, or the developer application associated with the subscription, complete the following steps:
  1. Click the Manage icon ⋮ alongside the required subscription.
  2. Click Show developer organization or Show application as required.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with Apps

An App is the publish target for an application. An App references an API Connect collective. When you publish an application by using the API Designer or the CLI **apic publish** command, the application is deployed to the collective referenced by the App.

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

You create an App from the dashboard in the API Manager user interface. You can monitor the collective referenced by your App by using the Liberty Admin Center.

For details on working with Apps, see the following subtopics:

- **Creating an App**
  You create an App from the dashboard in the API Manager user interface. When you create an App, you specify the API Connect collective that is the deployment target for an application that is published to the App.
- **Monitoring the Liberty collective associated with an App**
  To monitor the Liberty collective associated with an App, you use the Liberty Admin Center. For example, you can see the details of the applications that have been published to the App.
- **Removing an application from a collective**
  You can remove an application from a collective through the CLI. To delete the application, you need the application ID and the names of the server or servers that are associated with the application.
- **Obtaining the publish target URL for an App**
  Every App has a unique URL containing all the information required to publish an application to the App from the developer toolkit command line.

Related information

- Publishing a project from the command line
- Staging and publishing a project from the API Designer
- Installing API Connect Collective

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating an App

You create an App from the dashboard in the API Manager user interface. When you create an App, you specify the API Connect collective that is the deployment target for an application that is published to the App.

Before you begin

Install an API Connect collective on an appliance in your API Connect cloud. For details, see Installing an API Connect collective

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

Procedure

To create an App, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. To add an App:
   a. On the Dashboard page, click Add > App.
      Note: The Add > App option is not available until you have installed an API Connect collective.
   b. Enter the Display Name, and the Name that identifies the App.
   c. Select the collective to be associated with the App. An application that is published to this App will be deployed to the specified collective.
   d. Click Add.
      The new App is added to the dashboard.

Related information

- Publishing a project from the command line
- Staging and publishing a project from the API Designer
- Installing API Connect collective

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Monitoring the Liberty collective associated with an App

To monitor the Liberty collective associated with an App, you use the Liberty Admin Center. For example, you can see the details of the applications that have been published to the App.

Before you begin

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

Procedure

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. On the Dashboard page, click the App that you want to manage
3. Click Manage this app.
   The Liberty Admin Center login page opens.
4. Log in to the Liberty Admin Center, then click Explore. You can now monitor the Liberty collective associated with the App.
   Note: Your access is **read-only**.

What to do next

For details on monitoring a Liberty collective, see Exploring and managing resources with Admin Center in the WebSphere® Application Server product documentation.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing an application from a collective

You can remove an application from a collective through the CLI. To delete the application, you need the application ID and the names of the server or servers that are associated with the application.

Before you begin

Important:

- IBM® API Connect collectives are deprecated in IBM API Connect Version 5.0.7 in favor of container runtimes. For more information and background, see Open, scalable, flexible runtime management of APIs through API Connect enabled containers. For information on setting up and migrating to containers, see Installing a containerized runtime environment.
- Existing customers can continue to use their collectives with IBM API Connect Version 5.0.7, and if wanted can expand their collective deployments to new servers. API Connect collectives are supported for existing customers until the end of support of IBM API Connect Version 5.0 (see Software lifecycle page for IBM API Connect Version 5.0). Until then, users of API Connect collectives are encouraged to migrate to container runtimes to take advantage of their agility and scalability.
- New customers should not install API Connect collectives because this feature is no longer supported for new users.

Procedure

The application ID is used to locate files that you must delete to delete the application. Find the application ID and server name that is associated with each application by completing the following steps:

1. Log in to the Liberty Admin Center for your collective.
2. Click Explore > Applications.
3. Select the application that you want to delete to see the application ID displayed in the following format:

   ```
   apic.unique_app_id
   ```

4. Click the Dashboard icon, then click Servers.
5. Select the server that is associated with your application and the server name is displayed in the following format:

   ```
   unique_app_id -n
   ```

   where `n` is the server number.

Delete servers and associated files for each application in the collective by completing the following steps.

6. Note: Perform this step from the Controller through an SSH session.
   Delete each server that is associated with the application.
   a. Edit the corresponding ~/.liberty/wlp/usr/servers/controller/configDropins/overrides/application_id.scalingPolicy.xml and put `false` for `enabled` value inside the `scalingPolicy` tag and save.
   b. Remove ~/.liberty/wlp/usr/shared/stackGroups/apic/packages/*application_id*-package.tgz and ~/.liberty/wlp/usr/shared/stackGroups/apic/packages/*application_id*.deploy.xml files.
   c. Click each server in the application's cluster to confirm that each displays `Auto scaling policy is disabled`.

   d. Run **`wlpn-server stop <member>-<instance#>`** for the instances of the application.

7. Note: Perform this step from each member server.
   For each member that is associated with the application, remove the server from the collective by running the following command as a single entry.

   ```
   wlpn-collective remove --user=user --password=password --host=controller_host_URL --port=9443 --
   autoAcceptCertificates --hostName=registered_memberHost_name
   ```

   Where **--host** is the controller host URL, **--hostName** is the **`memberHost`** (hostName) that you provided to register the host, *username* and *password* are the username and password for the controller that is associated with the collective.
   Note: If you are removing a server remotely, both the controller host and the server host must have the wlpn folder installed in the same directory.
8. Note: Perform this step from each member server.
   Change to the ~/wlpn folder, then locate and delete the server instance folder for each server that is associated with the application. Server instance folders follow the pattern ~/wlpn/<*app_id*>-<*instance#*>.
9. Note: Perform this step from the controller.
   Remove ~/.liberty/wlp/usr/servers/controller/configDropins/overrides/*application_id*.scalingPolicy.xml.

Results

Your application is removed from the CLI and from the Liberty Admin Center.

What to do next

Repeat steps 1 and 2 to confirm that the application you deleted does not appear.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the publish target URL for an App

Every App has a unique URL containing all the information required to publish an application to the App from the developer toolkit command line.

About this task

When you publish an application from the developer toolkit command line, you can create an **`app`** configuration property that defines a URL that contains the target server, provider organization, and Catalog, so that you do not have to supply these values as command options. You can use the API Manager user interface to obtain the required command that sets the publish target URL for a specific Catalog.

For information on publishing an API from the developer toolkit command line, see Creating and publishing an API definition from the command line.

Procedure

To obtain the command that sets the publish target URL for an App, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. On the Dashboard page, click the Show catalog identifier icon 🔗 for the required App.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating APIs

An API is a set of functions that provides some business or technical capability and can be called by applications by using a defined protocol. In the context of API Manager, applications are typically mobile or web applications, and they use the HTTP protocol.

About this task

For information about creating API definitions, see the Creating API definitions by using the API Designer and Configuring API security by using the API Designer documentation.

SOAP API definitions can be created only through API Manager or the command line interface.

For information about creating SOAP API definitions through API Manager or for information about using OpenAPI (Swagger 2.0) definitions in API Manager, see the following topics:

- **Creating and managing OpenAPI (Swagger 2.0) definitions in API Manager**
  In API Manager you can import, download, and edit OpenAPI (Swagger 2.0) definition files of REST APIs. These files contain all information for the API that they describe and a modified version of the definition is available to application developers through the Developer Portal regardless of the method through which you created your API.
- API Manager only **Adding a SOAP API definition by discovering a service from a registry**
  You can add a SOAP API by finding the details for an existing SOAP service in a registry.
- **Updating a SOAP API**
  You can update the configuration of an existing SOAP API either by uploading a WSDL file, or by uploading a .zip archive that contains a primary WSDL file together with other WSDL or XSD files that it references.
- V5.0.1+ **Adding an existing web service to your API definition**
  Existing web services can be added to an API and used in the assembly with generated map and invoke policies.
- **Creating a new version of an API definition**
- V5.0.4+ **Adding an API definition to existing Products**
- **Testing an API with the API Manager test tool**
  You can test the API from within API Manager to ensure that is defined and implemented correctly.
- **Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Manager UI)**
  When you configure a REST or SOAP API definition, you can extend the OpenAPI (Swagger 2.0) specification by adding either a JSON or YAML extension schema depending on the version of IBM® API Connect you are using. V5.0.2+ You can also replace an extension with an updated version.
- V5.0.6+ **Organizing your APIs and Products into categories**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating and managing OpenAPI (Swagger 2.0) definitions in API Manager

In API Manager you can import, download, and edit OpenAPI (Swagger 2.0) definition files of REST APIs. These files contain all information for the API that they describe and a modified version of the definition is available to application developers through the Developer Portal regardless of the method through which you created your API.

Before you begin

To complete the API management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned a role that has permission to edit draft API definitions. If you have permission only to view draft API definitions, you cannot export an OpenAPI (Swagger 2.0) definition file. For information on managing user roles, see Administering user access.

- **Adding a REST API by using an OpenAPI (Swagger 2.0) file**
  You can use an OpenAPI (Swagger 2.0) definition file to add a REST API.
- **Updating a REST API from an OpenAPI (Swagger 2.0) definition file**
  You can update the configuration of a REST API by uploading an OpenAPI (Swagger 2.0) definition file.

- **Exporting an OpenAPI (Swagger 2.0) definition file for a REST API**
  After you create a REST API definition, you can export a file that contains the OpenAPI (Swagger 2.0) definition for the API.

Related information

- Creating a provider organization account

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a REST API by using an OpenAPI (Swagger 2.0) file

You can use an OpenAPI (Swagger 2.0) definition file to add a REST API.

Before you begin

Your file must conform to version 2.0 of the OpenAPI (Swagger 2.0) specification. The format of the file can be JSON or YAML.

Procedure

To add a REST API by loading an OpenAPI (Swagger 2.0) file, complete the following steps:

1. Click APIs.
   The APIs tab opens.
2. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
4. Click Add and then select V5.0.0 ONLY Swagger 2.0, V5.0.1+ OpenAPI (Swagger 2.0), or V5.0.4 + Import an existing OpenAPI from the Import section.
   The Import OpenAPI (Swagger) window opens.
5. Optional: To upload a file from your local file system, click Select file and, in your file system, select the file that you want to use.
   The following file types are supported if they contain a valid OpenAPI (Swagger 2.0) definition: .json, .yml, and .yaml.
6. Optional: To upload a file from a URL, click Or import from URL and then provide the correct URL in the URL field that is presented. If authentication is required to access the URL, provide a user name and password.
   The following file types are supported if they contain a valid OpenAPI (Swagger 2.0) definition: .json, .yml, and .yaml.
7. V5.0.4 + To create a new Product and include your API in that Product, complete the following steps. (If you want to create your API without adding it to a Product, proceed to step 8.)
   a. Select Add a product.
   b. API Designer only In the Product template field, select Default if you want to use the template defined as the default, to create the Product definition. This can either be the default .hbs template file provided with the developer toolkit, or another template file that you have configured as the default by using configuration variables. You can also select a custom template that you created. For information about template files and configuration variables, see Creating and using API and Product definitions templates and Toolkit command summary.
   c. Specify values for the Product title, name, and version.
   d. To publish the Product to a target Catalog, ensure that the Publish this product to a catalog check box is selected. You can clear this check box and stage or publish the Product later by using the API Designer UI and API Manager UI, as described in Staging a Product and Publishing a Product.
   e. V5.0.4 and earlier Select the Catalog that you want to use.
   f. V5.0.5 + If Spaces have been enabled, select the Catalog and Space that you require.Your Product is staged to the Space that you selected.
8. Click Import.
   A new REST API definition is created, including Paths and HTTP operations.

Results

When the API definition has been imported, it is shown in the list of API definitions in the APIs tab of the Drafts page.

What to do next

You can edit your API definition as you would any other REST API definition. For more information, see Composing a REST API definition.

To finish the creation of your API definition, complete the following tasks.

- Configure security for your API. For more information, see Configuring API security
- Enable other users to add and define API definitions. For more information, see Administering user access.

Related information

- IBM API Connect overview
- Managing your V5 APIs
- OpenAPI (Swagger 2.0) Specification
- What is OpenAPI (Swagger 2.0)?
- API and Product definition template examples

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Updating a REST API from an OpenAPI (Swagger 2.0) definition file

You can update the configuration of a REST API by uploading an OpenAPI (Swagger 2.0) definition file.

Before you begin

Important: When you upload an OpenAPI (Swagger 2.0) definition file, you **overwrite** the current configuration of the API definition. If you want to retain the current configuration so that you can roll back the changes if necessary, create a new version before uploading the definition file; for information, see Creating a new version of an API definition.
If you use the API Designer user interface to work with your API definitions, rather than the API Manager, you will be working directly with local OpenAPI (Swagger 2.0) definition files, rather than uploading them; for more information, see Creating API definitions by using the API Designer.

Procedure

To update a REST API from an OpenAPI (Swagger 2.0) definition file, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the REST API definition that you want to work with.
4. Click the More Actions icon ⋮ and then click Update.
5. Click Choose File or Browse depending on which is displayed and, in your file system, select the OpenAPI (Swagger 2.0) definition you want to use.
6. Click Update.
   The original REST API configuration is replaced with the REST API configuration defined in the OpenAPI (Swagger 2.0) file.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Exporting an OpenAPI (Swagger 2.0) definition file for a REST API

After you create a REST API definition, you can export a file that contains the OpenAPI (Swagger 2.0) definition for the API.

About this task

The file that is exported is in YAML format.

Procedure

To export an OpenAPI (Swagger 2.0) definition file for a REST API, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the REST API definition that you want to work with.
4. Click the More Actions icon ⋮ and then click Download.

Results

A file that contains the OpenAPI (Swagger 2.0) definition for your API is downloaded.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Manager only

Adding a SOAP API definition by discovering a service from a registry

You can add a SOAP API by finding the details for an existing SOAP service in a registry.

Before you begin

Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM® Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.
To complete the API management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned a role that has permission to edit draft APIs. If you have permission only to view draft APIs, you cannot create, edit, or delete APIs. For information on managing user roles, see Administering user access.

Procedure

To add a SOAP API by finding the details for an existing SOAP service in a registry, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. V5.0.4 + Click Add > API from SOAP service.
   V5.0.3 and earlier Click Add > API from WSDL.
   The "New API from WSDL" window opens.
4. If the registry has been previously defined, select it from the registry list.
   If no registries are defined, click Find in registry and define a registry; for details on defining a registry, see Adding a WSRR registry for a SOAP API and Adding a custom registry for a SOAP API.
5. Optional: You can delete the registry by clicking the Delete registry icon The Delete registry icon and then clicking OK.

6. To search for a service, enter the name, or part of the name, of the service in the Search field.
   If the service registry contains many services, the query can take a significant amount of time to return. To reduce the query time, enter an appropriate search string.
7. Select the service, or services, for which you want to create a SOAP API, click Next.
   When the operation is complete, the SOAP API definition is shown in the APIs tab of the Drafts page.

Results

The SOAP API is displayed.

What to do next

- Configure the details of the SOAP API, see Configuring a SOAP API definition.
- Configure security for your API. For more information, see Configuring API security

- **Adding a WSRR registry for a SOAP API**
  If you govern your existing SOAP services in WebSphere Service Registry and Repository (WSRR), add a service registry of this type. By adding a registry to API Connect, you can search the registry for SOAP services that you want to expose through API Connect as a SOAP API.
- **Adding a custom registry for a SOAP API**
  Custom registry support is deprecated and will not be available after the IBM API Connect Version 5.0 release. To integrate IBM API Connect with your own service registry, or a third party service registry, consider implementing a solution that uses the API Connect developer toolkit command-line tool to push service definitions to API Connect.

Related information

- IBM API Connect overview
- Managing your V5 APIs
- Adding a SOAP API definition by using a WSDL file

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Manager only

Adding a WSRR registry for a SOAP API

If you govern your existing SOAP services in WebSphere® Service Registry and Repository (WSRR), add a service registry of this type. By adding a registry to API Connect, you can search the registry for SOAP services that you want to expose through API Connect as a SOAP API.

Before you begin

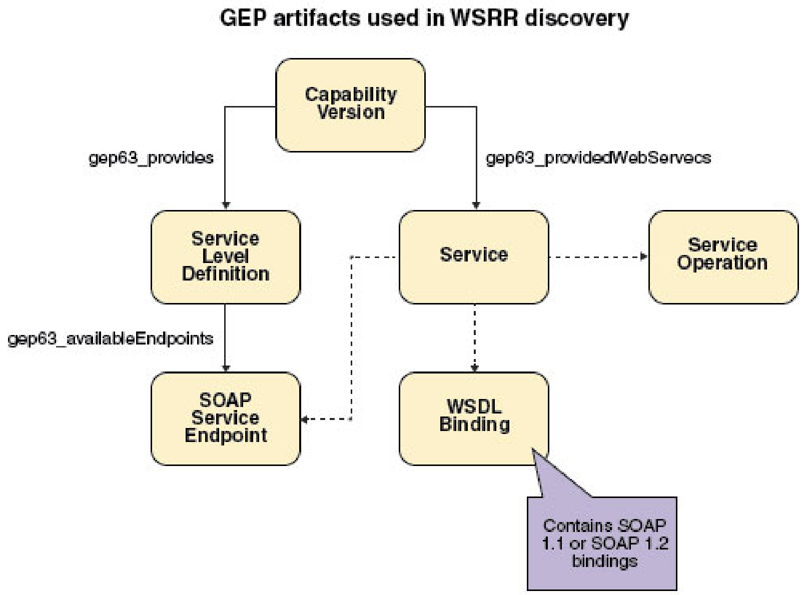
If you have customized your data model in WSRR so that it does not match the diagram that follows in this topic, you must add a query to WSRR so that IBM API Connect can discover the SOAP service in your registry. If only the lifecycles or classifications have been modified then this step is not required. For more information about queries, see Named queries for discovering a service in WSRR.
Important: If you are using API Connect in the cloud, then the services that you expose *must* be visible on the Internet, they must not be accessible only from within your corporate intranet. However, you can use TLS profiles configured in API Manager to protect the communication channel between the API Gateway in API Connect and the services that you expose on the Internet through your DMZ. You can also use the IBM Cloud Secure Gateway service to provide secure access to your on-premises services from IBM API Connect for IBM Cloud; for more information, see About Secure Gateway.
To complete the API management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned a role that has permission to edit draft APIs. If you have permission only to view draft APIs, you cannot create, edit, or delete APIs. For information on managing user roles, see Administering user access.

About this task

If you govern your existing SOAP services in WebSphere Service Registry and Repository (WSRR), you can add those services to API Connect by using the API Connect user interface to discover those services in the registry.

In WSRR, it is best practice not to put multiple web services under a service version. Therefore, when you are discovering from WSRR, API Connect expects that the WSDL file that is associated with a service version contains only one SOAP service definition.

When you use the objects that are modeled in the Governance Enablement Profile in WSRR, a *Service* refers to a Capability Version object that is related to a web service (by the provided web service relationship) which has either SOAP 1.1 or SOAP 1.2 bindings. The endpoints that are returned from this service are the available endpoints that are exposed on the Capability Version by a Service Level Definition (SLD) object.



The details of the queries that are used can be found in the Named queries for discovering a service in WSRR topic.

Procedure

To add a WSRR registry, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. V5.0.4 + Click Add > New OpenAPI from SOAP service.
   V5.0.3 and earlier Click Add > API from WSDL.
   The "New API from WSDL" window opens.
4. Click Find in registry.
   The panel displays text fields for the new registry.
5. Provide a Name for the registry, to be used in API Manager.
6. From the Service Registry Type drop-down menu, select IBM WebSphere Service Registry and Repository.
7. Complete the following mandatory fields:
   - Protocol. Select the protocol that you want to use for your registry.
   - Hostname. Enter the host name.
   - Port. Enter the port number.
8. If your instance is configured with security enabled, in the Username and Password fields enter the user name and password that the system uses to log in to the registry.
9. If multiple instances of WSRR are installed in the same WebSphere Application Server cell, complete the Context Root Prefix field. If only one WSRR instance is installed in your cell, leave this field blank.
   Your WSRR administrator can provide the appropriate context root prefix so that you can attach your SOAP API to the correct WSRR instance.
   Note: The *context root prefix* is also referred to as the *instance prefix* when you are installing WSRR.
   For more information about the instance prefix in WSRR, see the IBM WebSphere Service Registry and Repository Version 8.0 documentation (https://www-01.ibm.com/support/knowledgecenter/SSWLGF_8.0.0/maps/product_landing.html), and search for *Multiple WSRR instances and URLs* in WebSphere Service Registry and Repository (WSRR).

   For more information about the context root prefix in WebSphere Service Registry and Repository Studio, see the IBM WebSphere Service Registry and Repository Version 8.0 documentation in the IBM Knowledge Center at (https://www-

01.ibm.com/support/knowledgecenter/SSWLGF_8.0.0/maps/product_landing.html), and search for *Adding general connection location information* in WebSphere Service Registry and Repository Studio.

10. Click Next.
11. Optional: You can choose to define one or more conditions that are used to pre-filter the set of results that are returned from WSRR Then, when you search the registry for services, results are returned only from the pre-filtered set.
    To pre-filter content, you select one or more WSRR classification values; when you search the registry, only services that are classified with the specified values are included in the search results.

    For convenience, governance state and environment classifications are provided separately for you to select from, but you can select any of the classification values that are defined in the WSRR taxonomy. For example, you can specify that only services that are in the Account Administration business domain, in the Production environment, and in the Managed governance state are included in the pre-filtered set.

    You can select classification filters for service versions and for service endpoints. When you search the registry, you are searching for service versions; however, the service versions that are returned include only those service endpoints that match your service endpoint classification filters.

    Note: If you select two or more classifications, a service version or service endpoint must have **all** the selected classifications to be included in the pre-filtered set.
    To define your conditions, complete the following steps:

    a. Click the Show or hide filters icon .
    b. To add service version and service endpoint filters, click + filter for the type of filter and select the required classification values.
    c. Optional: To remove a filter, click the cross on the filter.
    d. Click Apply to filter the results.
12. Select any of the SOAP services for which you want to create API definitions and then click Next.
13. Optional: To create a Product and include your new API or APIs in the Product, select Create product and provide a Name for the Product.
14. Click Done.

Results

The new registry is added to the system and displayed in the list for you to select when you search for services.

What to do next

You can use the registry to add a SOAP API, see Adding a SOAP API definition by discovering a service from a registry.

Related reference

- Named queries for discovering a service in WSRR

Related information

- IBM API Connect overview
- Managing your V5 APIs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Manager only

Adding a custom registry for a SOAP API

Custom registry support is deprecated and will not be available after the IBM® API Connect Version 5.0 release. To integrate IBM API Connect with your own service registry, or a third party service registry, consider implementing a solution that uses the API Connect developer toolkit command-line tool to push service definitions to API Connect.

Procedure

To add a custom registry, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. V5.0.4 + Click Add > New OpenAPI from SOAP service.
   V5.0.3 and earlier Click Add > API from WSDL.
   The "New API from WSDL" window opens.
4. Click Find in registry.
5. Provide a Name for the registry, to be used in API Manager.
6. From the Service Registry Type drop-down menu, select Custom Registry.
7. Complete the following mandatory fields:
   - Protocol. Enter the protocol that you want to use for your registry.
   - Hostname. Enter the host name.
   - Port. Enter the port number.
   - Context Root. The context root must be set to the text that forms the first part of the resource path in any URL The correct text is determined with reference to the installation properties of the target service registry instance.
8. If your service registry instance is configured with security enabled, in the Username and Password fields enter the user name and password that the system uses to log in to the registry.
9. Click Next.
10. Select the SOAP services that you want to create API definitions for.
11. Click Done.

Results

Your SOAP services are made available through SOAP API definitions and your registry is added to API Manager for further use.

What to do next

You can use the custom registry to add a SOAP API, see Adding a SOAP API definition by discovering a service from a registry.

Related information

- IBM API Connect overview
- Managing your V5 APIs
- Getting started with the developer toolkit command-line tool

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Updating a SOAP API

You can update the configuration of an existing SOAP API either by uploading a WSDL file, or by uploading a .zip archive that contains a primary WSDL file together with other WSDL or XSD files that it references.

About this task

When you initiate the update action, the specified WSDL is parsed to find a service that matches the name of the API that you are updating. If a match is found, the WSDL service is converted into a YAML file that is used to update the API. If no match is found, the update action terminates.

The behavior of the SOAP API update action depends on the version of IBM® API Connect that you are using:

- V5.0.8 + Only those sections of the API that are affected by the new WSDL are replaced, the other sections are unchanged.

- V5.0.7 and earlier The update action completely **overwrites** the current configuration of the SOAP API definition, including all design properties and assembly configuration. If you want to retain the current configuration so that you can roll back the changes if necessary, create a new version before uploading the definition file; for information, see Creating a new version of an API definition

Procedure

To update a SOAP API from a WSDL file or a .zip archive, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the SOAP API definition that you want to work with.
4. Click the More Actions icon ⋮ and then click Update.
5. V5.0.7+ Click Upload file.
   V5.0.6 and earlier Click Choose File or Browse depending on which is displayed for your browser.
6. In your file system, select the WSDL file or a .zip archive that you want to use.
7. Click Update.
   The API is updated from the specified WSDL definition.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Adding an existing web service to your API definition

Existing web services can be added to an API and used in the assembly with generated map and invoke policies.

Before you begin

To complete the API management tasks that are described in this topic, you must either be the owner of the API provider organization, or be assigned a role that has permission to edit draft APIs. If you have permission only to view draft APIs, you cannot create, edit, or delete APIs. For information on managing user roles, see Administering user access.

Procedure

To invoke a web service invocation by using a WSDL definition and mapping its inputs and outputs to other variables, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the API that you want to invoke a web service, or create a new API. For more information of creating REST APIs, see Adding a REST API definition.
4. In the Services section, click the Add service icon ⊕.
5. You can include the service information by using a WSDL file, a .zip file that contains a WSDL file and the associated schemas, or from a registry
   - To upload your WSDL or .zip file, click Upload WSDL and then browse for your WSDL file.
   - To load your WSDL file from a URL, click Load from URL and then enter the URL and, if required, the user name and password required to access the URL. Click Next.
   - To load your WSDL file from a service registry, click Find in registry and then select the registry that you want to use and then click Next. For more information on adding registries, see Adding a WSRR registry for a SOAP APIAdding a custom registry for a SOAP API.

   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.

6. Select the web services that you want to add.
7. Click Done.
8. Click the Assemble tab.
   The assemble view opens.
9. From the Web service operations section of the palette, drag your web service to where you want to include it in your assembly.
   Two map policies surrounding a single invoke policy are created.
10. Configure the inputs of your web service using the "input" map policy. For more information on using the map policy, see Configuring the map policy in the user interface.
11. Configure the outputs of your web service by using the "output" map policy. For more information on using the map policy, see Configuring the map policy in the user interface.

Results

You have configured your API to invoke an existing web service.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a new version of an API definition

You can create multiple versions of an API definition and edit the versions independently.

Procedure

To create a new version of an API definition, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to create a new version of.
   The details of your API definition are displayed.
5. Click the More Actions icon ⋮ and then click Save as a new version.
   The "Save as a new version" window opens.
6. In the Version field, enter your new version number.
   Note: The version corresponds to the `info.version` property value of the API's OpenAPI (Swagger 2.0) definition. The `version.release.modification` version numbering scheme is recommended, for example `1.0.0`.
7. API Designer only V5.0.4 + Accept or change the default file name as required.
8. Click Save as a new version.

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 in the APIs tab of the Compose page of the API Designer.

Related tasks

- Adding a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.4 +

Adding an API definition to existing Products

If you created an API definition without adding it to a Product, or if you want to add an existing API definition to additional Products, you can do so while viewing the API details.

Procedure

To add an API definition to an existing Product, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click APIs.
   The APIs tab opens.
4. Click the API definition that you want to add to a Product.
   The details of your API definition are displayed.
5. Click the More Actions icon ⋮ and then click Add to existing products.
6. From the "Add to existing products" window, select one or more Products to which you want to add the API definition.
7. Click Add.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Testing an API with the API Manager test tool

You can test the API from within API Manager to ensure that is defined and implemented correctly.

About this task

The API Manager user interface has an integrated test tool. As part of the testing process the test tool stages and publishes a Product for you. The test tool then calls the API and displays the result of that call.

Procedure

To test an API, complete the following steps.

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane, and then click APIs.
   The APIs tab opens.
3. Click the name of the API that you want to test.
4. Click the Assemble tab.
5. Click the Test icon ▶.
6. If you have not yet configured the test tool for the API you are testing, or if you want to change the configuration of the test tool for your API, complete the following steps:
   a. If you have already configured the test tool, click Change setup.
   b. In the Catalog field, select the Catalog in which you want to test. The Catalog must be a development Catalog; for information on enabling development mode for a Catalog, see Creating and configuring Catalogs.
   c. To use an existing Product, in the Product field, select the Product with which you want to test your API.
   d. If your API is not included in the Product, click Add API.
   e. If you have made changes since you last published your Product, click Republish product.
   f. If you want to create a new Product to test the API with, provide a name for the Product in the Name field, and then click Create and publish.

   g. If there are multiple Plans in your chosen Product that contain your API, select the Plan you want to use in the Plan section's Name field.
   V5.0.3+ Note: If automatic subscription is enabled for the selected Catalog, you do not need to select a Plan because a test application, which is subscribed to all the Plans in the Catalog, is used automatically. For information on enabling automatic subscription for a Catalog, see Creating and configuring Catalogs.
   h. If you want to test by using an existing application, select the application from the Application field and then, if it is not already subscribed to the Product you are using, click Subscribe.
   V5.0.3+ Note: If automatic subscription is enabled for the selected Catalog, you do not need to provide an application because a test application is used automatically. For information on enabling automatic subscription for a Catalog, see Creating and configuring Catalogs.
   i. If you want to create a new application to test with, provide a name for the application in the App name field and then click Create and subscribe.
   Note:
      - You cannot access this application through the Developer Portal. Record the client ID and secret for your new application.
      - V5.0.3+ If automatic subscription is enabled for the selected Catalog, you do not need to provide an application because a test application is used automatically. For information on enabling automatic subscription for a Catalog, see Creating and configuring Catalogs.

   j. Click Next.
7. If your API is not included in the Product for which your test tool is configured, click Add API.
8. If you have made changes since you last published the Product with which you are testing your API, click Republish product.
9. In the Operation field, select the operation you want to test.
   The fields necessary to call your operation are populated by the test tool.
10. If you have input parameters, complete the corresponding fields.
11. If you want to run the test multiple times, select Repeat and then, in the Stop after: field, specify how many times you want your test to call the API. Select Stop on error to stop the test if it encounters an error.
12. Click Invoke to run the test.
13. Optional: Click Debug to view information about the API configuration.
    DataPower Gateway only V5.0.3+ Additional debug information, such as the input and output of the policy, is available for the invoke, map, and proxy policies.

    V5.0.7+ Restriction: If no API configuration information is displayed when you click the Debug button, a possible reason might be that access to analytics data within API Connect has been switched off. Contact your cloud administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)

Results

You successfully tested an API.

Note: You can also use the Explore tool within API Manager to test API endpoints. Click Explore and select Drafts or a specific Catalog. The API Explore tool opens and shows the operations for all of the APIs that are contained in your Drafts view or in the specified Catalog. The left pane of the Explore window can be used to select an operation to test. The center pane displays summary information about the endpoint, including its parameters, model instance data, and response codes, and the right pane provides template code to call the endpoint.

Related concepts

- Creating and configuring Catalogs

Related tasks

- Creating a Product
- Staging a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding an OpenAPI (Swagger 2.0) extension to an API definition (API Manager UI)

When you configure a REST or SOAP API definition, you can extend the OpenAPI (Swagger 2.0) specification by adding either a JSON or YAML extension schema depending on the version of IBM® API Connect you are using. V5.0.2 + You can also replace an extension with an updated version.

About this task

To make an extension available for adding to an API definition, you must first import a schema definition file for the extension. You import the extension into a Catalog; any extensions that are imported into Catalogs can be added to any of your API definitions.

V5.0.0 ONLY V5.0.1 ONLY Your schema definition file must be in JSON format.

V5.0.2 + Your schema definition file can be in YAML format or JSON format.

When you add an extension to an API definition, you can set values for any of the properties that are defined in the schema definition file for the extension. The values that you specify are subject to any validation criteria that are defined in the schema definition.

The following example shows an extension schema that defines a bank branch, in JSON format:

```
{
  "extension": "1.0.0",
  "info": {
    "title": "Banking services",
    "name": "banking",
    "version": "1.0.1",
    "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"]
  }
}
```

The following example shows an extension schema that defines a bank branch, in YAML format:

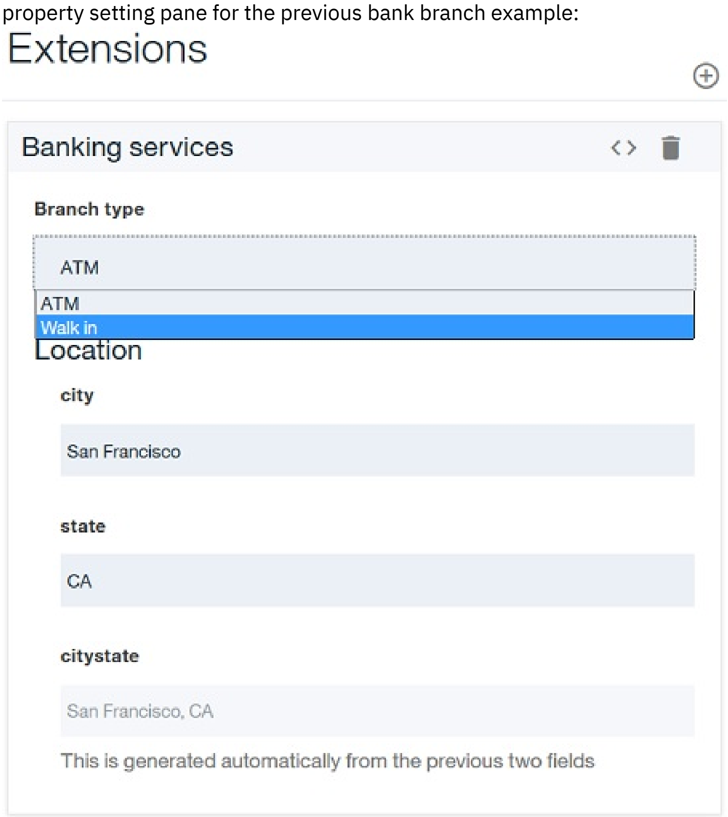
```
extension: '1.0.0'

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. To import an extension into a Catalog, complete the following steps:
   a. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

      The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
   b. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   c. Click Settings and select the Extensions tab.
   d. V5.0.0 ONLY V5.0.1 ONLY Import the schema definition file:
      i. Click the Import icon ⭱. The Add Extension Schema window opens.
      ii. Enter a Title, Name, and Version for the extension.
      iii. Select the JSON file that contains the schema definition for the extension, then click Add Extension. The new extension is added to the list of extensions.
   e. V5.0.2 + Import the schema definition file:
      i. Click the Import icon ⭱. The Import Extension window opens.
      ii. Select the YAML file that contains the schema definition for the extension, then click Import. The new extension is added to the list of extensions. The displayed title and version properties are derived from the corresponding property settings in the YAML file. If you import two or more extensions with the same `name` property in the YAML file but different `version` properties, they are grouped together in the list. You cannot import two extensions with the same name and version.
   f. Optional: V5.0.0 ONLY V5.0.1 ONLY To have instances of the extension sent to the Developer Portal, expand the extension and move the Visible in Developer Portal slider to the On position.
      Set this option if you want the extension to be available for use by application developers rather than being metadata for internal use only.
   g. Optional: V5.0.2 + To have instances of the extension sent to the Developer Portal, include the `portal-visible: true` setting in your YAML file; see the previous example.
   h. Click Save.
2. To add an extension to an API, complete the following steps:
   a. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   b. Click Drafts in the UI navigation pane, and then click APIs.
   c. Click the API that you want to work with.
      The API details page opens.

d. Navigate to the Extensions section, and click the Add Extension icon ⊕.
   The Add Extension window opens.
e. Select the extension that you want to add, then click Done.
   You can enter a search string to locate the required extension.
f. To set a value for any of the properties that are defined in the schema, specify the value in the appropriate field. Any validation criteria that are defined in the schema are applied to the value that you specify. The following screen capture shows the property setting pane for the previous bank branch example:



Extensions

Banking services

Branch type

ATM

ATM
Walk in

Location

city

San Francisco

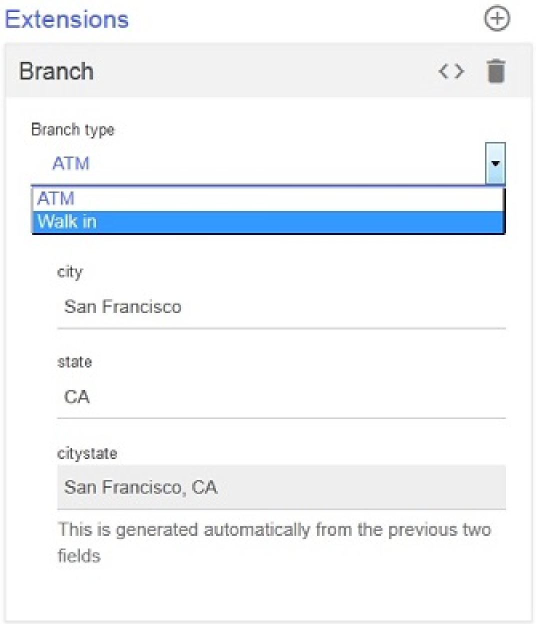
state

CA

citystate

San Francisco, CA

This is generated automatically from the previous two fields



Extensions

Branch

Branch type

ATM

ATM
Walk in

city

San Francisco

state

CA

citystate

San Francisco, CA

This is generated automatically from the previous two fields

g. V5.0.2 + To replace an extension with a new version, add an extension that has the same value for the `name` property as an extension that has already been added, but a different value for the `version` property. The Change Version of Extension window opens; click OK to replace the existing version with the new version, or Cancel to retain the existing version. For example, for the bank branch extension described earlier, the `Branch type` property might have an additional type added to the enumeration list.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

Organizing your APIs and Products into categories

You can organize your APIs and Products into categories. The APIs and Products that you categorize in the API Designer or API Manager UI 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

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 Designer or API Manager UI is different to tagging in the Developer Portal. For more information on tagging, see Providing navigation by tag hierarchy.

Procedure

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Click Drafts, then select the API or Product that you want to categorize.
   The Design window is displayed.
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 the Save icon 💾.
6. Navigate to Drafts and select the Product that you have categorized, or the Product that contains the API that you have categorized.
7. Stage the Product by clicking the Stage icon stage icon.
8. In the API Manager UI, publish the Product that you have staged by following the steps in Publishing a new Product.

Results

You have successfully published a Product that is categorized, or has an API that is categorized.
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 a LoopBack application through the API Designer.

What to do next

In the Developer Portal, you can 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Security and authentication

In API Manager, you can use TLS profiles to secure the transmission of data through websites, and also configure user registries to securely authenticate your Catalogs and APIs. V5.0.5+ You can also use LTPA keys to secure the transmission of data across WebSphere® Application Server domains.

For details of authentication in IBM API Connect, see the following subtopics:

- V5.0.5+ **LTPA keys**
  Lightweight Third Party Authentication (LTPA) is an IBM protocol that provides a cookie or binary security token based solution to support a single sign-on (SSO) environment. Create an LTPA key in API Manager to generate an LTPA token for accessing the back-end WebSphere Application Server web servers.
- **TLS profiles**
  In API Manager, TLS profiles are used to secure transmission of data through websites. 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

LTPA keys

Lightweight Third Party Authentication (LTPA) is an IBM protocol that provides a cookie or binary security token based solution to support a single sign-on (SSO) environment. Create an LTPA key in API Manager to generate an LTPA token for accessing the back-end WebSphere® Application Server web servers.

About this task

Use the API Manager UI to import an LTPA key. This key can then be used to protect the LTPA cookie or token.

IBM® API Connect supports the use of LTPA keys, but does not itself produce strong encryption keys or manage your encryption keys. LTPA keys must be exported from the LTPA peer, that is the WebSphere Application Server. For more information, see Exporting Lightweight Third Party Authentication keys in the IBM documentation for WebSphere Application Server V8.5.5.

Procedure

To create an LTPA key, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the navigation pane, select Admin.
3. Click Security > LTPA Keys > Add.
   The Create LTPA key window opens.
4. Enter a value in the Title field and, optionally, edit the Name and Version fields, and add a description if required.
   The name should be kept short, and can contain only lowercase alphanumeric characters (a-z and 0-9), or hyphen (-) or underscore (_) characters. A hyphen or underscore cannot be used as the first or last character in the name. The version should conform to the version.release.modification version numbering scheme, for example `1.0.0`.
   Note: The Version field is used by the Generate LTPA Token policy to indicate which version of the LTPA key to use for authentication. However, the policy can be configured to always use the latest version of the LTPA key. This feature means that you can create multiple versions of an LTPA key, and the policy automatically selects the latest version at run time. Note that the version numbering must be consistent for this feature to work correctly.
5. Click Select File and browse for the LTPA key file that you want to import.
6. In the LTPA key password field, enter the password for the key.
7. Click Save to save your changes.

Results

The LTPA key is imported into API Manager, and it can now be referenced in an API assembly by applying a Generate LTPA Token policy. For more information, see Generate LTPA Token (ltpa-generate).

Related information

- Single sign-on for authentication using LTPA cookies

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

TLS profiles

In API Manager, TLS profiles are used to secure transmission of data through websites. 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.

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 Updating a TLS profile and Generating a PKCS#12 file for a TLS profile.

Important: If you are using IBM® API Connect for IBM Cloud (the SaaS offering), you can use TLS profiles configured in the API Manager user interface to protect access to your back-end services, but you cannot use them to protect access on the front-end, such as connecting to the API Connect user interfaces or invoking API calls, because the front-end capability is controlled by the IBM Operations team on behalf of all customers. If you have a requirement to configure custom front-end TLS settings, please contact IBM by sending an email to ibmapi@us.ibm.com.
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. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the navigation pane, select Admin and click TLS Profiles.
3. In the TLS Profiles page, click Add.
4. Enter values in the Display Name and Name fields and, optionally, a description.
   The Name field cannot contain special characters.
5. Click Save to save your changes.
6. In the Present Certificate section, click the Upload Certificate icon ⊕.
   The Upload Certificate window opens.
7. Click Select File, then browse for the certificate file that you want to present for authentication.
   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.
8. In the Password text field, enter a password for the certificate.
   Note: The present certificate **must** be password protected.
9. Click Upload. The certificate is uploaded.
   Note: Only one present certificate can be uploaded at any one time. If you repeat the upload operation, the previous present certificate is replaced.
10. To validate the certificate, move the Request and validate the certificate against the supplied CAs in the truststore slider to the On position.
11. Click Save to save your changes.
12. In the Trust Store window section, click the Upload Certificate icon ⊕.
    The Upload Certificate window opens.
13. Click Select File, then browse for the Trust Store certificate.
14. If the trust store is password protected, enter the password In the Password text field.
    The trust store does not have to be password protected.

15. Click Upload.
    The certificate is populated.
    Note:
    - If the trust store certificate is expired, you must upload the entire certificate bundle to replace all current certificates.
    - API Connect supports **only** the P12 (PKCS12) and PEM certificate formats for the trust store.
16. Optional: Repeat steps 12 to 15 to add further trust store certificates.
17. Expand the Protocols section to display the SSL and TLS versions.
18. Select the check boxes that correspond to the TLS or SSL versions that you require.
19. V5.0.6 + To enable or disable SNI, expand the Features section or the V5.0.8 + Client Feature section, depending on the version of IBM API Connect you are using, then select Use SNI.
    V5.0.6 + Server Name Indication (SNI) is an extension to the TLS protocol. SNI is enabled by default, allowing clients to access multiple virtual domains on a single HTTPS server's IP address and port number. The TLS client injects the SNI extension with the desired host name in its initial handshake with the server. The server replies with the appropriate certificate to continue the interaction. Servers that do not support SNI often ignore this extension, but if you encounter compatibility issues, you can disable SNI.

20. Click Save.
    The certificates are uploaded and the SSL or TLS versions are saved.
    Note: After being uploaded, private keys cannot be downloaded from API Connect.

Results

The certificates are added to API Manager and the SSL or TLS version is saved.

- **Generating a PKCS#12 file for a TLS profile**
  API Manager supports only the PCKS#12 (P12) file format.
- **Updating a TLS profile**
  When a TLS or SSL certificate has expired, is close to expiration, or is invalidated, you need to update the TLS profile with a new CA certificate or PKCS#12 (P12) file.

Related tasks

- Working with user registries

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Generating a PKCS#12 file for a TLS profile

API Manager supports only the PCKS#12 (P12) file format. A P12 file is an archive file that contains all the required cryptographic objects.

About this task

This procedure is for illustrative purposes and uses OpenSSL commands. OpenSSL is an open source implementation of the SSL and TLS protocols.

To generate a P12 file, you must have the following files.

- A private key
- A root certificate that was signed by a Certificate Authority (CA)
- The intermediate certificates from the CA

Although all the steps are presented, you might not need to complete all the steps. Complete only the steps that are necessary based on which cryptographic material you already have for your environment.

Procedure

1. Generate the private key and certificate signing request (CSR).

```
openssl req -new -newkey rsa:length -nodes -keyout domain.key -out domain.csr
```

2. Send the CSR to your certificate authority (CA).
3. Download the signed certificate, usual a CRT file, and store the signed certificate and CA chain certificate in the same file as the CSR.
4. Create the PKCS#12 file.
    - With a single authority, use the following command.

      ```
      openssl pkcs12 -export -out file_to_generate.p12 -inkey domain.key -in cert_from_CA.crt -
      certfile CA_chain.crt
      ```

    - With multiple authorities, use the following command.

      ```
      openssl pkcs12 -export -out file_to_generate.p12 -inkey domain.key -in cert_from_CA.crt -name
      "cert_alias_name" -chain -CAfile certs.pem
      ```

      The `certs.pem` file contains a list of your certificate authorities from your intermediate authorities to the root authorities.

      ```
      --BEGIN CERTIFICATE--
      base64_intermediate_certificate
      --END CERTIFICATE--
      --BEGIN CERTIFICATE--
      base64_root_certificate
      --END CERTIFICATE--
      ```

Results

If you receive no errors, your P12 file is generated and ready to upload. If you receive an error review the documentation for the tool that you used to create the P12 file.

What to do next

Upload your P12 file to API Connect.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Updating a TLS profile

When a TLS or SSL certificate has expired, is close to expiration, or is invalidated, you need to update the TLS profile with a new CA certificate or PKCS#12 (P12) file.

About this task

CA certificate and P12 file expiration dates are displayed in the TLS Profiles page of API Manager.
Note: If you update a TLS profile that is associated with a Gateway service, the updates are not automatically propagated to Gateway servers. To resynchronize your servers with the latest configurations, see Gateway resynchronization.

Procedure

To update a TLS profile with an invalidated or expired certificate or P12 file, complete the following instructions:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. V5.0.4 and earlier In the navigation pane, select Admin and click TLS Profiles.
3. V5.0.4 and earlier Select the TLS Profile with an invalidated or expired certificate or P12 file.
   The page refreshes to display the TLS Profile details.
4. V5.0.5 + In the side bar, select Admin and select the Security tab.
5. V5.0.5 + In the side bar, click TLS Profiles.
6. V5.0.5 + From the list of TLS profiles, click the manage icon ⋮ for the TLS Profile with an invalidated or expired certificate or P12 file, then click Edit TLS profile.
   The page refreshes to display the TLS Profile details.
7. In the Present Certificate section, click the Upload Certificate icon ⊕.

The Upload Certificate window opens.

8. Click Select File, then browse for the new CA certificate or a P12 file.
9. Click Upload.
10. Click Save.

Results

The updated certificate or P12 file is added to the Cloud Manager.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 IBM 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 IBM API Connect for both user authentication and API security.

Authentication URL
: You can configure a non-LDAP user registry by using an authentication URL. An authentication URL enables integration with third-party authentication providers. You can use an authentication URL in IBM API Connect for both user authentication and API security.

SCIM
: Important: Support for the SCIM registry type is deprecated and the feature will not be available after the IBM API Connect Version 5.0 release.
IBM API Connect can authenticate with a user registry by using the System for Cross-domain Identity Management (SCIM) standard. Consider using SCIM if you have a custom user registry; you can implement a SCIM "bridge" to enable IBM API Connect to connect to your registry. SCIM can be used for user authentication but not for API security.
Note: The SCIM user registry type is supported only by the Developer Portal, or with a custom developer portal that is based on the public IBM API Connect REST APIs.

Local User Registry
: You can authenticate users with a local user registry. A local user registry is an internal registry stored within API Connect.

- **Working with user registries**
  To secure your 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 API Manager.
- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with user registries

To secure your API Connect Catalogs, you authenticate with user registries.

About this task

Perform the following steps to secure an API Connect Catalog with a user registry.
Important: If you are using IBM® API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.
Tip: Consider carefully how to configure the user registries for the Developer Portal. From API Connect Version 5.0.8.2, a user can delete their account in the Developer Portal. If a user is registered with several Developer Portal sites, and their account credentials are stored in one shared local user registry, then if the user deletes their account from any of the sites, the user loses their access to every Developer Portal site that shares that registry. From API Connect Version 5.0.8.4, if the local user registry is shared among the other API Connect applications (API Manager, API Designer, and Cloud Manager), the user loses access to all applications sharing the local user registry. Therefore, you might want to configure a different user registry for every Developer Portal site and for each application.

Procedure

To change the settings of the user registry used for a Developer Portal, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. Select the Catalog you want to work with in the Dashboard section of API Manager.
3. Open the Settings tab and then open the Portal tab.
4. In the User Registration and Invitation section of the Portal tab, use the drop-down menu to specify one of the following options:

   New LDAP Registry

   For more information, see Creating an LDAP registry.

   New Local User Registry

   If you select New Local User Registry, the Create New User Registry window opens.

   Note:

   LDAP has an option to set case sensitivity. However, the Developer Portal does not support case-sensitive usernames so that LDAP option is not supported in a Developer Portal.

   a. In the Display Name field, provide a name for the registry as it will appear in the API Manager user interface.
   b. In the Name field, provide a name for the registry that does not include any special characters.
   c. In the Description field, provide a description for the registry.
   d. If you want user names to be case-sensitive, set the case-sensitive usernames toggle to the Enabled position.
   e. Click Create.

   New Authentication URL

   If you select New Authentication URL, the New Authentication URL window opens.

   a. In the Name field, enter the URL name.
   b. In the Display Name field, enter a name to prompt users for their credentials.
   c. In the Realm field, provide additional details of which of their credentials they should provide.
   d. In the Description field, enter a description.
   e. In the URL text field, enter the URL.
   f. To authenticate with TLS, set the TLS Profile toggle to the Enabled position and use the drop-down menu to select a previously created profile.
   g. If the URL uses case-sensitive user names, set the case-sensitive usernames toggle to the Enabled position. By default, case sensitive user names are not used.
   h. Click Create.
5. To allow developers to invite collaborators to their organization and to allow them to assign particular roles, set the Developers can invite collaborators and assign the following roles toggle to the Enabled position.
6. Click the appropriate radio buttons to specify which roles developers can assign to collaborators who they invite.

   The options are as follows:

   Viewer

   The collaborator can access the Developer Portal and use APIs that are available to the organization. The collaborator cannot subscribe to Plans.

   App Developer

   The collaborator can create and edit the organization's applications, can manage client keys, and can subscribe to Plans.

   Viewer and App Developer

   The collaborator can perform all activities available to the developer.
7. Specify whether developers may perform their own sign-up process by setting the the Self-service onboarding toggle to the Enabled or Disabled position.

   When disabled, a user cannot join a developer organization without an invitation from the organization's owner.
8. Click Save to create the user registry.

Results

The user registry information is added to API Manager. If you selected self-service onboarding, the ability to self-serve is enabled.

What to do next

You can now use API Manager to chart and manage your Catalogs.

- **Creating an LDAP registry**
  You use the New LDAP Configuration window in API Manager to create a new LDAP registry.

Related tasks

- Modifying the configuration details for a user registry

Related information

- Developer Portal: discover and use APIs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating an LDAP registry

You use the New LDAP Configuration window in API Manager to create a new LDAP registry.

About this task

LDAP registries can be used to authenticate users to APIs, or for securing a Catalog to authenticate Developer Portal users. For more information about creating an LDAP registry definition for securing a Catalog, see Working with user registries.

There are three methods that API Connect can use to communicate with an LDAP registry. You can use the following methods when using the registry to authenticate users of an API:

Compose (UPN) (API authentication only)
: 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.

Compose (DN) (API authentication only)
: Select this format if you can compose the user LDAP Distinguished Name (DN) from the user name. The following example is of a DN format that can be composed from the user name: `uid=<username>,ou=People,dc=company,dc=com`. If you are unsure whether Compose (DN) is the correct option, contact your LDAP administrator.

Search (DN) (any authentication)
: 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.

If you are creating LDAP registries that you want to use with the Developer Portal, you can use only Search (DN).

Important: If you are using IBM® API Connect for IBM Cloud (the SaaS offering), any LDAP registry that you use *must* be visible on the internet, it must not be accessible only from within your corporate intranet.

Procedure

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. V5.0.4 and earlier Click Admin in the UI navigation pane, then click User Registries.
   The User Registries tab opens.
3. V5.0.5 + In the navigation pane, select Admin and select the Security tab.

4. V5.0.5+ Click User Registries.
5. If you are creating an LDAP registry that you want to use with the Developer Portal, click Add > LDAP Registry.
   The New LDAP Configuration window for configuring a Search (DN) authentication method opens.
6. If you are creating an LDAP registry to authenticate users of an API, click Add > LDAP Registry (API Security Only).
   The New LDAP Configuration window for configuring any of the three types of authentication format opens.
7. In the New LDAP Configuration window, provide a Name and a Display Name for the LDAP registry configuration and, optionally, a Description.
8. In the Hostname field, enter the host name.
9. In the Port field, enter a port number that API Connect can use to communicate with the LDAP registry.
10. In the LDAP version drop-down menu, select the registry version.
11. To protect user credentials, move the Use TLS slider to the On position, then use the drop-down menu to specify the TLS Profile, or indicate that you do not want to use a custom TLS profile by moving the slider to the Off position.
    Note:
    - If the Use TLS option is not chosen, user credentials are not protected in transit to the LDAP user directory. If your LDAP server is configured with TLS, choose this option to encrypt the transferred data.
    - The selected TLS profile must have the Public option selected, otherwise it can't be used for encryption between the LDAP server and API Connect. It is preferable to create a new TLS profile for this purpose rather than using a default supplied profile. For details on creating a TLS server profile, see TLS profiles.
12. Use the case-sensitive usernames toggle to specify whether user names are case-sensitive. Case-sensitive user names are disabled by default.
13. If you are using API Manager to create LDAP registries for authenticating users for APIs only, click Compose (UPN), Compose (DN), or Search (DN).
14. Complete one of the following steps, as appropriate for the method you selected:
    - If you selected Compose (UPN), enter the domain part of the user principal name in the Domain text field. For example, `@acme.com`.
    - If you selected Compose (DN), provide the user distinguished name prefix and suffix.
    - If you selected Search (DN), complete the following steps:
      - If specific permissions are not needed to search the registry, select Anonymous Bind or, if specific permissions are necessary, select Authenticated Bind.
      - If you selected Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory in the Admin DN field and then enter the user password in the Password field.
      - Supply a base DN in the Base DN field or click Test Bind and Get Base DN to populate the Base DN field with a retrieved base DN.
      - Optional: Update the Base DN with extra DNs. (Searches are performed from Base DN).
      - In the corresponding fields, provide a user name Prefix and Suffix.

    Note: As LDAP registries used by the Cloud Manager or a Developer Portal support only Search (DN), you cannot compose the user LDAP Distinguished Name from the user name. LDAP registries that are shared from the Cloud Manager or created from the Catalogs page might require an administrator DN and password to search for users. 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.
15. If you are using API Manager to create LDAP registries for authenticating with the Developer Portal, complete the follow steps to configure a Search (DN) LDAP registry:
    a. If specific permissions are not needed to search the registry, select Anonymous Bind or, if specific permissions are necessary, select Authenticated Bind.
    b. If you selected Authenticated Bind, enter the Distinguished Name of a user authorized to perform searches in the LDAP directory in the Admin DN field, and then enter the user password in the Password field.
    c. Supply a base DN in the Base DN field, or click Test Bind and Get Base DN to populate the Base DN field with a retrieved base DN.
    d. Optional: Update the Base DN with extra DNs. (Searches are performed from Base DN).
    e. In the corresponding fields, provide a user name Prefix and Suffix.
16. 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. To specify an LDAP authorization group, complete the following steps:
    a. Enable the Use Authorization Group option.
    b. To use a static group, select Static Group, then supply the Group Base DN, Prefix, and Suffix details for the group.
       A static group is one in which the individual members of the group are explicitly listed.
    c. To use a dynamic group, select Dynamic Group, then supply the Filter condition for the group.
       A dynamic group is one which is defined according to the set of attributes that the group members share in common.
17. When you are done, click Test configuration.
    If the test is successful, a confirmation message is displayed. If the test is not successful, an error message is displayed. Correct your settings and run the test again.
18. Click Create.

Results

The new LDAP user registry is created.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 API Manager.

About this task

If you want to authenticate an API Connect Catalog with a user registry that is not yet defined in IBM® 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 in API Manager:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ≡.
   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. V5.0.4 and earlier Click Admin in the UI navigation pane, then click User Registries.
   The User Registries tab opens.
3. V5.0.4 and earlier Click the registry that you want to edit in the pane on the left.
   The details of the registry are displayed.
4. V5.0.5 + In the navigation pane, select Admin and select the Security tab.
5. V5.0.5 + Click User Registries.
6. V5.0.5 + Click the Manage icon ⋮ for the user registry that you want to edit, then click Edit user registry.
   The details of the registry are displayed.
7. Modify the configuration details as required, then click Save to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with Products in the API Manager

In API Connect, Plans and APIs are grouped together in Products. You use the API Designer to create, edit, and stage your Product, and the API Manager to manage the Product lifecycle, and the availability and visibility of APIs and Plans.

Note: The API Manager UI also includes the ability to create, edit, and stage Products, however, the preferred method for these tasks is by using the API Designer UI in the developer toolkit. For more information about Products, and how to create and stage them, see Working with Products in the API Designer.
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. V5.0.5 + The syndication feature in IBM API Connect means that Product lifecycle states can also be managed within Spaces in the associated Catalog.
- V5.0.8 + **Billing for the use of your Products**
  In addition to offering free plans for your customers to use your Products, you can also define Plans that automatically bill your customers who are using your Products in IBM API Connect.
- **Managing your Products**
  You can manage your Products in API Manager by using the Products tab 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. V5.0.5 + The syndication feature in IBM API Connect means that Products can also be managed by using the Products tab of the associated Space.

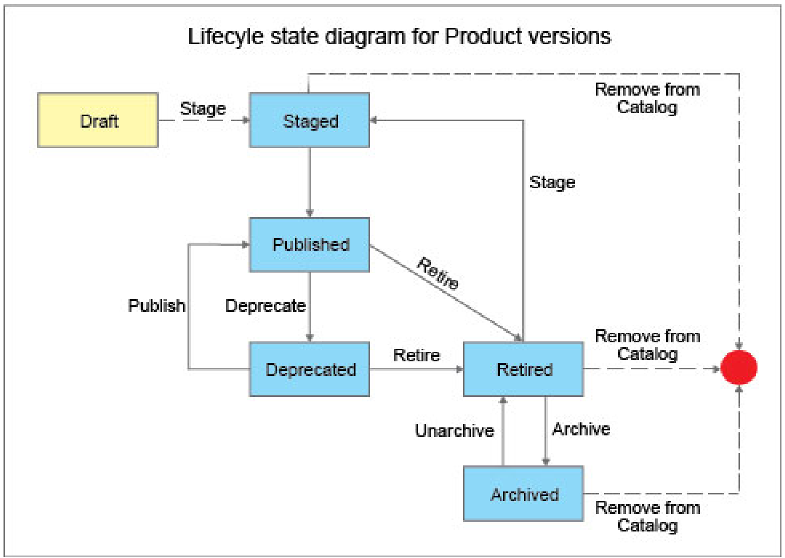
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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. V5.0.5 + 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.



V5.0.5 + 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 IBM 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.
- V5.0.5 + 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.
V5.0.5 + 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. Staging of Product versions is usually carried out in the API Designer. For more information about staging a Product, see Staging 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. 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.

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.

Archived

When you archive 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. The Product version is not displayed by default in the Products tab of the Catalog in API Manager. For more information about archiving Products, see Archiving a Product.

Related concepts

- Working with Catalogs

Related tasks

- Managing your Products
- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Billing for the use of your Products

In addition to offering free plans for your customers to use your Products, you can also define Plans that automatically bill your customers who are using your Products in IBM® API Connect.

Many companies have started billing customers for the use of their APIs as a additional revenue stream. With API Connect, you can configure billing subscriptions for members who have access to your Developer Portal, and credit card billing that automatically processes through a credit card processing service account.

When you make your Products available to your customers on your API Connect Developer Portal, you assign them one or more Plans that define the terms of use for the Product.

You can define free Plans and billing Plans. Your customers have different needs for the use of the Product, so you can offer different levels of service. An example of this is a smaller customer that only needs a rate of 5 Product API calls per hour. A larger company might need a rate of 1000 Product calls per hour. Since the smaller customer does not want to pay the same amount as the larger company for fewer calls, they can subscribe to a more limited Plan.

Though much of the procedure is the same for setting up a billing plan and a free plan, there are some extra actions that you need to complete when you define billing Plans.

- V5.0.8 + **Adding incoming billing information**
  You must add billing information to your IBM API Connect account that specifies where to collect the payments for the paid subscriptions to your API Plans.
- V5.0.8 + **Defining a subscription Plan with billing**
  You can create one or more subscription Plans in IBM API Connect that allow consumers to subscribe to your API Plans with integrated billing.

Related tasks

- Publishing a Product
- Managing your Products

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Adding incoming billing information

You must add billing information to your IBM® API Connect account that specifies where to collect the payments for the paid subscriptions to your API Plans.

Before you begin

If you use Stripe as your credit card processing vendor, you must have port 443 open to HTTPS communication between the Stripe API and your Developer Portal management and the Management cluster servers. See Firewall requirements for more information about this requirement.

About this task

When your customers subscribe to your Plans with billing, they set up automatic payments through a credit card with a supported credit card service provider. To complete the transactions when they are processed, you must specify the account information for your account with a supported credit card processor. The subscription payment amount is processed from the account that is provided by your customer, and credited to the account that you provide.

To connect your API Connect account to a payment card processing account, complete the following steps:

Procedure

1. Open the API Connect API Manager interface.
2. Ensure that you are in the organization to which you want to add the billing information.
   Important:
   There is no sharing of data between Stripe accounts, and there is no sharing of data between a single account's test keys and production keys. If you plan to use test keys for a Stripe account, make sure that you use them on a test provider organization. Only use the live API keys of your production Stripe account with your production provider organizations.

   The Stripe subscription plans, customers, payment methods, and subscriptions created using a test key are not visible when using a production key or when using the key to another account. If you change your Stripe keys after publishing a monetized product (for example, to a different account, or from test keys to production keys), the references that link APIC resources to the corresponding Stripe resources cannot be resolved. Subsequent operations might fail, resulting in inconsistent data between the APIC and Stripe systems.

3. In your API Connect Designer interface, select Navigate to ☰) > Admin.
4. Select the Billing tab.
5. Select Add + on the Billing Integrations screen.
6. Select the type of integration from the list.
7. Connect your Stripe account.
   a. If you do not already have a Stripe account, set one up at: www.stripe.com.
   b. Select api keys in the message window to view the keys that you need from your existing Stripe account.
   c. Copy the publishable key token from the Stripe API keys list and add it to the *Stripe Publishable Key* field in the API Manager window.
   d. Copy the Secret key token from the Stripe API keys list and add it to the *Stripe Secret Key* field in the API Manager window.
   e. Select Create to apply the keys to your account.
      The STATUS for the Stripe Integration that you created shows as Connected.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Defining a subscription Plan with billing

You can create one or more subscription Plans in IBM® API Connect that allow consumers to subscribe to your API Plans with integrated billing.

Before you begin

If you use Stripe as your credit card processing vendor, you must have port 443 open to HTTPS communication between the Stripe API and your Developer Portal management and the Management cluster servers. See Firewall requirements for more information about this requirement.

About this task

When you decide to charge customers to use your API Products, your customers have different requirements for the number of API calls that they are willing to pay for. You can solve this issue by defining Plans at different levels of service, and at different costs.

As an administrator of the account, you can define free plans, as well as Plans that contain billing information. The procedures for defining a free Plan and a Plan with billing are similar.

To create a subscription plan with billing for an API Product, complete the following steps:

Procedure

1. Open the API Connect API Manager interface.
2. In your API Connect Designer interface, select Navigate to ☰) > Drafts.
3. Select the Products tab.
4. If you see the Product that you want to define your plan listed, select the Product name to open it and continue with step 6.

Otherwise, create a Product by continuing with step 5.

5. Create a Product.
   1. Select Add + to add a new Product.
   2. Select New Product if you want to create a new Product manually, or select Import an existing product if you have an existing Product that you want to use as a base for this one.
   3. Specify the required information for your new Product.
   4. Select Create product
6. With the Product open and the Design tab selected, select Plans in the navigation menu.
7. Expand the details about the Default Plan by selecting the Expand icon.
8. Add a new name for your Plan in the *Title* field.
   This is the name that your subscribers see when they view your Plan in the Developer Portal.
9. Provide a name for your Plan in the *Name* field.
   This must be a string of characters and numbers, and is not displayed to subscribers in the Developer Portal.
10. Add an extended description of your Plan that provides more information about the Plan. Your subscribers also see this description when they are selecting a plan.
11. Complete the billing details for your plan.
    1. Select your billing model from the list in the *Billing Model* field.
       The Monthly Subscription option bills the customer for service every 30 days.
    2. Select the Currency that the billing process should use.
    3. Enter the cost that should be billed to a subscriber every 30 days.
       This field must contain an amount that is at least 1, and is a required field for a billing plan.
    4. Enter the number of trial days that a subscriber can use the Product without charge, after which the payment process begins.
12. Set the rate limits for the Plan by entering the number of calls that are allowed per unit of time.
    Remember that this rate limit only applies to this Plan. You can also create other Plans with other rate limits.
13. Select the Enforce hard limit box if you want the gateway to reject any calls beyond the specified rate, rather than adding them to a queue to be resolved when the limits reset.
14. Set the Burst limits for the Plan by entering the number of calls that are allowed per unit of time.
    Remember that this rate limit only applies to this Plan. You can also create other Plans with other rate limits.
15. Select the box for Require subscription approval to require approval for someone to subscribe to this Plan.
16. Save your Product.
17. Complete this procedure for any other Plans that you want to create.

Results

Your Plan with billing is ready for publishing to your Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing your Products

You can manage your Products in API Manager by using the Products tab 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. V5.0.5 + 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.

V5.0.5+ For more information about the syndication feature, see Using syndication in IBM API Connect.

Procedure

To manage your Product, complete the following steps.

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5+ If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Click a Product name to expand the view and see further details.
   For each Product version you can see the following details and options:
   - The name of the Product version.
   - The current lifecycle state of the Product (staged, published, deprecated, retired, or archived).
   - The APIs that are included in the Product and whether they are online or offline. You can use the slider to quickly make an API offline or online. For example, if maintenance work on the API is needed, move the slider to the off position while the work is being performed, and then move the slider back to the on position after the work is complete.
   - The Plans that are included in the Product, and the number of subscribers that are subscribed to each Plan.
   - Manage actions are also available, depending on the state of the Product. V5.0.7+ To view the manage actions, click the Manage icon ⋮ and select the action that you require. V5.0.6 and earlier To view the manage actions, click the Manage icon ••• and select the action that you require.
4. V5.0.7+ To display the analytics information for a version of a Product from the expanded Product view, click the Manage icon ⋮ for that Product version, and then click Product analytics. V5.0.6 and earlier To display the analytics information for a version of a Product from the expanded Product view, click the Manage icon ••• for that Product version, and then click Product analytics.
   You are redirected to the Analytics tab, which shows an analytics dashboard for the Product. The default dashboard is titled ***Catalog_name* -**
   Product_name or ***Space_name* -**
   Product_name, and shows analytics information in the form of *visualizations*, which are represented as charts and numerical metrics. The default dashboard for the Product displays the following metrics and charts for the last seven days:
   - The developer organization usage
   - The number of calls that are made to all APIs included in the Product
   - A pie chart that shows the distribution of applications that have accessed each Plan in the Product
   - A bar chart that shows the number of API calls per day
     V5.0.7+ 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 has been switched off. Contact your cloud administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)

   The dashboard for the Product is similar in layout to the analytics dashboards for Catalogs and provides similar functions. For a description of the layout of the screen elements, see The default Overview dashboard. You can work with the dashboard as follows:
   - Modify the dashboard by adding, editing, or removing visualizations, or by resizing or rearranging the visualizations.
   - Apply filters to change the time range of the data displayed or the type of data displayed, and also set an auto-refresh rate for the data.
   - Share a view of your customized dashboard with other users.
   - View and export the raw analytics data and API event records that are collated for each visualization, and also export the API event data that is associated with the entire dashboard to a CSV file.

   Note: V5.0.7+ Below the search bar on the dashboard, the blue oval and Actions twistie, which are displayed, depict the inclusion filter that is applied for the Product-based data shown in the visualizations. V5.0.6 and earlier Below the search bar on the dashboard, the green oval and Actions twistie, which are displayed, depict the inclusion filter that is applied for the Product-based data shown in the visualizations.
   To return to the list of Products, click the Products tab again.
5. To display the analytics information for a Plan from the expanded Product view, click the Analytics icon 📊 for that Plan.
   You are redirected to the Analytics tab, which shows an analytics dashboard for the Plan. The default dashboard is titled ***Catalog_name* -**
   Product_name*:*Product_version*:*Plan_name or ***Space_name* -**
   Product_name*:*Product_version*:*Plan_name, and displays the following metrics and chart for the last seven days:
   - The number of calls that are made to all APIs that are subscribed to the Plan in that Product
   - The number of applications that are subscribed to the Plan
   - A line chart that tracks the number of calls made by an application within the defined rate limit time window against the maximum number of calls allowed during that time window; this view helps in identifying calls that exceeded the rate limit, and which were either rejected with a 429 status code or recorded in the Activity log
     V5.0.7+ 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 has been switched off. Contact your cloud

administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)
The dashboard for the Plan is similar in layout to the analytics dashboards for Catalogs and provides similar functions. For a description of the layout of the screen elements, see The default Overview dashboard. You can work with the dashboard as follows:

- Modify the dashboard by adding, editing, or removing visualizations, or by resizing or rearranging the visualizations.
- Apply filters to change the time range of the data displayed or the type of data displayed, and also set an auto-refresh rate for the data.
- Share a view of your customized dashboard with other users.
- View and export the raw analytics data and API event records that are collated for each visualization, and also export the API event data that is associated with the entire dashboard to a CSV file.

Note: V5.0.7+ Below the search bar on the dashboard, the blue ovals and Actions twistie, which are displayed, depict the inclusion filters that are applied for the Product and Plan-based data shown in the visualizations. V5.0.6 and earlier Below the search bar on the dashboard, the green ovals and Actions twistie, which are displayed, depict the inclusion filters that are applied for the Product and Plan-based data shown in the visualizations.
To return to the list of Products, click the Products tab again.

6. To display the analytics information for an API from the expanded Product view, click the Analytics icon 📊 for that API.
You are redirected to the Analytics tab, which shows an analytics dashboard for the API. The default dashboard is titled
Catalog_name -
API_name or ***Space_name -***
API_name, and displays the following charts and metrics for the last seven days:

   - A pie chart that shows the total count for each status code that was generated for the API over the time period
   - A bar chart that shows the distribution of error types that were generated each day
   - The minimum time taken to process the API request in milliseconds
   - The average time taken to process the API request in milliseconds
   - The maximum time taken to process the API request in milliseconds
   - A line chart that collectively tracks the minimum, average, and maximum time taken to process the API request in milliseconds, for comparison
   - The total number of calls that were made to the API
   - A bar chart that shows the number of calls per day to the API
V5.0.7+ 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 has been switched off. Contact your cloud administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)

   The dashboard for the API is similar in layout to the analytics dashboards for Catalogs and provides similar functions. For a description of the layout of the screen elements, see The default Overview dashboard. You can work with the dashboard as follows:

   - Modify the dashboard by adding, editing, or removing visualizations, or by resizing or rearranging the visualizations.
   - Apply filters to change the time range of the data displayed or the type of data displayed, and also set an auto-refresh rate for the data.
   - Share a view of your customized dashboard with other users.
   - View and export the raw analytics data and API event records that are collated for each visualization, and also export the API event data that is associated with the entire dashboard to a CSV file.

   Note: V5.0.7+ Below the search bar on the dashboard, the blue oval and Actions twistie, which are displayed, depict the inclusion filter that is applied for the API-based data shown in the visualizations. V5.0.6 and earlier Below the search bar on the dashboard, the green oval and Actions twistie, which are displayed, depict the inclusion filter that is applied for the API-based data shown in the visualizations.
   To return to the list of Products, click the Products tab again.

7. To manage the lifecycle of a version of a Product, click the Manage icon ··· alongside the Product version, and select the required lifecycle action.
8. To view the approval history of a version of a Product, click the Manage icon ··· alongside the Product version, and select Approval history.
9. V5.0.4+ To set the migration target of a version of a Product, click the Manage 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.
The migration target is displayed next to the migration source Plans. For information about how to migrate the subscriptions to the target Product, see Migrating application subscriptions to another Plan.
V5.0.5+ 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 IBM API Connect.

What to do next

For details of the ways in which you can manage your Products, see the following subtopics:

- V5.0.8 + **Considerations when changing a Product with billing**
  In some cases, you have to change the Products and Plans with billing that your customers are already subscribed to in IBM API Connect.
- **Publishing a Product**
  APIs become accessible when a Product is published and made visible on the 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.
- **Migrating application subscriptions to another Plan**
  By using the Products view within a Catalog in API Manager, you can migrate application subscriptions between Plans, including between different Products. V5.0.5 + The syndication feature in IBM API Connect means that you can also use the Products view within a Space in a Catalog to migrate application subscriptions.
- **Migrating subscribed users to a new Product version**
  When new versions of Products are created, you must do some Product management to ensure that users that are subscribed to Plans in the previous version of the Product, are migrated to the Plans of the new Product.
- **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. V5.0.5 + 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 Products view 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. V5.0.5 + The syndication feature in IBM API Connect means that you can also use the Products view 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 Products view within a Catalog in API Manager. When a Product is retired, all associated APIs are taken offline, and any subscriptions become inactive. V5.0.5 + The syndication feature in IBM API Connect means that you can also use the Products view within a Space in a Catalog to retire a Product.
- **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. V5.0.5 + 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.
- **Archiving a Product**
  You can archive a retired Product by using the Products view within a Catalog in API Manager. When a Product is archived, all associated APIs are taken offline, and any subscriptions become inactive. V5.0.5 + The syndication feature in IBM API Connect means that you can also use the Products view within a Space in a Catalog to archive a Product.
- **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 Approvals tab in the Catalog that contains the associated Product in API Manager. V5.0.5 + The syndication feature in IBM API Connect means that you can also use the Approvals tab within a Space in a Catalog to approve or decline requests.
- **Downloading Product Details**
  You can download the details of your Product from API Manager, so that you can store them for later recovery.
- **Importing a Product**
  You can create a Product from a .yaml file by using the API Manager.

Related concepts

- Creating and configuring Catalogs

Related tasks

- V5.0.5 + Working with Spaces

Related information

- Staging a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Considerations when changing a Product with billing

In some cases, you have to change the Products and Plans with billing that your customers are already subscribed to in IBM® API Connect.

There are times when you must modify a Product or paid subscription Plan that customers are already subscribed to. Some different scenarios include the following types:

Deprecating
: Preparing an existing Product to be removed from production. There is no known replacement, and it is no longer available for new subscribers. Existing subscribers can continue to use the Product until it is removed.

Migrating
: Moving from one API Connect Plan to another. This often involves a change in billing cost. This change cannot be completed by the Product owner or by the Developer Portal administrator.

Replacing
: Updating an existing Product, but maintaining the existing cost and service requirements for the Plan. The Developer Portal administrator can make this change, as long as the costs do not change.

Retiring
: Removing a Product or Plan from production.

Superseding
: Retaining the existing Product and Plan for subscribers who would like to keep it, but offering a different Product and Plan for the updated version of a Product. The customer can continue with the existing Product and Plan, or move to the new Product and Plan. The administrator cannot automatically migrate a customer to a higher-priced version of the Product. New subscribers can only subscribe to the new version of the Product, and cannot see the old version of the Product.

While these follow the same basic procedures that are described in Managing your Products for Products and Plans without billing, there are some considerations that only apply to Products that have paid Plans.

You cannot migrate a customer directly from a free Plan to a Plan with billing.

Customers who subscribe to Plans with billing must set up their credit card processing account. If they only subscribe to free plans, they do not have credit card processing accounts.

There are two ways to migrate customers from a free plan to a paid plan:

- Supersede the free Plan and provide a migration target to a paid Plan. This deprecates the free Plan so no new users can subscribe to it, and gives the customer a Migrate button in the Developer Portal to move to the new plan.
- Set a migration target to a paid Plan. In this case the free Plan is not deprecated, but it still gives the customer the Migrate button in the Developer Portal for moving to the new Plan. With this method, new customers can still subscribe to the free Plan.

Both of these methods require action from the customer before they move from the free Plan to the paid Plan.
Note: Free trial days do not apply to customers when they migrate. They are only available for new customers.

You must map the existing Plans to the new Plans when you migrate or replace a Product.

When you replace or supersede a Plan with billing with another Plan, you must map the Plans from the Product that is being removed to the Plans that are in the new Product. It is convenient if there are the same number of plans for each, but you can map more than one Plan from the existing Product to one plan in the new Product.

Billing procedure is determined by the credit card processor when migrating or retiring.

When customers are migrated to a new paid Plan, the billing process is determined by the credit card processing provider. For example, if the migration happens in the middle of a billing cycle, do you pay the new fee for the entire cycle? Your customers should be aware of the terms with the credit card processing provider.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Publishing a Product

APIs become accessible when a Product is published and made visible on the 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 a LoopBack application through the API Designer.
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 developer 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. V5.0.5 + 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 with another 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Publishing a new Product

Publish a Product from within its containing Catalog in API Manager. V5.0.5 + 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

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 tab of API Designer or API Manager 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 tab of API Designer or API Manager 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.

V5.0.4 ONLY When you use the test tool in a development Catalog, any Product that you test is forced through and overwrites staged and published Products even if the APIs are being used in the Developer Portal.

Note:

- When you publish, replace, or supersede a Product, there is a short delay before the APIs in the Product are available to be called through the API Connect gateway.
- V5.0.5+ 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 IBM API Connect.

Procedure

To publish a Product, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5+ If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to work with, click the Manage icon ⋮ and then click Publish.
   The Edit visibility and subscribers dialog box is displayed.
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 developer 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 (or, if your Product was in the Deprecated state, click Republish).
   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.
- V5.0.5+ 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Replacing a Product with another Product

You can replace a published Product with another Product by using the API Manager.

Before you begin

The replacement Product must be in the Staged, Deprecated, or Published state. The Product to be replaced 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.

V5.0.5+ Restriction: If the Products are contained within a Space in the Catalog, and the replacement action is being made at the Space level, the replacement Product must be located in the same Space as the Product to be replaced. If the replacement action is being made at the Catalog level, Products can be replaced across Spaces. For more information about the Space feature, see Using syndication in IBM API Connect.

About this task

When you replace a Product with another Product, the following actions are taken:

- The replacement Product is published.
- The same visibility, subscriber, and gateway enforcement settings from the original Product are used in the replacement Product.
- The subscribers to the original Product are migrated to the replacement Product.

- V5.0.8 + For paid Plans, a message indicating that there is a change is displayed. The billing schedule and charges begin immediately with the new Plan. Free trial days do not apply to migrated Plans.
  Note: Customers cannot be migrated automatically from a free Plan to a paid Plan. To move your customers from a free Plan to a paid Plan, you can supersede the product with a new product and set a migration target to the paid Plan. The customer then selects a button to migrate, and must enter their credit card information before the process is complete.
- 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.

V5.0.4 ONLY When you use the test tool in a development Catalog, any Product that you test is forced through and overwrites staged and published Products even if the APIs are being used in the Developer Portal.

Note:

- When you publish, replace, or supersede a Product, there is a short delay before the APIs in the Product are available to be called through the API Connect gateway.
- 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the replacement Product version, click the Manage icon ⋮, then click Replace an existing Product.
   The Replace an existing product window opens.
4. Click the Product that you want to replace, and click Next.
5. From the drop-down list, select which Plans from your new Product correspond to Plans in your old Product.
   API Manager automatically completes this page when there is a clear correspondence. For example, when there are Plans in both the new and old Products that are called Gold.
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.
   - V5.0.5 + 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 superseding Product must be in the Staged, Deprecated, or Published state. The Product to be superseded 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.

V5.0.5 + Restriction: If the Products are contained within a Space in the Catalog, and the superseding action is being made at the Space level, the superseding Product must be located in the same Space as the Product to be superseded. If the supersede action is being made at the Catalog level, Products can be superseded across Spaces. For more information about the Space feature, see Using syndication in IBM API Connect.

About this task

When you supersede a Product with another Product, the following actions are taken:

- The superseding Product is published.
- The same visibility, subscriber, and gateway enforcement settings from the original Product are used for the superseding Product.
- The original Product is moved to the Deprecated state. When a Product is deprecated, application developers that are already subscribed to the Product can continue to use it, but no new developers can subscribe to the Product. A 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.

V5.0.4 ONLY When you use the test tool in a development Catalog, any Product that you test is forced through and overwrites staged and published Products even if the APIs are being used in the Developer Portal.

Note:

- When you publish, replace, or supersede a Product, there is a short delay before the APIs in the Product are available to be called through the API Connect gateway.
- 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.

The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.

3. Alongside the superseding Product version, click the Manage icon ⋮ and then click Supersede an existing Product..
   The Supersede an existing product window opens.
4. Click the Product that you want to supersede and then click Next.
5. From the drop-down list, select which Plans from your new Product correspond to Plans in your old Product.
   API Manager automatically completes this page when there is a clear correspondence. For example, when there are Plans in both the new and old Products that are called Gold.
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.
   - V5.0.5 + 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 superceded 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.

V5.0.8 + Customers who are already subscribed to the deprecated Plan are not disconnected from their credit card billing provider subscription to the deprecated Product, as it is still available for existing customers to use. New customers cannot subscribe to the deprecated Plans.

Related concepts

- The Product lifecycle

Related tasks

- Deprecating a Product
- Publishing a new Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Migrating application subscriptions to another Plan

By using the Products view within a Catalog in API Manager, you can migrate application subscriptions between Plans, including between different Products. V5.0.5 + The syndication feature in IBM® API Connect means that you can also use the Products view within a Space in a Catalog to migrate application subscriptions.

Before you begin

Your Plan that you are migrating subscriptions from must be a part of a Product that is in one of the following states:

- Published
- Deprecated

V5.0.8 + If you are migrating to a paid plan, you must have a credit card account set up with a supported credit card processing vendor.

To complete the Product and Plan 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 Plan and its parent Product. If you have View permission for Products in this Catalog, you will 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.

V5.0.5 + 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 IBM API Connect.

About this task

Application developers initially subscribe their applications to one or more Plans by using the Developer Portal. For more information, see Selecting a Plan in the developer portal. However, by using API Manager, for a chosen Plan you can migrate one or more of its application subscriptions to another Plan, which may be a part of the same Product or a part of a different Product.

Procedure

To migrate Plan subscriptions, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Expand the Product version that is the parent of the Plan you want to migrate subscriptions from.
   The view expands to display all of the APIs and Plans contained in the Product.
4. Alongside the Plan version whose application subscriptions you want to migrate, click the Migrate subscriptions icon ⇥.
   The Migrate subscriptions to another plan dialog box opens.
5. Select the application subscriptions that you want to migrate, and then click Next.
6. Click the Plan that you want to migrate the application subscriptions to, and then click Migrate.

Results

The new Plan has the migrated subscriptions, and those subscriptions are removed from the Plan from which they were migrated.

Related tasks

- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Migrating subscribed users to a new Product version

When new versions of Products are created, you must do some Product management to ensure that users that are subscribed to Plans in the previous version of the Product, are migrated to the Plans of the new Product.

About this task

You can migrate users in different ways, depending on the changes you made to a Product.

Procedure

To move all users to the Plan of a Product that has some fixes applied:

- You should replace the original version of the Product with the new version of the Product; the subscribers are automatically subscribed to the new version of the Product. For more information about replacing Products, see Replacing a Product with another Product.

If an enhancement or new feature is added:

- You should supersede or co-publish the new version of the Product, and then migrate the users manually. For more information, see Migrating application subscriptions to another Plan.

If a Product is being deprecated, but you want existing subscribers to move to a different Product:

- You should migrate the existing subscriptions to another Product when you deprecate the original Product. For more information, see Deprecating a Product.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. V5.0.5+ 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:

- When you publish, replace, or supersede a Product, there is a short delay before the APIs in the Product are available to be called through the API Connect gateway.
- V5.0.5+ 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 IBM API Connect.

Procedure

To change the availability of a Product, complete the following steps:

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5+ If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to work with, click the Manage icon ⋮, then click Edit visibility.
   The Edit visibility dialog box opens.

4. Specify the following options:

    - The users that the Product is visible to

        You can choose Public (Developer Portal), Authenticated (Developer Portal), 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.
        - Use the Temporarily disable visibility check box to toggle whether the Product is or is not visible, without changing the other visibility settings. This setting is available only if the Product is in the Published state.

    - Who can subscribe to the Product

        You can choose Authenticated (Developer Portal), or Custom. You can add or remove one or more developer 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.
        - Use the Temporarily disable subscribability check box to toggle whether the Product can or cannot be subscribed to without changing other subscription settings. This setting is available only if the Product is in the Published state.
5. Click Republish or Done, depending on the lifecycle state of your Product.
    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.
    - Also, 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

- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deprecating a Product

You can deprecate a published Product by using the Products view 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. V5.0.5 + The syndication feature in IBM® API Connect means that you can also use the Products view 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to work with, click the Manage icon ⋮, then click Deprecate.
   The Deprecate product dialog box opens.
4. Click OK 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.
   - V5.0.5 + 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. If you want to migrate the current Plan subscriptions to a different Product, see Migrating application subscriptions to another Plan.

Related concepts

- V5.0.5 + Using syndication in IBM API Connect

Related tasks

- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retiring a Product

You can retire a published or deprecated Product by using the Products view within a Catalog in API Manager. When a Product is retired, all associated APIs are taken offline, and any subscriptions become inactive. V5.0.5 + The syndication feature in IBM® API Connect means that you can also use the Products view 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to work with, click the Manage icon ⋮ and then click Retire. The Retiring *product_name_version* window opens. Click OK.
   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.
   - V5.0.5 + 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 canceled. You can stage the Product again later if required; for details, see Staging a Product.
V5.0.8 + The billing with the credit card processing provider is stopped when the Product or Plan is retired. The billing terms and prorating of subscription time is determined by the credit card processing provider.

Related concepts

- V5.0.5 + Using syndication in IBM API Connect

Related tasks

- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. V5.0.5 + 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, Retired, or Archived state. For more information, see Staging a Product, Retiring a Product, or Archiving 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to remove, click the Manage icon ⋮, click Delete from catalog, and then click OK.

Results

Your Product is removed from the Catalog. You can stage the Product again if required; for details, see Staging a Product.
V5.0.5 +If your Product was contained within a Space in a Catalog, your Product is removed from both the Space and the Catalog.

V5.0.8 +Any subscriptions to the product are suspended. If the subscription is a paid subscription, the billing ends after the Product is removed.

Related concepts

- V5.0.5 +Using syndication in IBM API Connect

Related tasks

- V5.0.5 +Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Archiving a Product

You can archive a retired Product by using the Products view within a Catalog in API Manager. When a Product is archived, all associated APIs are taken offline, and any subscriptions become inactive. V5.0.5 +The syndication feature in IBM® API Connect means that you can also use the Products view within a Space in a Catalog to archive a Product.

Before you begin

The Product that you are archiving 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.

Procedure

To archive a Product, complete the following steps.

1. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.

The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.

2. V5.0.5 + If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Alongside the Product version that you want to work with, click the Manage icon ⋮ and then click Archive. The Archiving product window opens. Click OK.

Results

Your Product is moved to the Archived state. It is also removed from the Developer Portal and is no longer visible to the application developers.
Note:

- By default, the filter settings for the Products tab are set to hide archived Products, so if you archive a Product it is removed from the list. To include archived Products in the list on the Products tab, complete the following steps:
  1. Click the Show or hide filters icon .
  2. Select the Archived state.
- You can move your Product back into the Retired state by clicking the Manage icon ⋮, and then clicking Unarchive. V5.0.8 + Customers who were subscribed to the Product before it was archived must subscribe to the Product again. The subscriptions are not restored when it is moved from the Archived state.

Related concepts

- V5.0.5 + Using syndication in IBM API Connect

Related tasks

- V5.0.5 + Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 Approvals tab in the Catalog that contains the associated Product in API Manager. V5.0.5 + The syndication feature in IBM® API Connect means that you can also use the Approvals tab 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.

V5.0.5 + The syndication feature in IBM 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 Approvals tab on the Catalog page, 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.
- V5.0.5+ 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. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
2. V5.0.5+ If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
3. Select the Approvals tab, then locate the Product lifecycle change request that you want to deal with.
4. Click the Approve ✓, Decline ⊗, or Cancel Request ✕ icon as required. Depending on the action you selected, either the Approve this request, Decline this request, or Cancel this request window opens.
   Note:
   - The option to cancel a lifecycle change request for a Product is available only if you are assigned a user role that has permission to manage Products in the Catalog in which the Product resides.
   - 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.
5. Optional: Enter a comment.
6. Click Approve, Decline, or Cancel Request.

To work with a subscription request, complete the following steps.

7. On the Dashboard of the API Manager UI, select the Catalog that you want to work with.
   The Products tab of the Catalog opens, and all of the Products available in that Catalog are displayed. You can select which states are shown by clicking the filter icon in the Search banner and selecting one or more of the State check boxes.
8. V5.0.5+ If the Product that you want to work with is contained within a Space, select the required Space by clicking the down arrow next to the Catalog name in the submenu navigation banner.
   The Products tab of the Space opens, and all of the Products available in that Space are displayed. You can select which states are shown by using the filter check boxes in the State bar.
9. Select the Approvals tab, 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.
10. Click the Approve ✓ or Decline ⊗ icon 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

- V5.0.5+ Using syndication in IBM API Connect

Related tasks

- V5.0.5+ Working with Spaces

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Downloading Product Details

You can download the details of your Product from API Manager, so that you can store them for later recovery.

Before you begin

To complete this task, you must have created a Product. For more information, see Creating a Product.

About this task

You can download the details of your Product from API Manager 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 Product.

Procedure

To download a .yaml file containing details of your Product, complete the following instructions:

1. Select the Products tab in the Drafts section of API Manager.
   The Products tab opens.
2. Expand the Product version that you want to download details of.
   The Product Design page opens.
3. Click the More Actions icon ⋮.
4. Click Download.

Results

You have downloaded a .yaml representation of your Product.

Related tasks

- Importing a Product

Related information

- Creating a Product

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Importing a Product

You can create a Product from a .yaml file by using the API Manager.

Before you begin

To complete this task, you must have a .yaml representation of a Product. This can be created from API Manager as described in the Downloading Product Details task.

About this task

You can import details of your Product into API Manager 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.

Procedure

To create a Product from a .yaml file, complete the following instructions:

1. Select the Products tab in the Drafts section of API Manager.
   The Products tab opens.
2. V5.0.4 + Click Add > Import an existing product.

   V5.0.3 and earlier Click the Add Product icon ⊕ and then click Import YAML.
3. From the Import Product window, click Choose File or Browse (as displayed by your browser) to select the file you want to upload. Then click Import.
   The imported Product is displayed in the list of Products.

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 in API Manager, 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 concepts

- The Product lifecycle

Related tasks

- Downloading Product Details

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Analytics

You can filter, sort, and aggregate your API event data, and then 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.

V5.0.7 + API analytics is built on the Kibana V5.1 open source analytics and visualization platform, which is designed to work with the Elasticsearch real-time distributed search and analytics engine. V5.0.6 and earlier API analytics is built on the Kibana V4.3 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:

- V5.0.5 + **Analytics and syndication**
  The data for Analytics is collated from API events that are logged when API operations are invoked. 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.
- **Accessing analytics**
  You can view predefined or customized analytics information for your IBM API Connect Catalogs within dashboards. V5.0.5 + If Spaces are enabled in your Catalogs, you can also view predefined or customized analytics information for your IBM API Connect Spaces within dashboards.

- **The default Overview dashboard**
  In IBM API Connect, each Catalog has its own default dashboard named Overview, which shows standard analytics information in the form of *visualizations*. The visualizations are represented as charts. The data depicted is retrieved from indexed (searchable) data for all API events, and is scoped to events that occur for the selected Catalog within the provider organization, for a specified time range.
- **Creating custom dashboards**
  You can create custom dashboards that group together a set of related visualizations.
- **Opening saved dashboards**
  You can open an existing dashboard if you want to view or edit the contents, or if you want to share it so that other users can view the contents.
- **Editing dashboards**
  You can edit the default dashboard or 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.
- **Sharing dashboards**
  You can share dashboards with other users who have an interest in the API event information displayed. You share a dashboard by sharing its link.
- **Exporting dashboards**
  You can export dashboards so they can be imported by other users, or into other Catalogs on your system. V5.0.5+ If Spaces are enabled in a Catalog, exported dashboards can also be imported into a Space.
- **Importing dashboards**
  You can import dashboards from other users for use, or you can import dashboards from another Catalog on your system. V5.0.5+ If Spaces are enabled in a Catalog, you can also import dashboards from 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.
- V5.0.5+ **Deleting custom dashboards**
  If a custom dashboard is no longer required, you can permanently delete it.
- V5.0.5+ **Restoring the default dashboards**
  If you have modified any of the default dashboards by adding, editing, removing, resizing, or repositioning visualizations, or by changing the time range for the visualization data, you can restore the original settings for those dashboards if required.
- **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.
- **Monitoring event fields**
  A monitoring event is logged approximately every 10 seconds on each management node, and every 5 minutes for each gateway node. The monitoring event record contains information about the status and health of the management node or gateway.
- **Audit event fields**
  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.
- **Log event fields**
  A log event is recorded each time the gateway encounters a problem. The log event record contains details about the problem that occurred with the gateway.
- **Viewing and exporting analytics and API event data**
  You can obtain analytics and API event data from the API Manager user interface or by using REST API calls.
- **Viewing notifications of activities and alerts**
  You can use the Notifications view to track all user activity in the API Manager user interface; for example, the creation of a Product, or the deletion of an API.

Related information

- Elastic
- Elasticsearch
- Kibana

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

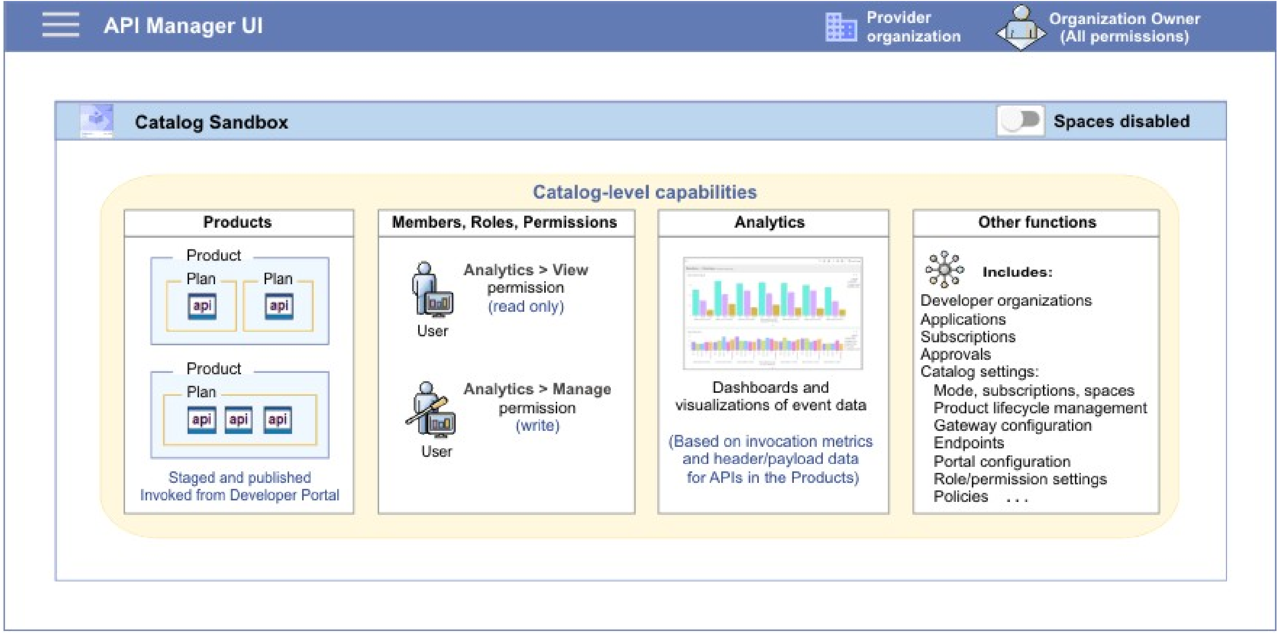
Analytics and syndication

The data for Analytics is collated from API events that are logged when API operations are invoked. 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 act as deployment targets through which APIs (in their containing Plans and Products) are staged and published to developer 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*, share dashboards with other users, 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 create, edit, delete, export, and import dashboards, create, edit, delete, export, and import the charts, tables, and maps (or *visualizations*) depicted in dashboards, and reset customized versions of the default dashboards and visualizations to their original forms if necessary.
  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.

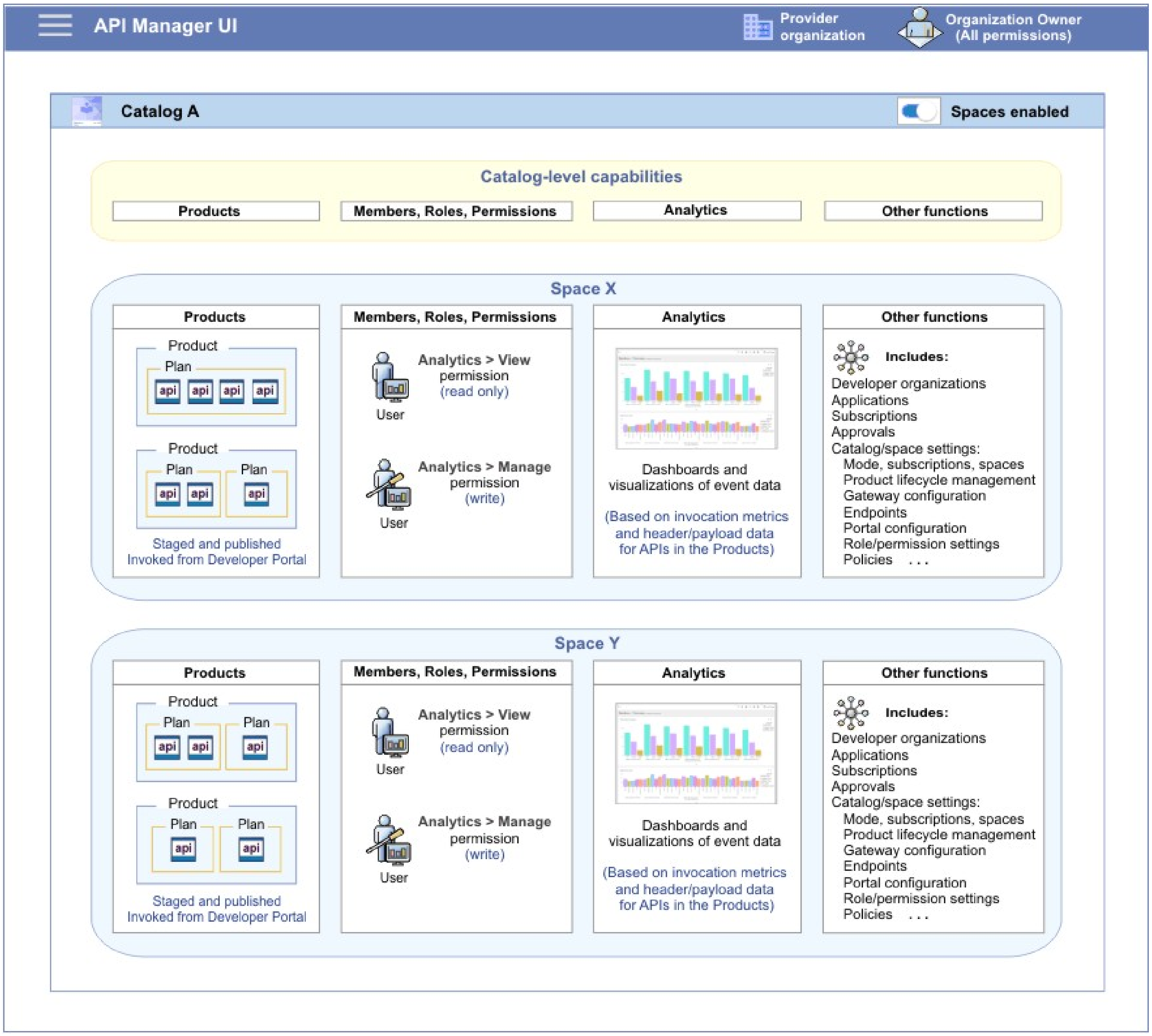


API Manager UI
Provider organization
Organization Owner (All permissions)
Catalog Sandbox
Spaces disabled
Catalog-level capabilities
Products
Product
Plan
Plan
api
api
Product
Plan
api
api
api
Staged and published
Invoked from Developer Portal
Members, Roles, Permissions
Analytics > View permission (read only)
User
Analytics > Manage permission (write)
User
Analytics
Dashboards and visualizations of event data
(Based on invocation metrics and header/payload data for APIs in the Products)
Other functions
Includes:
Developer organizations
Applications
Subscriptions
Approvals
Catalog settings:
Mode, subscriptions, spaces
Product lifecycle management
Gateway configuration
Endpoints
Portal configuration
Role/permission settings
Policies . . .

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, share dashboards with other users, 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 Space: These users have an implicit View permission. They can additionally create, edit, delete, export, and import dashboards, create, edit, delete, export, and import the charts, tables, and maps (or *visualizations*) depicted in dashboards, and reset customized versions of the default dashboards and visualizations to their original forms if necessary.
  For information about enabling Spaces in a Catalog, setting up Spaces, and assigning roles and permissions to a Space, see Using syndication in IBM 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.


API Manager UI
Provider organization
Organization Owner (All permissions)
Catalog A
Spaces enabled
Catalog-level capabilities
Products
Members, Roles, Permissions
Analytics
Other functions
Space X
Products
Product
Plan
api api api api
Product
Plan
Plan
api api
api
Staged and published
Invoked from Developer Portal
Members, Roles, Permissions
Analytics > View permission (read only)
User
Analytics > Manage permission (write)
User
Analytics
Dashboards and visualizations of event data
(Based on invocation metrics and header/payload data for APIs in the Products)
Other functions
Includes:
Developer organizations
Applications
Subscriptions
Approvals
Catalog/space settings:
Mode, subscriptions, spaces
Product lifecycle management
Gateway configuration
Endpoints
Portal configuration
Role/permission settings
Policies . . .
Space Y
Products
Product
Plan
Plan
api api
api
Product
Plan
Plan
api
api
Staged and published
Invoked from Developer Portal
Members, Roles, Permissions
Analytics > View permission (read only)
User
Analytics > Manage permission (write)
User
Analytics
Dashboards and visualizations of event data
(Based on invocation metrics and header/payload data for APIs in the Products)
Other functions
Includes:
Developer organizations
Applications
Subscriptions
Approvals
Catalog/space settings:
Mode, subscriptions, spaces
Product lifecycle management
Gateway configuration
Endpoints
Portal configuration
Role/permission settings
Policies . . .

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 and visualizations are added to the Catalog. If you enable Spaces in the Catalog, a similar set of default dashboards and visualizations are added to each Space. You can also create custom dashboards and visualizations in the Catalog and its Spaces if you have the required permission.
Note: At the Catalog level, an additional default dashboard is available for the Developer Portal (named `portal_default`), which is not provided at the Space level because only one Developer Portal is configured per Catalog.
A layered implementation is used to define an "inheritance" flow for dashboards and visualizations in a Catalog and its Spaces. This structure determines whether updates made to the dashboards and visualizations in a Catalog are reflected in a Space, and affects what you see when you attempt to edit, delete, or restore default dashboards or visualizations, or when you attempt to create, edit, or delete custom dashboards or visualizations.

V5.0.7+ To help you identify the state of a dashboard or visualization as default, custom, or inherited, tags are applied to dashboards and visualizations within the API Manager user interface. These tags are visible on the Manage Saved Objects page, which lists all your saved dashboards and visualizations, and from which you can manage dashboards and visualizations.

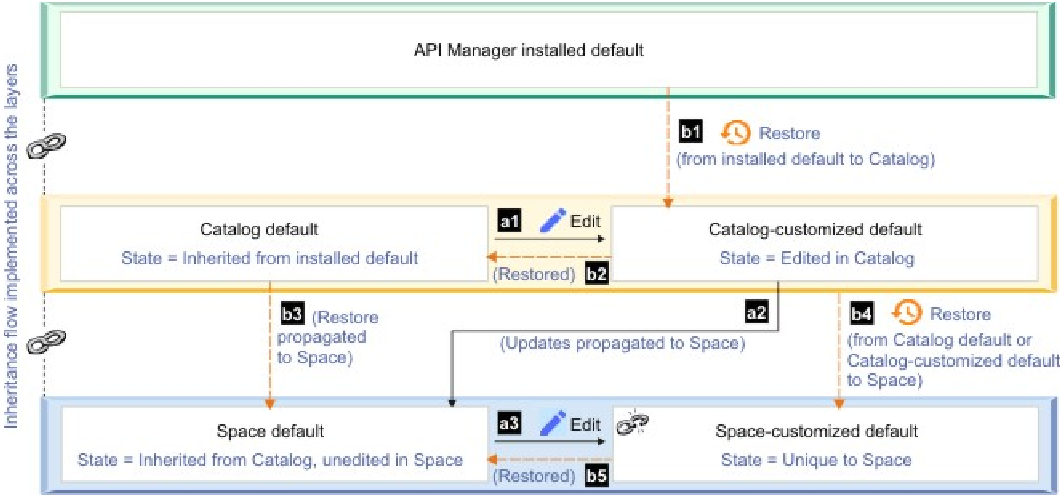
Default dashboards and visualizations

For the default dashboards and visualizations, the following principles apply for edit, delete, or restore operations:

- **Edit:** If you edit a default dashboard or visualization in a Catalog, the changes are automatically propagated to the corresponding default dashboard or visualization in the Spaces in that Catalog (depicted by a1 and a2 in the following diagram). Customizations to a default dashboard or visualization in a Catalog are cascaded to a Space until you edit the

dashboard or visualization within the Space, which then dissociates the dashboard or visualization in the Space from the one in the Catalog (depicted by a2 and a3). Any customizations to a default dashboard or visualization in a Space are reflected in that Space only (depicted by a3).
- **Delete:** The default dashboards and visualizations, whether in their original state or a customized state, cannot be deleted from a Catalog or Space.
- **Restore:** When you use the restore feature to reset a default dashboard or visualization that you previously customized, the restored dashboard or visualization in a Catalog recovers its settings from the API Manager defaults that were originally installed (depicted by b1 and b2). The changes are automatically propagated to the corresponding dashboard or visualization in the Spaces in that Catalog, provided the dashboard or visualization in a Space has not been independently modified within that Space (depicted by b3). If manually restored from a Space, the dashboard or visualization in that Space will recover its default settings from the current (original or customized) state of the corresponding dashboard or visualization in the Catalog (depicted by b4 and b5).
V5.0.5 ONLY Warning: The autosave feature causes dashboards to be automatically saved when you load or update them. It is important to note that loading a dashboard in a Space will cause that dashboard to be saved, thereby making it unique and breaking the inheritance flow. As a result, if you load a dashboard in a Space *before* you edit or restore the corresponding dashboard in the Catalog, you might see different results from what is documented.

The following diagram illustrates the inheritance flow that applies to the default dashboards or visualizations across Catalogs and Spaces:


API Manager installed default
Inheritance flow implemented across the layers
b1 Restore
(from installed default to Catalog)
Catalog default
State = Inherited from installed default
a1 Edit
(Restored) b2
Catalog-customized default
State = Edited in Catalog
b3 (Restore propagated to Space)
a2
(Updates propagated to Space)
b4 Restore
(from Catalog default or Catalog-customized default to Space)
Space default
State = Inherited from Catalog, unedited in Space
a3 Edit
(Restored) b5
Space-customized default
State = Unique to Space

Custom dashboards and visualizations

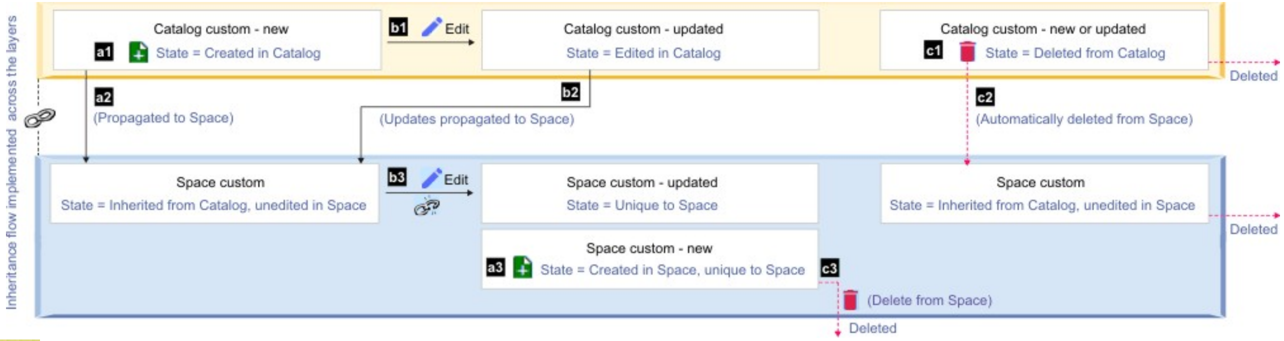
For custom dashboards and visualizations that you create, the following principles apply for create, edit, or delete operations:

- **Create:** When you create a dashboard or visualization in a Catalog, it is automatically added to the Spaces in that Catalog (depicted by a1 and a2 in the following diagram). If you create a dashboard or visualization in a Space, the dashboard or visualization is added to that Space only (depicted by a3).
- **Edit:** If you edit a custom dashboard or visualization in a Catalog, the changes are automatically propagated to the corresponding dashboard or visualization in the Spaces in that Catalog, provided the dashboard or visualization in a Space has not been independently modified within that Space (depicted by b1 and b2). If you edit an inherited custom dashboard or visualization in a Space, the changes are reflected in that Space only (depicted by b3).
- **Delete:** If you delete a custom dashboard or visualization from a Catalog, the corresponding dashboard or visualization in the Spaces in that Catalog are automatically deleted, provided the dashboard or visualization in a Space has not been independently modified within that Space (depicted by c1 and c2). If a dashboard or visualization was created in a Space and is unique to that Space, the dashboard or visualization is removed from the Space when deleted (depicted by c3).
V5.0.6 + If a dashboard or visualization in a Space was inherited from a Catalog, and has not been edited in the Space, you cannot delete that dashboard or visualization from the Space. You must delete it from the Catalog if required. Inherited dashboards or visualizations which have been updated in a Space also cannot be deleted from that Space, and must be deleted from the Catalog.

  V5.0.5 ONLY If a dashboard or visualization in a Space was inherited from a Catalog, and has not been edited in the Space, if you delete that dashboard or visualization from the Space, it is also deleted from the Catalog. If other Spaces in the Catalog contain an unedited version of the dashboard or visualization, it is also deleted from those Spaces.

  V5.0.5 ONLY Warning: The autosave feature causes dashboards to be automatically saved when you load or update them. It is important to note that loading a dashboard in a Space will cause that dashboard to be saved, thereby making it unique and breaking the inheritance flow. As a result, if you load a dashboard in a Space *before* you edit or delete the corresponding dashboard in the Catalog, you might see different results from what is documented.

The following diagram illustrates the inheritance flow that applies to the custom dashboards or visualizations across Catalogs and Spaces:


Inheritance flow implemented across the layers
Catalog custom - new
a1
State = Created in Catalog
b1
Edit
Catalog custom - updated
State = Edited in Catalog
Catalog custom - new or updated
c1
State = Deleted from Catalog
Deleted
a2
(Propagated to Space)
b2
(Updates propagated to Space)
c2
(Automatically deleted from Space)
Space custom
State = Inherited from Catalog, unedited in Space
b3
Edit
Space custom - updated
State = Unique to Space
Space custom
State = Inherited from Catalog, unedited in Space
Deleted
Space custom - new
a3
State = Created in Space, unique to Space
c3
(Delete from Space)
Deleted

V5.0.7+ How tags are applied to dashboards and visualizations in the Manage Saved Objects page

V5.0.7+ In the Manage Saved Objects page, which shows a listing of all your saved dashboards and visualizations and from which you can edit, delete, restore, export, and import dashboards and visualizations, tags are used to highlight the state of a dashboard or visualization as default, custom, or inherited:

- At the Catalog level, the following tags are applied to dashboards and visualizations in the list to indicate their state:
  - Each default dashboard or visualization, which has not been customized, is tagged with a `Default` label only.
  - Each default dashboard or visualization that has been customized is tagged with a `Default` label, and additionally tagged with a `Custom` label.
  - Each custom dashboard or visualization is tagged with a `Custom` label only.
- At the Space level, the following tags are applied to dashboards and visualizations in the list to indicate their state and inheritance settings:
  - Each default dashboard or visualization, which has not been customized, is tagged with a `Default` label only.
  - Each default dashboard or visualization that has been customized within that Space is tagged with a `Default` label, and additionally tagged with a `Custom` label.
  - Each default dashboard or visualization that was customized in a Catalog, inherited by its Spaces, and which has not been customized in that Space, is tagged with a `Default` label, and additionally tagged with an `Inherited` label.
  - Each default dashboard or visualization that was customized in a Catalog, inherited by its Spaces, and which has subsequently been customized within that Space, is tagged with a `Default` label, and additionally tagged with `Inherited` and `Custom` labels.
  - Each custom dashboard or visualization that was created in that Space is tagged with a `Custom` label only.
  - Each custom dashboard or visualization that was created in a Catalog, inherited by its Spaces, and which has not been customized in that Space, is tagged with an `Inherited` label only.
  - Each custom dashboard or visualization that was created in a Catalog, inherited by its Spaces, and which has subsequently been customized within that Space is tagged with `Inherited` and `Custom` labels.

For information about accessing the Manage Saved Objects page from the API Manager UI, see:

- Editing visualizations
- Exporting visualizations
- Importing visualizations
- Deleting custom visualizations
- Restoring the default visualizations
- Editing dashboards
- Exporting dashboards
- Importing dashboards
- Deleting custom dashboards
- Restoring the default dashboards

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Accessing analytics

You can view predefined or customized analytics information for your IBM® API Connect Catalogs within dashboards. V5.0.5+ If Spaces are enabled in your Catalogs, you can also view predefined or customized analytics information for your IBM API Connect Spaces within dashboards.

About this task

V5.0.5+ 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 will depend on your assigned roles and permissions. For more information about Catalogs and Spaces, see Working with Catalogs and Using syndication in IBM API Connect.

V5.0.7+ 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 has been switched off. Contact your cloud administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)

Procedure

- To open the analytics dashboard for any Catalog, complete the following steps:
  1. Open the API Manager Dashboard if it is not currently on display:
     a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
     b. To pin the UI navigation pane, click the Pin menu icon 📌 to change its state to "pinned" 📌.
     c. Click Dashboard.
  2. From the list of available Catalogs in the API Manager Dashboard, select the required Catalog.
     The Products tab is displayed with the list of Products in the Catalog.
  3. Click the Analytics tab.
     The default analytics dashboard for the Catalog is displayed.
- V5.0.5+ To open the analytics dashboard for any Space, complete the following steps:
  1. Open the API Manager Dashboard if it is not currently on display:
     a. If you have not previously pinned the UI navigation pane, click the Navigate to icon ☰ to open the API Manager UI navigation pane.
     b. To pin the UI navigation pane, click the Pin menu icon 📌 to change its state to "pinned" 📌.
     c. Click Dashboard.
  2. From the list of available Catalogs in the API Manager Dashboard, select the Catalog that contains the required Space.
  3. In the upper-left drop-down menu (below the IBM API Connect primary banner), select the Space that you want to work with.
     Tip: If you do not have access permission to the Products at the Catalog level or in a particular Space, a message is displayed if you try to access that Catalog or Space from the menu.
     The Products tab for the Space opens. This tab lists only those Products that are in the selected Space.
     Note: If an API is used in a single Space within a Catalog, the Space name is included in the analytics for that API. However, if the API is used in multiple Spaces within a Catalog, the API invoke must be associated with a Product plan to prevent ambiguity and ensure that the API can be included in analytics.
  4. Click the Analytics tab.
     The default analytics dashboard for the Space is displayed.

Related concepts

- The default Overview dashboard
- Analytics and syndication

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The default Overview dashboard

In IBM® API Connect, each Catalog has its own default dashboard named Overview, which shows standard analytics information in the form of *visualizations*. The visualizations are represented as charts. The data depicted is retrieved from indexed (searchable) data for all API events, and is scoped to events that occur for the selected Catalog within the provider organization, for a specified time range.

V5.0.5+ If Spaces are enabled in your Catalogs, each Space also has its own default dashboard (named Overview) with the same set of visualizations that are displayed for Catalogs. The data in the dashboard is scoped to events that occur for the selected Space in a Catalog, for a specified time range. For more information about Catalogs and Spaces, see Working with Catalogs and Using syndication in IBM API Connect.

The default Overview dashboard for Catalogs or Spaces displays the following charts, for a chosen time range, within a set of visualization containers:

- A bar chart of the five most active Products in the Catalog or Space. The y-axis shows the activity count, and the x-axis shows a daily distribution of activity count per Product. The colored bars that are used to identify individual Products are shown in descending order of average count across the time range.
- A bar chart of the five most active APIs in the Catalog or Space. The y-axis shows the count, and the x-axis shows a daily distribution of activity count per API. The colored bars that are used to identify individual APIs are shown in descending order of average count across the time range.

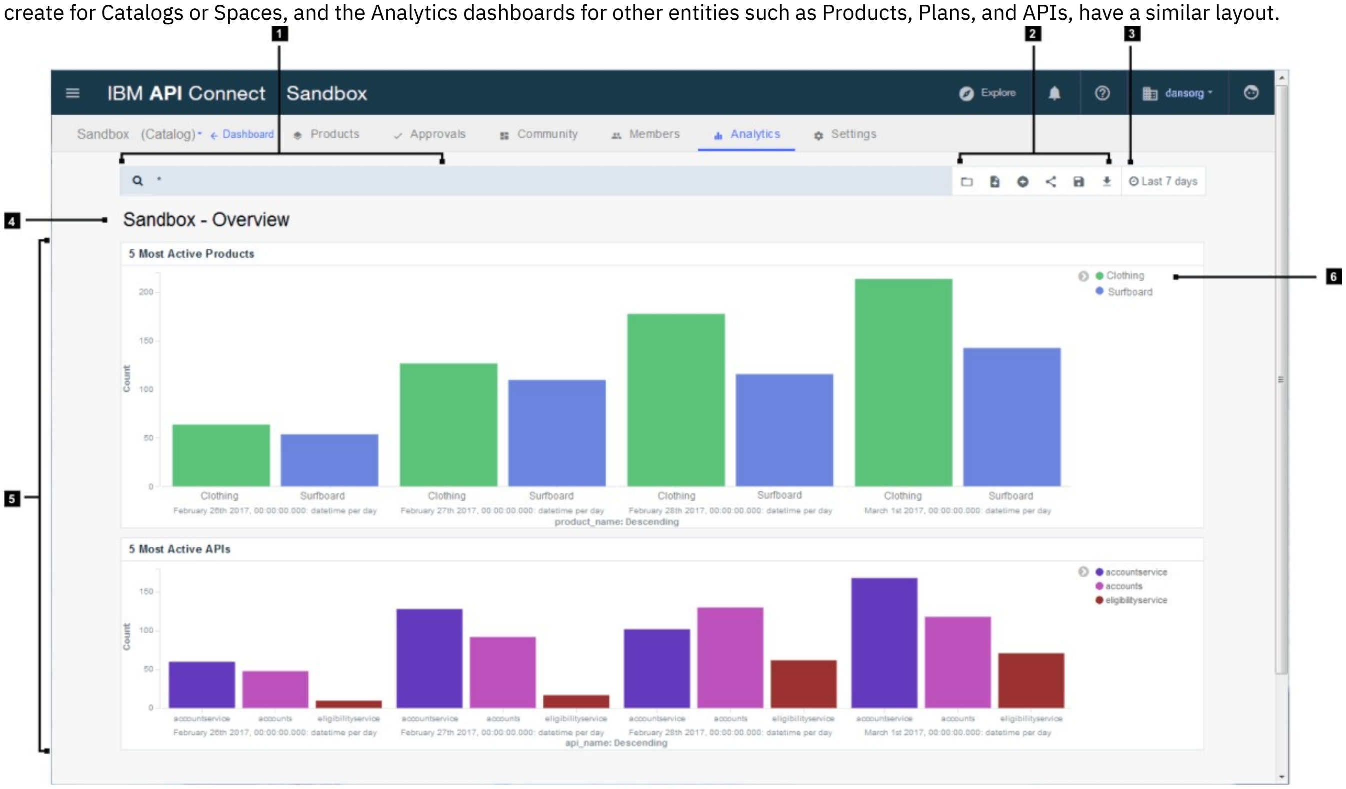
For a description of the screen elements of a dashboard, see:

- V5.0.7+ The screen elements of a dashboard
- V5.0.6 and earlier The screen elements of a dashboard

V5.0.7+

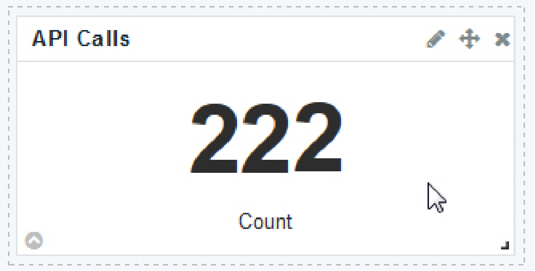
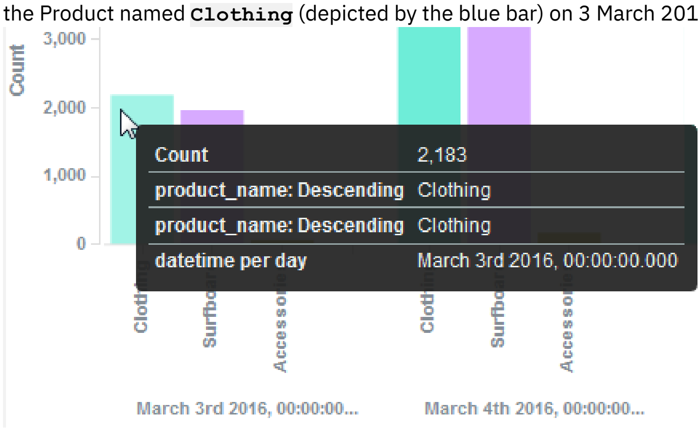
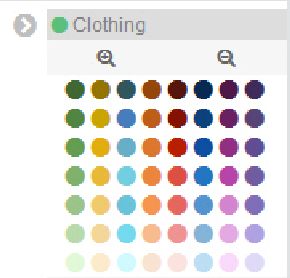
The screen elements of a dashboard

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.

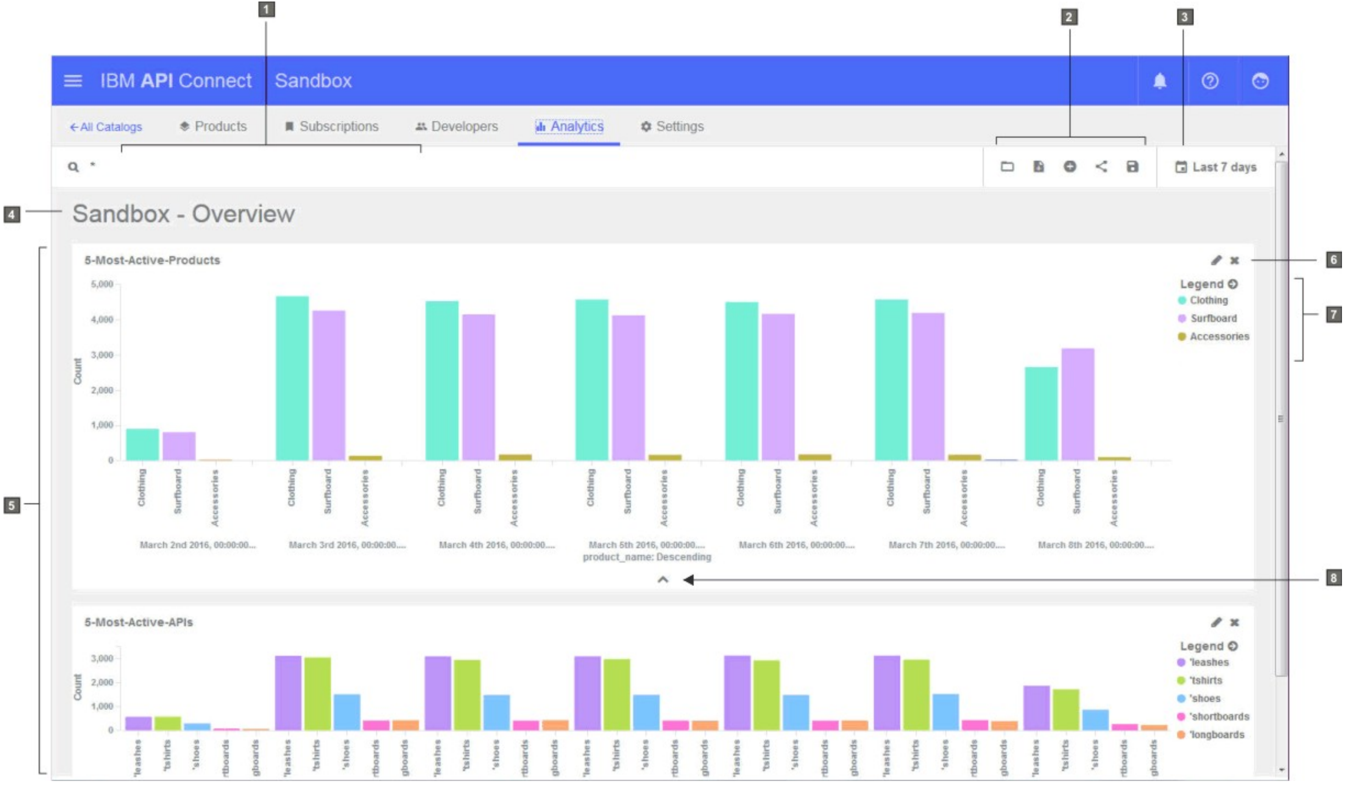
1 Search bar	Use this bar to specify search queries that apply additional filters to the data shown in the visualizations. For more information, see Filtering API events in your visualizations.
2 Dashboard configuration icons	Use these icons to open, create, share, or save dashboards, add visualizations to a dashboard, or export API event data for the dashboard to a CSV file. For more information, see: • Opening saved dashboards • Creating custom dashboards • Sharing dashboards • Editing dashboards • Exporting API event data from a dashboard
3 Time Picker icon	Use this icon to apply predefined, relative, or absolute time filters to the data shown in the dashboard, and to optionally set an auto-refresh interval for the data. For more information, see Specifying a time period and auto-refresh rate for the data in your visualizations.

4 Dashboard title	Displays the title of the default dashboard for each Catalog as ***`Catalog_name`* `- Overview`**. Custom dashboards for Catalogs are typically titled ***`Catalog_name - Custom_Dashboard_name`***. If Spaces are enabled in your Catalogs, the title of the default dashboard for each Space is displayed as ***`Space_name`* `- Overview`**.
5 Visualization area	Shows visualizations of your indexed data in chart form or as a numerical metric. Each visualization is displayed within a boxed area known as a *container*. When you hover over a container, you can see the following controls: - Icons in the upper right corner that enable you to edit the visualization, reposition the visualization in the dashboard, and remove the visualization from the dashboard. - An an icon in the lower right corner that enables you to resize the visualization. - A caret icon ⌃ in the lower left corner, with a toggle action that enables you to display the raw data behind the visualization or show the chart again. For more information about the raw data that is displayed for a visualization, see Viewing API event data from visualizations and Exporting analytics data from a visualization.  When you hover over any element in certain chart types (for example, a line or a bar in a line chart or vertical bar chart), a large black tooltip is displayed with details about that element. (You specify whether to show or hide these tooltips when you configure visualizations.) In the following example, the tooltip indicates that 2,183 calls were made to APIs in the Product named **`Clothing`** (depicted by the blue bar) on 3 March 2016.  Count 3,000 2,000 1,000 0 Count 2,183 product_name: Descending Clothing product_name: Descending Clothing datetime per day March 3rd 2016, 00:00:00.000 Clothin Surfboa Accesso Clothin Surfboa Accesso March 3rd 2016, 00:00:00... March 4th 2016, 00:00:00... Note: In the default Overview dashboard, you might notice that two entries are shown for the product_name field within tooltips in the first visualization, and that two entries are shown for the api_name field within tooltips in the second visualization. These entries are duplicated purely because a Terms aggregation was applied twice to these fields to sort data in the configured visualizations. Detailed information about aggregations is available in the Elasticsearch aggregations reference.
6 Legend	Available for certain visualization types only; for example, line charts, pie charts, or vertical bar charts. Click the arrow icon ➜ to hide the legend and increase your viewing area; click the arrow again to show the legend. You can also click the legend labels to display additional controls, as shown in the following example:  From the *color picker*, you can click any of the colored dots to change the color of the associated line, slice, or bar on a chart. You can also apply inclusion (or *positive*) filters and exclusion (or *negative*) filters by using the Positive Filter icon 🔍 and Negative Filter icon 🔍. For more information about applying these filters, see V5.0.7+ Applying a filter by using the legend in a visualization.

V5.0.6 and earlier

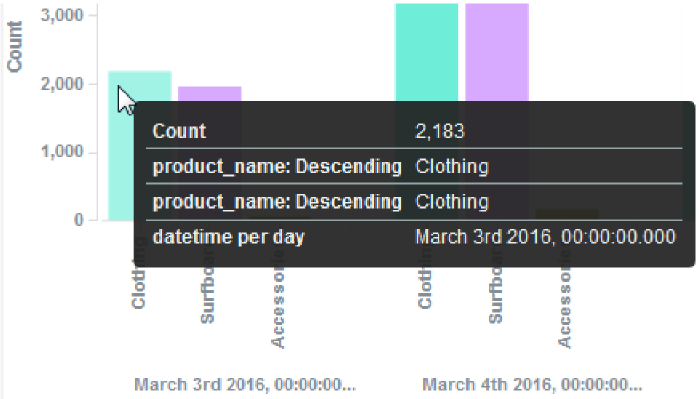
The screen elements of a dashboard

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.

1 Search bar	Use this bar to specify search queries that apply additional filters to the data shown in the visualizations.
2 Dashboard configuration icons	V5.0.3 and earlier Use these icons to open, create, share, or save dashboards, or to add visualizations to a dashboard. V5.0.4 V5.0.5 V5.0.6 Use these icons to open, create, share, or save dashboards, add visualizations to a dashboard, or export API event data for the dashboard to a CSV file.
3 Time Picker icon	Use this icon to apply predefined, relative, or absolute time filters to the data shown in the dashboard, and to optionally set an auto-refresh interval for the data.
4 Dashboard title	Displays the title of the default dashboard for each Catalog as ***Catalog_name* - Overview**. Custom dashboards for Catalogs are typically titled V5.0.4 and earlier ***Catalog_name* - *Custom_Dashboard_name*** or V5.0.5 V5.0.6 ***Custom_Dashboard_name***. V5.0.5 V5.0.6 If Spaces are enabled in your Catalogs, the title of the default dashboard for each Space is displayed as ***Space_name* - Overview**.

5 Visualization area	Shows visualizations of your indexed data in chart form or as a numerical metric. Each visualization is displayed within a *container*, which can be resized and repositioned in the dashboard. When you hover over any element in certain chart types, a large black tooltip is displayed with details about that element. (You specify whether to show or hide these tooltips when you configure visualizations.) In the following example, the tooltip indicates that 2,183 calls were made to APIs in the Product named `Clothing` (depicted by the blue bar) on 3 March 2016.  Note: In the default Overview dashboard, you might notice that two entries are shown for the product_name field within tooltips in the first visualization, and that two entries are shown for the api_name field within tooltips in the second visualization. These entries are duplicated purely because a Terms aggregation was applied twice to these fields to sort data in the configured visualizations. Detailed information about aggregations is available in the Elasticsearch aggregations reference.
6 Visualization icons	Use these icons to edit a visualization or to remove it from the dashboard.
7 Legend	Available for certain visualizations only. Use the arrow icon ➔ to show or hide the legend, and use the legend symbols to apply filters.
8 Caret	Use the caret (^) to display the raw data behind the visualization. Click the caret again to show the chart. For more information about the raw data that is displayed, see Viewing API event data from visualizations and the *Exporting analytics data from a visualization* section in Exporting analytics and API event data to CSV files.

Related tasks

- Accessing analytics

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating custom dashboards

You can create custom dashboards that group together a set of related visualizations.

Before you begin

V5.0.5 + 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

V5.0.5 + 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 Analytics and syndication.

V5.0.4 and earlier V5.0.6 + 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.

V5.0.5 ONLY When you create a dashboard, you can either add previously saved visualizations, or you can create visualizations as part of the process, and then arrange and resize them as required in the dashboard.

Important: If the Elasticsearch index that stores the definitions of your custom dashboard becomes corrupted and reverts back to its initial settings, it is possible that your custom dashboards might become inaccessible. This can happen when a split-brain situation occurs, for example. To prevent the permanent loss of your custom dashboard definitions, you can create backup copies of their definitions by using the Export function in the Manage Visualizations menu of each dashboard.

Procedure

To create a custom dashboard, complete the following steps:

1. From a default or custom dashboard, click the New Dashboard icon to open an empty dashboard.
2. V5.0.4 and earlier V5.0.6 + To add visualizations to this dashboard, complete the following steps:
   a. Click the Choose from existing visualizations link in the dashboard. Or, click the Add Visualization icon ⊕.
   The list of saved visualizations is displayed.
   b. Add one or more saved visualizations to the dashboard as follows:
      i. From the saved list, 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 in a container in the dashboard, beneath the saved list. Depending on the number of visualizations 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.
3. V5.0.5 ONLY To add visualizations to this dashboard, complete either of the following steps:
   - Add one or more saved visualizations to the dashboard:
     - Click the Choose from existing visualizations link in the dashboard. Or, click the Add Visualization icon ⊕.
     - From the saved list, 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 in a container in the dashboard, beneath the saved list.

     - Select any other visualizations that you want to add.
   - Create, and then add a new visualization to the dashboard:
     - If you have not yet added any visualizations to the dashboard, click the Create a visualization link in the dashboard. If you have added one or more existing visualizations to the dashboard, click Create visualizations from the open saved list.
     - From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations.
     When you close the "Configure visualization" page, the visualization is shown in the new dashboard being created.

   Tip: To add more visualizations to the dashboard after creating and adding a visualization, click the Add Visualization icon ⊕. From the saved list, either select existing visualizations to add to the dashboard, or click Create visualizations to create and add new visualizations to the dashboard.
4. V5.0.7 + Close the list of saved visualizations by clicking the Add Visualization icon ⊕.

   V5.0.6 and earlier If necessary, close the list of saved visualizations by clicking the caret icon ^.
5. V5.0.7 + 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 corner when you hover over a container.
   V5.0.6 and earlier Arrange the visualizations in the required order by using the container headers to drag and drop the visualizations in the preferred location in the dashboard.

6. Resize the visualization containers by dragging outwards or downwards from the lower right 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 7 days filter is applied. You can specify a different filter by using the Time Picker icon (V5.0.7 + V5.0.6 and earlier), as described in Applying filters to change the sampling of data displayed in visualizations.
8. To save the new dashboard, complete the following steps:
   a. Click the Save Dashboard icon .
   b. Specify a name for the dashboard in the text field.
   V5.0.4 and earlier V5.0.7 + On saving, the *Custom_Dashboard_name* name that you specify will be automatically appended to the Catalog name to create this title for the new dashboard: ***`Catalog_name - Custom_Dashboard_name`***.

V5.0.5 V5.0.6 On saving, the *Custom_Dashboard_name* name that you specify will be used as the title for the new dashboard.

c. Select the Store time with dashboard check box if you want to save the dashboard with the time period that you specified in step 7.
d. 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.

- **Creating and managing 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. When you create a visualization, you can choose how the data is to be presented.

Related tasks

- Opening saved dashboards
- Editing dashboards
- Sharing dashboards

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating and managing 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. When you create a visualization, you can choose how the data is to be presented.

Visualizations are stored in a saved list and can be combined to build dashboards with correlated information. Visualizations can also be exported and imported for sharing, and those that you no longer require can be deleted.

You can configure the following visualization types:

- Area chart
- Data table
- Line chart
- Markdown widget
- Metric
- Pie chart
- Tile map
- Vertical bar chart
- V5.0.7+ Tag cloud

- **Creating visualizations**
  You can create visualizations from the default dashboards or from custom dashboards. All visualizations that you create are added to a list of saved visualizations, and can be used in any dashboard.
- **Editing visualizations**
  You can edit a visualization either from a dashboard that contains that visualization or by managing your saved objects.
- **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 visualizations.
- **Exporting visualizations**
  You can export visualizations so they can be imported by other users, or into other Catalogs on your system. V5.0.5+ If Spaces are enabled in a Catalog, exported visualizations can also be imported into a Space.
- **Importing visualizations**
  You can import visualizations from other users for use in your dashboards, or you can import visualizations from another Catalog on your system. V5.0.5+ If Spaces are enabled in a Catalog, you can also import visualizations from 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.

- V5.0.5+ **Deleting custom visualizations**
  If no longer required for use in your dashboards, you can permanently delete your custom visualizations.
- V5.0.5+ **Restoring the default visualizations**
  If you have modified any of the default visualizations, you can restore the original settings for those visualizations if required.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating visualizations

You can create visualizations from the default dashboards or from custom dashboards. All visualizations that you create are added to a list of saved visualizations, and can be used in any dashboard.

Before you begin

V5.0.5+ 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

V5.0.5+ 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 Analytics and syndication.

Procedure

To create a visualization, complete any of the following steps from a dashboard:

- Create the visualization from a new dashboard page:
  1. Click the New Dashboard icon .
  2. Click the Add Visualization icon ⊕. Then, click Create visualizations.
     V5.0.5+ Alternatively, click the Create a visualization link in the dashboard.
  3. From the "Create New Visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations.
     V5.0.6 and earlier From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations.

     You return to the new dashboard page. V5.0.5 ONLY The visualization is shown in the new dashboard page.
- V5.0.4 and earlier V5.0.6+ Create the visualization from any open dashboard:
  1. Click the Add Visualization icon ⊕.
  2. Click Create visualizations.
  3. From the "Create New Visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations. When you close the visualization builder page, you return to the previous dashboard that was on display.
     V5.0.6 and earlier From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations. When you close the "Configure visualization" page, you return to the previous dashboard that was on display.
- V5.0.5 ONLY Create the visualization while creating a dashboard:
  1. Click the New Dashboard icon .
  2. Click the Create a visualization link in the dashboard.
  3. From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations.
     When you close the "Configure visualization" page, the visualization is added to the new dashboard being created.
- V5.0.5 ONLY Create the visualization while editing a dashboard:

1. Load the dashboard to which you want to add a new visualization.
2. Click the Add Visualization icon ⊕.
3. Click Create visualizations.
4. From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations.
   When you close the "Configure visualization" page, you return to the previous dashboard that was on display. The new visualization is added at the bottom of the 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring visualizations

Use this information either when creating a visualization or when editing an existing visualization.

Before you begin

V5.0.7+ As a starting point for this task, you must have completed one of the following actions:

- Clicked the Create a visualization link in a new dashboard page to open the Create New Visualization page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Add Visualization icon ⊕ from a new dashboard page, and then clicked Create Visualizations to open the Create New Visualization page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Add Visualization icon ⊕ from an existing dashboard, and then clicked Create Visualizations to open the Create New Visualization page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Edit icon ✎ in the upper right corner of a visualization container in an existing dashboard, to open the visualization builder page. (In this case, start from step 2 in the procedure that follows.)
- Clicked the Edit button ✎ for a visualization in the Manage Saved Objects page, to open the visualization builder page. (In this case, start from step 2 in the procedure that follows.)

V5.0.6 and earlier As a starting point for this task, you must have completed one of the following actions:

- V5.0.5 V5.0.6 Clicked the Create a visualization link in a new dashboard page to open the "Select a visualization " page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Add Visualization icon ⊕ from a new dashboard page, and then clicked Create visualizations to open the "Select a visualization " page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Add Visualization icon ⊕ from an existing dashboard, and then clicked Create visualizations to open the "Select a visualization " page. (In this case, start from step 1 in the procedure that follows.)
- Clicked the Edit icon ✎ in the upper right corner of a visualization container in an existing dashboard, to open the "Configure visualization" page. (In this case, start from step 3 in the procedure that follows.)
- Clicked the Edit icon ✎ for a visualization in the Manage Saved Objects page, to open the "Configure visualization" page. (In this case, start from step 3 in the procedure that follows.)

V5.0.5 + 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. V5.0.7+ From the Create New Visualization page, choose the type of visualization that you want to create:
   V5.0.6 and earlier From the "Select a visualization" page, choose the type of visualization that you want to create:
   - Area chart
   - Data table
   - Line chart
   - Markdown widget
   - Metric
   - Pie chart
   - Tile map
   - Vertical bar chart
   - V5.0.7+ Tag cloud

   Tip: If necessary, review the guidance on the page to help you decide what type you need.
2. V5.0.7+ 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 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 4) after the initial metric or bucket aggregations are configured, and then save in stages as you configure more aggregations for the visualization.
3. V5.0.6 and earlier From the "Configure visualization" page, define the content and layout for the visualization you are creating or editing. You can configure your settings by using the aggregation builder on the left, and then view the results of your actions in the preview canvas on the right.
   a. From the Data tab in the aggregation builder, configure the metric and bucket aggregations for the visualization.

b. From the Options tab in the aggregation builder, specify view options for the visualization.

For full details about completing the aggregation builder for your selected visualization, see the following information in the Kibana documentation:

- Area Charts
- Data Table
- Line Charts
- Markdown Widget
- Metric
- Pie Charts
- Tile Maps
- Vertical Bar Charts

For additional information about aggregations, see the Aggregation Builder section in the Kibana documentation.

As you configure each setting in the aggregation 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 4) after the initial metric or bucket aggregations are configured, and then save in stages as you configure more aggregations for the visualization.

4. When the configuration is complete, save the visualization:
   a. Click the Save Visualization icon 💾.
   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.)
5. V5.0.7 + Click the Close button to close 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.
6. V5.0.6 and earlier Click the Close icon ✕ to close the "Configure visualization" page.
   You return to the previous dashboard.
   - V5.0.4 and earlier V5.0.6 + If you created a visualization, it is added to the saved list.
   - V5.0.5 ONLY If you created a visualization from an existing dashboard or while creating a dashboard, the visualization is added to the dashboard and 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

V5.0.5 + 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

V5.0.5 + 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 Analytics and syndication.

V5.0.4 and earlier All dashboards that contain the modified visualization will be updated to reflect the changes made for that visualization.

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. V5.0.7 + 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 corner of the container. The visualization builder page opens.
       V5.0.6 and earlier From a dashboard that contains the visualization you want to edit, locate the visualization and then click the Edit icon ✎ in the upper right corner of the container. The "Configure visualization" page opens.
    2. Update and save the visualization configuration as described in Configuring visualizations.
       V5.0.7 + When you close the visualization builder page, you return to the dashboard and can see the updated visualization.
       V5.0.6 and earlier When you close the "Configure visualization" 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 the Add Visualization icon ⊕.
       The list of saved visualizations is displayed.
    2. V5.0.7 + Click Manage Visualizations to open the Manage Saved Objects page.
       V5.0.6 and earlier 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. V5.0.7 + From the Visualizations tab, locate the visualization you want to edit.
       V5.0.6 and earlier 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. V5.0.7 + Click the Edit button ✎ for the visualization. The visualization builder page opens.
       V5.0.6 and earlier Click the Edit icon ✎ for the visualization. The "Configure visualization" page opens.
    5. Update and save the visualization configuration as described in Configuring visualizations.
       V5.0.7 + 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.

       V5.0.6 and earlier When you close the "Configure visualization" 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 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
- V5.0.7 + Applying a filter by using the legend in a visualization
- V5.0.6 and earlier 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 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: **`field_name:value`** Field names and their values are case-sensitive.	**`app_name:legacyquote`**
Where required for numeric fields, use comparison operators such as greater than (>), less than (<), or equal to (=).	**`rateLimit.count:>10`**

Use the logic operators AND, OR, and NOT to combine search terms.	**`app_name:livequote OR`**
	`app_name:legacyquote`

      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: V5.0.7+ 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 corner when you hover over the container) to display the raw data behind the visualization. V5.0.6 and earlier Two quick ways to check for the field names on certain visualizations are to check the tooltip, or to click the caret icon ⌃ at the bottom of 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 dashboard.

Procedure

To apply a time filter and auto-refresh rate, complete the following steps:

1. V5.0.7+ From the dashboard, click the Time Picker icon 🕘.
   V5.0.6 and earlier 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 above 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 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.
   Attention: The usage of frequent auto-refresh has been observed to impact CPU usage on the server. If you observe issues of this nature, consider reducing the frequency of auto-refresh or disabling this setting, and monitor for any impact.

4. V5.0.7+ 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 of the Time Picker icon 🕘. If set, the auto-refresh interval is shown to the left of the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

   V5.0.6 and earlier Close the Time Picker by clicking the caret ^ at the bottom of the panel. The search query is resubmitted 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 of the Time Picker icon 📅. If set, the auto-refresh interval is shown to the left of the Time Picker icon, together with a Pause icon that can be used to pause auto-refresh if required.

   Tip: V5.0.7+ 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. V5.0.6 and earlier To switch off the auto-refresh capability, click the auto-refresh value to the left of the Time Picker icon, and click Off.

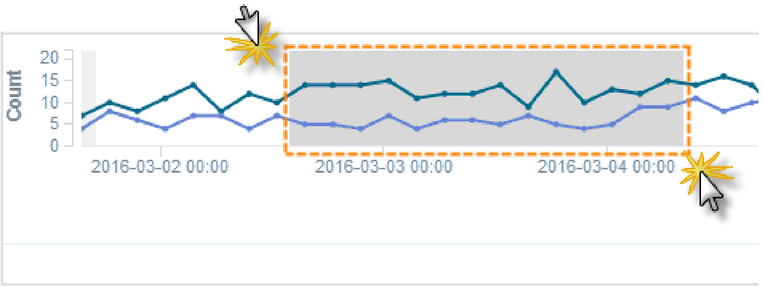
Drilling down into data in your time-based visualizations

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 of the Time Picker icon (V5.0.7+ 🕘 V5.0.6 and earlier 📅).

  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.
- V5.0.7+ 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.
- V5.0.6 and earlier Bar charts only: Click one of the bars in the chart to zoom in.

V5.0.7+

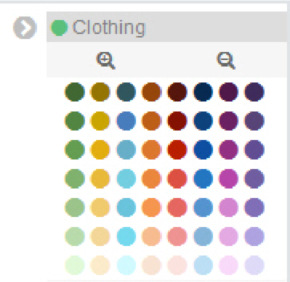
Applying a filter by using the legend in a visualization

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

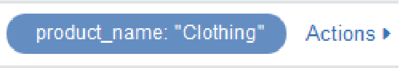
To apply a filter from the legend of a chart such as an area chart, line chart, pie chart, or vertical bar chart, complete the following steps:

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.



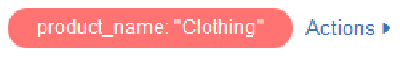
Tip: You can change the color of the lines, slices, or bars on a chart by clicking the colored dots in the color picker to select a preferred color. To retain these colors, you will have to save the dashboard.

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 below 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.

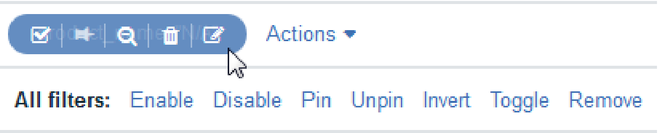
     product_name: "Clothing" Actions ▸

   - 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 below 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.



Actions ▾

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 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 *not* 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 product_name: "Clothing".	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 Edit Filter section within the *Filtering by Field* 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`.

▸V5.0.6 and earlier

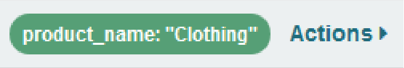
Applying a filter by using the legend in a visualization

For any chart that includes a legend, you can use the legend symbols to apply *inclusion* 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 further apply other filter conditions.

Procedure

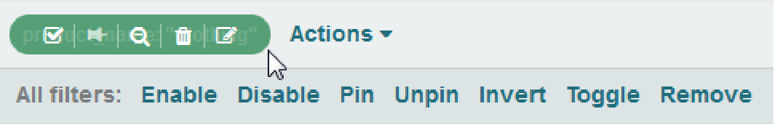
To apply a filter from the legend of a chart, complete the following steps:

1. In the chart, click a legend symbol to filter for that item only.
   A green filter oval and an Actions twistie are displayed below 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 ▸

2. 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 and also shows the Actions twistie in an expanded state. The Actions twistie provides equivalent options for the icons.

Icon	Description	Equivalent Actions option
☑	Click this icon to temporarily disable the inclusion filter that was set in step 1. When you disable an inclusion filter, the filter oval is shown in a green 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.	Enable Disable Toggle is an alternative for Enable/Disable.
📌	Click this icon to pin the filter. A pin icon is shown to the left of the filter condition in the green oval as an indication. Pinning a filter causes it to be applied to other dashboards that you open.	Pin Unpin
🔍	Click this icon to toggle the filter to instead show excluded items (that is, those items that do *not* match the filter condition). The toggle action switches between applying an inclusion filter (the default) and an exclusion filter to the data in the visualization. When the exclusion filter is applied, the filter oval switches from green to red product_name: "Clothing" .	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 customize the JSON code and then specify an alias that can be used as the filter name. To see examples of how to use this option, see the description for Custom Filter within the *Working with Filters* section of 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Exporting visualizations

You can export visualizations so they can be imported by other users, or into other Catalogs on your system. V5.0.5 + 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

V5.0.5 + 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 the Add Visualization icon ⊕.
   The list of saved visualizations is displayed.
2. V5.0.7+ Click Manage Visualizations to open the Manage Saved Objects page.
   V5.0.6 and earlier 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. V5.0.7+ 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 the Export button.
   - To export several visualizations, locate each visualization and select its check box. Then click the Export button.
   - To export all visualizations, select the Select All check box and then click the Export button.
     Note: You can also collectively export all dashboards and visualizations by clicking the Export Everything button.

   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.
4. V5.0.6 and earlier 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 the Export Selected icon ➔.
   - To export several visualizations, locate each visualization and select its check box. Then click the Export Selected icon ➔.
   - To export all visualizations, select the Select All check box and then click the Export All icon ➔ that is positioned to the right of the page title.
     Note: When you choose to export all visualizations, you also export all saved dashboards.

   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.
5. 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.
6. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Related tasks

- Importing visualizations

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Importing visualizations

You can import visualizations from other users for use in your dashboards, or you can import visualizations from another Catalog on your system. V5.0.5+ 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

V5.0.5+ 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 the Add Visualization icon ⊕.
   The list of saved visualizations is displayed.
2. V5.0.7+ Click Manage Visualizations to open the Manage Saved Objects page.
   V5.0.6 and earlier 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. V5.0.7+ The Visualizations tab is displayed by default. V5.0.6 and earlier The visualizations tab is displayed by default.

3. V5.0.7+ Click the Import button.
   V5.0.6 and earlier Click the Import icon ➔.
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, confirm the overwrite when prompted.
   V5.0.7+ All visualizations in the .json file are added as individual objects to the Visualizations tab.

   V5.0.5 V5.0.6 All visualizations in the .json file are added as individual objects to the visualizations tab.

   V5.0.4 and earlier All visualizations in the .json file are added as individual objects to the visualizations tab and will be visible when you next access the Manage Saved Objects page.
5. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Related tasks

- Exporting visualizations

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Deleting custom visualizations

If no longer required for use in your 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.

V5.0.6 + 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.

For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in Analytics and syndication.

Procedure

To delete one or more visualizations, complete the following steps:

1. From a dashboard, click the Add Visualization icon ⊕.
   The list of saved visualizations is displayed.
2. V5.0.7 + Click Manage Visualizations to open the Manage Saved Objects page.
   V5.0.5 V5.0.6 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. V5.0.7 + 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 button.
   - To delete several visualizations, locate each visualization and select its check box. Then click the Delete button.

   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 `Default` label. Default visualizations that have been customized are additionally tagged with a `Custom` label. If working within a Space, default visualizations that were inherited from a Catalog are also tagged with an `Inherited` label. Any visualization that is tagged with a `Default` label cannot be deleted.
   - Each custom visualization is tagged with a `Custom` label. If working within a Space, custom visualizations that were inherited from a Catalog are also tagged with an `Inherited` label. Any visualization that is tagged with an `Inherited` label cannot be deleted.
4. V5.0.5 V5.0.6 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 Selected icon 🗑.
   - To delete several visualizations, locate each visualization and select its check box. Then click the Delete Selected icon 🗑.

   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.
5. 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.
6. V5.0.7 + Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Restoring the default visualizations

If you have modified any of the default visualizations, you can restore the original settings for those visualizations if required.

Note: This action cannot be reversed; so consider whether you want to save a backup of the modified visualization (using a different name) before you reset its defaults.

Before you begin

To restore a default visualization, 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

A visualization in a Catalog is restored to the API Manager installed defaults. If Spaces are enabled in the selected Catalog, a restore action at the Catalog level is automatically propagated to the corresponding visualization in the Spaces provided the visualization in a Space has not been independently modified within that Space. A visualization that is restored from a Space is reset to the current (installed or customized) defaults at the Catalog level. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in Analytics and syndication.

Procedure

To restore a default visualization, complete the following steps:

1. From a dashboard, click the Add Visualization icon ⊕.
   The list of saved visualizations is displayed.
2. V5.0.7+ Click Manage Visualizations to open the Manage Saved Objects page.
   V5.0.5 V5.0.6 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. V5.0.7+ From the Visualizations tab, locate the default visualization that you want to restore.
   Tips:
   - If you have a large set of saved visualizations, you can either browse through the list or use the filter to locate the visualization.
   - Each default visualization is tagged with a `Default` label, and each custom visualization is tagged with a `Custom` label.
   - Each default visualization that has been customized (and which can be restored) is additionally tagged with a `Custom` label.
   - Each default visualization that can be restored also has an associated Restore button ↺ in the list.
4. V5.0.5 V5.0.6 From the visualizations tab, locate the default visualization that you want to restore.
   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.
   - Each default visualization that can be restored has an associated Restore icon ↺ in the list.
5. V5.0.7+ Click the Restore button ↺ for the visualization. The Restore button ↺ and the `Custom` label for the visualization are removed.
   V5.0.5 V5.0.6 Click the Restore icon ↺ for the visualization. A confirmation message is displayed in the notification banner and the Restore icon ↺ for the visualization is removed.
6. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Opening saved dashboards

You can open an existing dashboard if you want to view or edit the contents, or if you want to share it so that other users can view the contents.

Procedure

To open a dashboard, complete the following steps:

1. From any other dashboard, click the Load Saved Dashboard icon 🗀.
   The list of saved dashboards is displayed.
2. From the saved list, select the required dashboard.
   Tip: If you have a large set of saved dashboards, you can either browse through the saved list or use the dashboard filter to find the dashboard.
   The dashboard is displayed.

Related tasks

- Editing dashboards
- Sharing dashboards

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Editing dashboards

You can edit the default dashboard or 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

V5.0.5 + 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

V5.0.5 + 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 Analytics and syndication.

V5.0.4 and earlier V5.0.6 + 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.

V5.0.5 ONLY You can either add a saved visualization, or create and add a new visualization while editing the dashboard. Dashboards include an autosave feature, which will cause any changes you make to be automatically saved.

Procedure

To edit a dashboard, complete the following steps:

1. Access the dashboard that you want to edit in either of the following ways:
   - Open the dashboard as described in Opening saved dashboards.
   - Access the dashboard by managing your saved objects:
     - From a dashboard, click the Load Saved Dashboard icon 🗀. The list of saved dashboards is displayed.
     - Click Manage dashboards 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.
     - V5.0.7 + From the Dashboards tab, locate the dashboard you want to edit.
       V5.0.6 and earlier From the dashboards tab, locate the dashboard you want to edit.
       Tip: If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboard.
     - V5.0.7 + Click the Edit button ✎ for the dashboard. The dashboard opens.
       V5.0.6 and earlier Click the Edit icon ✎ for the dashboard. The dashboard opens.
2. Modify the contents as required:

- Add an existing visualization to the dashboard:
  - Click the Add Visualization icon ⊕. 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.
  - V5.0.7+ Close the list of saved visualizations by clicking the Add Visualization icon ⊕ again.

    V5.0.6 and earlier Close the list of saved visualizations by clicking the caret ^ at the bottom of the panel.
- V5.0.5 ONLY Create and add a new visualization to the dashboard:
  - Click the Add Visualization icon ⊕ to display the list of saved visualizations.
  - Click Create visualizations.
  - From the "Select a visualization" page, choose a visualization type, and then configure and save it as described in Configuring visualizations. The visualization is added to the dashboard and to the saved list.
- 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.)
  - V5.0.7+ Hover over the visualization and then click the Edit icon ✎ that is displayed in the upper right corner of the visualization container. The visualization builder page opens.
    V5.0.6 and earlier Click the Edit icon ✎ in the upper right corner of the visualization container. The "Configure visualization" page opens.
  - Update and save the visualization configuration as described in Configuring visualizations.
    V5.0.7+ When you close the visualization builder page, you return to the dashboard being edited. V5.0.6 and earlier When you close the "Configure visualization" page, you return to the dashboard being edited.
- V5.0.7+ Remove a visualization from the dashboard by hovering over the visualization and then clicking the Delete icon ✖ that is displayed in the upper right corner of the visualization container.
  V5.0.6 and earlier Remove a visualization from the dashboard by clicking the Delete icon ✖ in the upper right 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 corner by dragging inwards or upwards to decrease the size, or dragging outwards or downwards to increase the size.
- V5.0.7+ 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 corner when you hover over a container.
  V5.0.6 and earlier Rearrange the visualizations by using the container header to drag and drop the visualizations in the required location in the dashboard.
- 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 (V5.0.7+ 🕘 V5.0.6 and earlier 📅), as described in Applying filters to change the sampling of data displayed in visualizations.

3. When your changes are complete, save the dashboard as follows:
   a. Click the Save Dashboard icon 💾.
   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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Sharing dashboards

You can share dashboards with other users who have an interest in the API event information displayed. You share a dashboard by sharing its link.

About this task

V5.0.4 and earlier The recipient of the link can use it to view the data visualizations in your dashboard, but cannot edit the dashboard. A recipient must be defined as a member in the provider organization, and will be required to enter their API Manager credentials to view the dashboard.

V5.0.5 + The recipient of the link can use it to view the data visualizations in your dashboard, and can edit the dashboard if they have the relevant permission. A recipient must be defined as a member in the provider organization, and will be required to enter their API Manager credentials to view the dashboard. If Spaces are enabled in the selected Catalog, the dashboard data from a shared link will be scoped to a Catalog or to a Space depending on the level of access. For more information about Catalogs and Spaces, see Working with Catalogs and Using syndication in IBM API Connect.

V5.0.8 + You can share one of two types of links when you share your dashboard. The *administrative permissions links* provide the interface options to modify the dashboard when those permissions are assigned to the recipient. The *view permissions links* provide the interface options that are intended for recipients with view permissions. The administrative options are not displayed on the user interface when it is a view permissions link. If a recipient who has view permissions opens the administrative permissions link, the administrative options are displayed but are not active.

Procedure

To share a link to a dashboard, complete the following steps:

1. Open the dashboard that you want to share, as described in Opening saved dashboards.
2. From the dashboard, click the Share icon <.
3. Optional: V5.0.8 + In version 5.0.8.2, or later, select the Switch to View Permissions links to display the links for recipients who are not administrators.
   Use this option when your recipients only have permissions to view your dashboards. The links that are provided by selecting this option do not display the administrative options on interface. If you are sending the link to administrators or to recipients with various levels of permissions, skip this step.
4. V5.0.7 + In the resulting panel, locate the Link section and then click the associated Copy hyperlink to copy the URL for the dashboard.
5. V5.0.6 and earlier In the resulting panel, select the entire multi-line `https://`*`my_host_name`*`...` URL that is shown under the Share a link section. Then, copy the URL.
6. Send the URL that you copied to any users with whom you want to share your dashboard. Ask these users to open a new browser tab or window, and then paste the link text in the address bar.
   After authenticating, these users should now be able to view your dashboard and its visualizations, and can refresh their browsers periodically to see any updates to the data.
   V5.0.8 + Remember: You must have administrative permissions and open the administrative permissions link to view the administrative options.
7. V5.0.7 + To collapse the panel, click the Share icon <.

   V5.0.6 and earlier To collapse the panel, either click the caret ^ at the bottom of the pane, or click the Share icon <.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Exporting dashboards

You can export dashboards so they can be imported by other users, or into other Catalogs on your system. V5.0.5 + 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

V5.0.5 + 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. From a dashboard, click the Load Saved Dashboard icon 🗀.
   The list of saved dashboards is displayed.
2. V5.0.7+ Click Manage Dashboards to open the Manage Saved Objects page.
   V5.0.6 and earlier Click Manage dashboards 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. V5.0.7+ 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 click the Export button.
   - To export several dashboards, locate each dashboard and select its check box. Then click the Export button.
   - To export all dashboards, select the Select All check box and then click the Export button.
     Note: You can also collectively export all dashboards and visualizations by clicking the Export Everything button.

   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.
4. V5.0.6 and earlier 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 click the Export Selected icon ➡.
   - To export several dashboards, locate each dashboard and select its check box. Then click the Export Selected icon ➡.
   - To export all dashboards, select the Select All check box and then click the Export All icon ➡ that is positioned to the right of the page title.
     Note: When you choose to export all dashboards, you also export all saved visualizations.

   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.
5. 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.
6. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Related tasks

- Importing dashboards

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Importing dashboards

You can import dashboards from other users for use, or you can import dashboards from another Catalog on your system. V5.0.5+ 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

V5.0.5+ 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. From a dashboard, click the Load Saved Dashboard icon 🗀.
   The list of saved dashboards is displayed.

2. V5.0.7+ Click Manage Dashboards to open the Manage Saved Objects page.
   V5.0.6 and earlier Click Manage dashboards 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. V5.0.7+ The Dashboards tab is displayed by default. V5.0.6 and earlier The dashboards tab is displayed by default.
3. V5.0.7+ Click the Import button.
   V5.0.6 and earlier Click the Import icon ➔.
4. 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, confirm the overwrite when prompted.
   V5.0.7+ All dashboards in the .json file are added as individual objects to the Dashboards tab.

   V5.0.5 V5.0.6 All dashboards in the .json file are added as individual objects to the dashboards tab.

   V5.0.4 and earlier All dashboards in the .json file are added as individual objects to the dashboards tab and will be visible when you next access the Manage Saved Objects page.
5. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Related tasks

- Exporting dashboards

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Deleting custom dashboards

If a custom dashboard is no longer required, you can permanently delete it.

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.

V5.0.6+ 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 Analytics and syndication.

Procedure

To delete one or more dashboards, complete the following steps:

1. From a dashboard, click the Load Saved Dashboard icon 🗀.
   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. V5.0.7+ Click Manage Dashboards to open the Manage Saved Objects page.
   V5.0.5 V5.0.6 Click Manage dashboards 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. V5.0.7+ 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 button.
   - To delete several dashboards, locate each dashboard and select its check box. Then click the Delete button .

   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 **`Default`** label. Default dashboards that have been customized are additionally tagged with a **`Custom`** label. If working within a Space, default dashboards that were inherited from a Catalog are also tagged with an **`Inherited`** label. Any dashboard that is tagged with a **`Default`** label cannot be deleted.
   - Each custom dashboard is tagged with a **`Custom`** label. If working within a Space, custom dashboards that were inherited from a Catalog are also tagged with an **`Inherited`** label. Any dashboard that is tagged with an **`Inherited`** label cannot be deleted.
4. V5.0.5 V5.0.6 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 Selected icon 🗑.
   - To delete several dashboards, locate each dashboard and select its check box. Then click the Delete Selected icon 🗑.

   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.
5. 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.
6. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5+

Restoring the default dashboards

If you have modified any of the default dashboards by adding, editing, removing, resizing, or repositioning visualizations, or by changing the time range for the visualization data, you can restore the original settings for those dashboards if required.

Note: This action cannot be reversed; so consider whether you want to save a backup of the modified dashboard (using a different name) before you reset its defaults.

Before you begin

To restore a default dashboard, 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

A dashboard in a Catalog is restored to the API Manager installed defaults. If Spaces are enabled in the selected Catalog, a restore action at the Catalog level is automatically propagated to the corresponding dashboard in the Spaces provided the dashboard in a Space has not been independently modified within that Space. A dashboard that is restored from a Space is reset to the current (installed or customized) defaults at the Catalog level. For more information, see the *Dashboards and visualizations: Understanding the inheritance flow across Catalogs and Spaces* section in Analytics and syndication.

Procedure

To restore a default dashboard, complete the following steps:

1. From a dashboard, click the Load Saved Dashboard icon 🗀. The list of saved dashboards is displayed.
2. V5.0.7+ Click Manage Dashboards to open the Manage Saved Objects page.
   V5.0.5 V5.0.6 Click Manage dashboards 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. V5.0.7+ From the Dashboards tab, locate the default dashboard that you want to restore.
   Tips:
   - If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboard.
   - Each default dashboard is tagged with a `Default` label, and each custom dashboard is tagged with a `Custom` label.
   - Each default dashboard that has been customized (and which can be restored) is additionally tagged with a `Custom` label.
   - Each default dashboard that can be restored also has an associated Restore button ↺ in the list.
4. V5.0.5 V5.0.6 From the dashboards tab, locate the default dashboard that you want to restore.
   Tip:
   - If you have a large set of saved dashboards, you can either browse through the list or use the filter to locate the dashboard.
   - Each default dashboard that can be restored has an associated Restore icon ⟲ in the list.
5. V5.0.7+ Click the Restore button ↺ for the dashboard. The Restore button ↺ and the `Custom` label for the dashboard are removed.
   V5.0.5 V5.0.6 Click the Restore icon ⟲ for the dashboard. A confirmation message is displayed in the notification banner and the Restore icon ⟲ for the dashboard is removed.
6. V5.0.7+ Click the Close button to close the page and return to the dashboard.
   Click the Close icon ✕ to close the page and return to the dashboard.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. The maximum payload that can be logged is 2 MB. If you anticipate larger payloads, add an Activity Log policy that prevents the payload from logged for either success or error responses.

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`.
- 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. For information about how to view event records that are generated for your APIs, see Viewing and exporting analytics and API event data. 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. V5.0.8+ Beginning with version 5.0.8.3, you can customize which of these fields are exported to ensure that you are only storing information that you need for your visualizations. For more information, see Configuring visualizations. V5.0.5+ 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.

Table 1. API event record fields

Field name	Type	Description
@timestamp	Date	A time stamp that records when the record was written by the Logstash data collection engine that feeds data into Elasticsearch.
@version	String	The most recent version of the record that was written by the Logstash data collection engine that feeds data into Elasticsearch.
api_id	String	The API identifier.
api_name	String	The name of the API.
api_version	String	The version number of the API.
app_id	String	The identifier for the registered application.
app_name	String	The name of the registered application.
V5.0.7+ app_type	V5.0.7+ String	V5.0.7+ The application type, with a value of `Production` or `Development`.
V5.0.7+ billing.amount	V5.0.7+ String	V5.0.7+ The amount that a customer is billed for a subscription to the Product.
V5.0.7+ billing.currency	V5.0.7+ String	V5.0.7+ The monetary currency that is used to determine the subscription fee.
V5.0.7+ billing.model	V5.0.7+ String	V5.0.7+ The type of billing plan that a customer is billed for a subscription to the Product.
V5.0.7+ billing.provider	V5.0.7+ String	V5.0.7+ The credit card processing partner that is billing for a subscription to the Product.
V5.0.7+ billing.trial_period_days	V5.0.7+ Integer	V5.0.7+ The number of days that a customer can use a billing plan without being charged.
bytes_received	Number	The number of bytes received on the inbound request.
bytes_sent	Number	The number of bytes sent on the outbound request.
catalog_id	String	The Catalog identifier.
catalog_name	String	The Catalog name.
V5.0.8+ client_id	V5.0.8+ String	V5.0.8+ 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 `X-Forwarded-For` header.
datetime	Date	A time stamp that records when the API was executed. The time stamp is always shown in coordinated universal time UTC.
debug	String	Debug information for the API call if `APIm-Debug` is set to `true`.
developer_org_id	String	The identifier for the developer organization that owns the application.
developer_org_name	String	The name of the developer organization that owns the application.
developer_org_title	String	The display title of the developer organization that owns the application.

Field name	Type	Description
V5.0.8 + endpoint_url	V5.0.8 + String	V5.0.8 + When the request failed, identifies the proxy or invoke target URL on which the request failed. It is not included with a successful request.
gateway_ip	String	The IP address of the gateway.
headers.*field_name*	String	A component of the header section of a message.
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.
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_id	String	The Product identifier.
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.
rateLimit.count	Number	The number of API calls made in the defined rate limit time window.
rateLimit.limit	Number	The maximum number of requests an application is allowed to make to the API during a specified time window.
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_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.
space_id	String	The Space identifier.
space_name	String	The Space name.
status_code	String	The status code set on the outbound response.
tags	String	Tags related to the API event creation.
time_to_serve_request	Number	The round-trip time (in milliseconds) from when the Gateway receives the API request until it returns a response to the caller, including the time spent waiting for a response to an API call to the external endpoint. The amount of time can be affected by other factors, such as custom Gateway scripts; it is not just the amount of time required by the external endpoint for processing the request. The value of this field is calculated by the gateway and passed to Analytics.
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)

V5.0.7+

Restriction: The `billing` object fields are only supported in Version 5.0.7.2, and later.

```
{
  "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",
  "billing": {
     "trial_period_days": "0",
     "amount": "0",
     "currency": "USD",
     "model": "free",
     "provider": "none"
  },
  "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 and header details (`header` logging setting)

V5.0.7 +

Restriction: The `billing` object fields are only supported in Version 5.0.7.2, and later.

```
{
  "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 AwAAABaygGU-"
    },
    {
      "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": "accountservice",
  "catalog_name": "sb",
  "resource_path": "post",
  "plan_name": "default",
  "developer_org_name": "macs-shack",
  "billing": {
    "trial_period_days": "0",
    "amount": "0",
    "currency": "USD",
    "model": "free",
    "provider": "none"
  },
  "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)

V5.0.7+

Restriction: The `billing` object fields are only supported in Version 5.0.7.2, and later.

```
{
  "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 AQAAAPSLVfg-"
    },
    {
      "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",
  "billing": {
    "trial_period_days": "0",
    "amount": "0",
    "currency": "USD",
    "model": "free",
    "provider": "none"
  },
  "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
    ]
  }
}
```

- V5.0.8 + **Customizing the retained event record fields**
  Beginning with version 5.0.8.3, you can customize which event record field information is saved for your IBM® API Connect.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Customizing the retained event record fields

Beginning with version 5.0.8.3, you can customize which event record field information is saved for your IBM® API Connect.

About this task

When you collect large amounts of event logging data, you can quickly run out of storage space. If there is data that is stored that you do not use, you can remove these fields so they are no longer captured and saved with the records. To identify which event record fields to include, complete the following steps:

Procedure

1. Open your Cloud Manager interface.
2. Select the Settings tab to modify the settings.
3. Select Analytics in the navigation to view the analytics settings.
4. In the Analytics Fields section, select Edit to update the specified analytics fields.
   The default values is all analytics fields are captured and recorded.
5. Deselect the fields that you do not want included in the records.
6. Select Update to save any changes, or Cancel to discard your changes.

Related concepts

- The default Overview dashboard
- Analytics and syndication

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Monitoring event fields

A monitoring event is logged approximately every 10 seconds on each management node, and every 5 minutes for each gateway node. The monitoring event record contains information about the status and health of the management node or gateway.

Monitoring event fields for gateway nodes

Table 1 list the fields that might be included in the monitoring event for a gateway node, the type of information in the fields, and their descriptions.

Table 1. Gateway node monitoring event fields

Field name	Type	Description
@timestamp	Date	A time stamp set on the management node that records when the record was written by the Logstash data collection engine that feeds data into Elasticsearch.
cpu	Number	The percentage value of the processing resources that are being used.
datetime	Date	A time stamp set on the gateway that records when the monitoring event was created on a management node or gateway. The time stamp is always shown in coordinated universal time UTC.
gwMeanTranTime	Number	The mean transaction time for the gateway.
gwMem	Number	The memory that was available on the gateway server.
gwTranRate	Number	The transaction rate for the gateway.
memory	Number	The percentage value of the amount of memory that is being used.
nodeId	String	The ID of the node where the event was created.
nodeType	String	The type of node where the event occurred. Valid options are gw (gateway) and ma (management).
plGwRxTp	Number	The gateway platform receive throughput.
plGwTxTp	Number	The Gateway platform transmit throughput.
plLoad	String	The percentage value of the system load.
used_disk	Number	The percentage value of the disk space that is being used.

The following fields are part of the gateway monitoring event record and contain internal information that does not convey any gateway monitoring information:

- @version – Always 1
- client_geoip – Empty array
- gateway_geoip – Empty Array
- headers.*field_name* – Contains internal headers
- host - Always 127.0.0.1
- tags – Empty

Monitoring event fields for management nodes

Table 2 list the fields that might be included in the monitoring event for a management node, the type of information in the fields, and their descriptions.

Table 2. Management node monitoring event fields

Field name	Type	Description
@timestamp	Date	A time stamp set on the management node that records when the record was written by the Logstash data collection engine that feeds data into Elasticsearch.
cpu	Number	The percentage value of the processing resources that are being used.
datetime	Date	A time stamp set on the gateway node that records when the monitoring event was created on a management node or gateway. The time stamp is always shown in coordinated universal time UTC.
memory	Number	The percentage value of the amount of memory that is being used.
nodeId	String	The ID of the node where the event was created.
nodeType	String	The type of node where the event occurred. Valid options are gw (gateway) and ma (management).
used_disk	Number	The percentage value of the disk space that is being used.

The following fields are part of the gateway management record and contain internal information that does not convey any management node monitoring information.

- @version – Always 1
- asRtcGcActivity, asRtMem, asRunningJobs, asWaitingJobs, asWipUsedDisk, asWmcGcActivity, asWmcMem - Always null
- client_geoip, gateway_geoip – Empty array
- gwMeanTranTime, gwMem, gwTranRate, plGwRxTp, plGwTxTp, plLoad – Always null
- headers.*field_name* – Contains internal headers
- host - Always 127.0.0.1
- tags – Empty JSON array

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Audit event fields

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.

Table 1 lists the fields that are included in an audit event record.

Table 1. Audit event fields

Field name	Type	Description
@timestamp	Date	A time stamp that records when the record was written on the management node by the Logstash data collection engine that feeds data into Elasticsearch.
assetType	String	The asset type.
consumerOrgid	String	The identifier of the consumer organization that generated the audit event.”
datetime	Date	Date and time of the audit event.
eventId	String	The identifier of the catalog that generated the audit event.
eventType	String	The type of event.
id	String	Identifier of the event.
message	String	The audit event message.
nlsMessage	Object	Metadata that is related to the audit notification.
notificationType	String	The type of notification.
orgID	String	The identifier of the provider organization that generated the audit event.
source	String	The node where the event was requested.
userId	String	The ID of the user who generated the event.

The following events are listed in the record and contain internal information that does not convey any useful audit event information:

- @version – Always 1
- assetId
- client_geoip, gateway_geoip – Empty array
- headers.*field_name* – Contains internal headers

- host - Always 127.0.0.1
- tags – Empty JSON array

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Log event fields

A log event is recorded each time the gateway encounters a problem. The log event record contains details about the problem that occurred with the gateway.

Table 1 lists the static set of fields that are displayed in an log event record.

Note: Logging of log events is disabled by default because it might have a negative impact on server performance. If you want to use this type of logging, enable the "webapi-mntr" logging target in DataPower.

Table 1. Sample log event fields

Field name	Type	Description
@timestamp	Date	A time stamp that records when the record was written on the management node by the Logstash data collection engine that feeds data into Elasticsearch.
category	String	The log category.
client	String	The IP address of the client server.
datetime	Date	A time stamp that records when the log event occurred on the gateway. The time stamp is always shown in coordinated universal time UTC.
level	String	The severity level of the event.
message	String	Body of the message that was returned to the log.
messageId	String	Identifier of the message.
nodeId	String	The ID of the node where the event occurred.
nodeType	String	The type of node where the event occurred. Valid option is gw (gateway).
source	String	The gateway domain and log category of the log event.
transaction	String	The identification number of the transaction.
transactionType	String	The transaction type.

The following fields are part of the gateway log event record and contain internal information that does not convey any log event information:

- @version – Always 1
- client_geoip, gateway_geoip – Empty array
- headers.*field_name* – Contains internal headers
- tags – Empty JSON array

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing and exporting analytics and API event data

You can obtain analytics and API event data from the API Manager user interface 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.

- V5.0.4 + **Viewing API event data from visualizations**
  In the API Manager UI, you can access the raw data behind a visualization, and then view individual event records that were generated for the APIs.
- V5.0.8 + **Obtaining cluster health information by using REST API calls**
  You can obtain analytics data for your clusters by using Elasticsearch API calls.
- **Exporting analytics and API event data to CSV files**
  You can export the aggregated analytics data from individual visualizations to a comma-separated values (CSV) file. V5.0.4 + You can also export API event data that is associated with an entire dashboard to a CSV file.
- **Obtaining analytics data by using REST API calls**
  You can obtain analytics data for a specific Catalog and API provider organization by using API Manager REST API calls.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.4 +

Viewing API event data from visualizations

In the API Manager UI, you can access the raw data behind a visualization, and then view individual event records that were generated for the APIs.

V5.0.7 + Note: The images displayed in the following procedure depict the UI from earlier API Connect V5.0.x releases, but the illustrations are similar enough to provide a visual guide for completing the steps if needed.

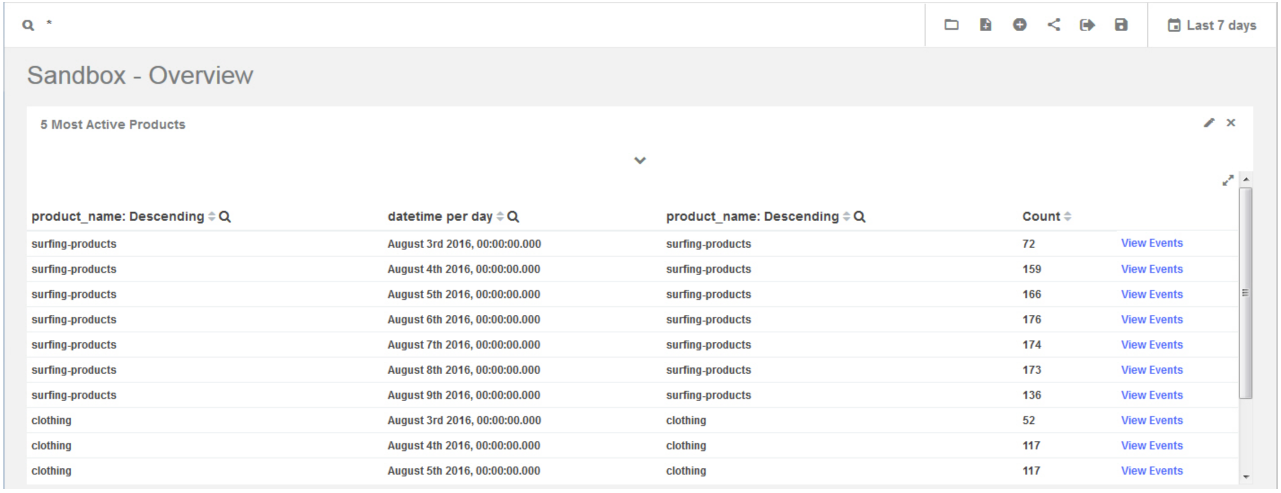
Procedure

To view API event data from a visualization, complete the following steps:

1. V5.0.7 + From a dashboard, view the raw data behind the visualization by hovering over the visualization container and then clicking the caret icon ⌃ that is displayed in the lower left corner.

   V5.0.4 V5.0.5 V5.0.6 From a dashboard, click the caret icon ⌃ on the visualization container to display the raw data behind 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. For example, for a time-based visualization such as the 5 Most Active Products bar chart, which is shown in the default Overview dashboard, individual rows of data are displayed to show the daily activity count for each matching product, over the last seven days.

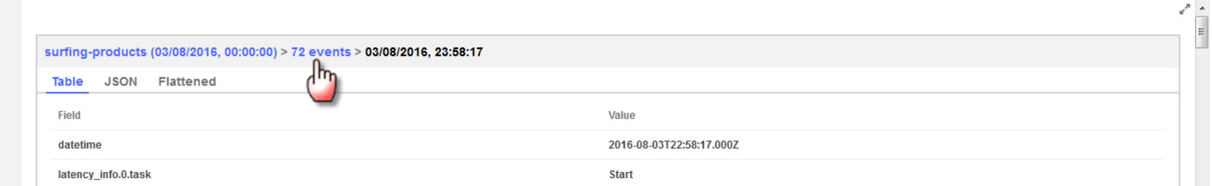


product_name: Descending	datetime per day	product_name: Descending	Count	
surfing-products	August 3rd 2016, 00:00:00.000	surfing-products	72	View Events
surfing-products	August 4th 2016, 00:00:00.000	surfing-products	159	View Events
surfing-products	August 5th 2016, 00:00:00.000	surfing-products	166	View Events
surfing-products	August 6th 2016, 00:00:00.000	surfing-products	176	View Events
surfing-products	August 7th 2016, 00:00:00.000	surfing-products	174	View Events
surfing-products	August 8th 2016, 00:00:00.000	surfing-products	173	View Events
surfing-products	August 9th 2016, 00:00:00.000	surfing-products	136	View Events
clothing	August 3rd 2016, 00:00:00.000	clothing	52	View Events
clothing	August 4th 2016, 00:00:00.000	clothing	117	View Events
clothing	August 5th 2016, 00:00:00.000	clothing	117	View Events

2. To view the associated event records for any row in the grid, click the View Events link for that row.
   Up to 100 of the most recent events are listed, and each event is identified by the timestamp at which it occurred. A breadcrumb trail at the top of the events list shows a short description for the associated data row (for example, ***`product_name`***

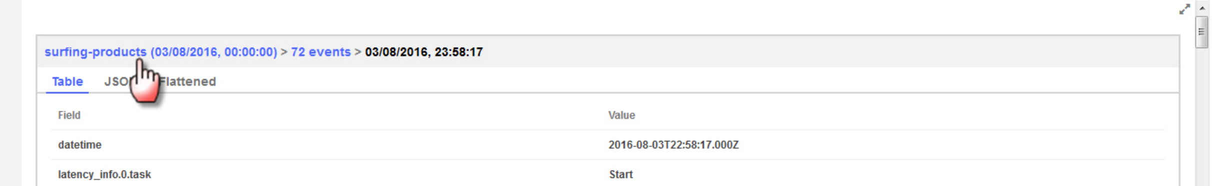
`(timestamp)`) and the total number of event records (for example, `72 events`).

3. To view an event record, click the event row.
   The event data is shown in field/value pairs in the following formats: tabular, JSON (with a nested structure), and flattened JSON (with a single level). For information about the listed event fields, see API event record fields.
4. Click the Table, JSON, or Flattened tab to view the event data in your preferred format.
   Note: The event data content that is displayed 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. For more information, see activity-log policy.
5. View additional event records for the selected data row, or view event records for a different data row in the grid:
   - To return to the list of generated events for the selected data row, click the *n* events link in the breadcrumb trail (where *n* represents a number), as shown in the following figure. Then repeat step 3 and step 4 to view other event details.


surfing-products (03/08/2016, 00:00:00) > 72 events > 03/08/2016, 23:58:17

Field	Value
datetime	2016-08-03T22:58:17.000Z
latency_info.0.task	Start

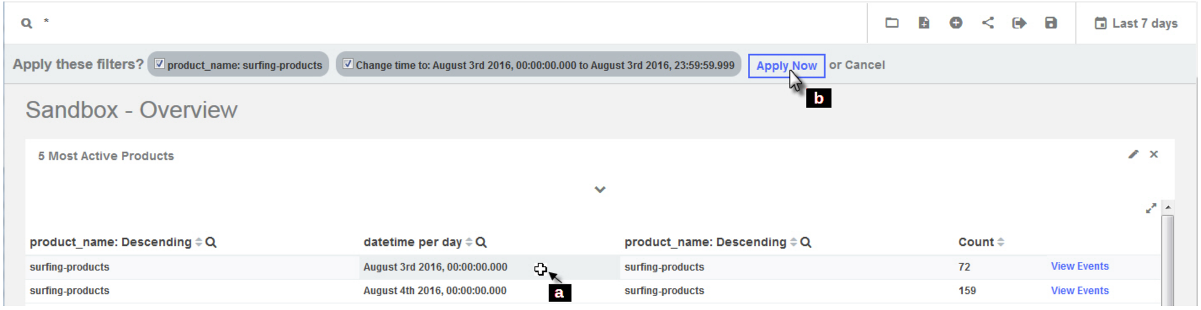
   - To return to the data rows in the initial grid, click the first link in the breadcrumb trail, as shown in the following figure. Then repeat step 2 to step 4 to view the events generated for another row.


surfing-products (03/08/2016, 00:00:00) > 72 events > 03/08/2016, 23:58:17

Field	Value
datetime	2016-08-03T22:58:17.000Z
latency_info.0.task	Start

6. Optional: From a visualization in the dashboard, specify a filter for events that were generated for a specific aggregated data set, and then apply that filter across all visualizations in the dashboard (where relevant):
   a. From the initial "raw data" view of a visualization, locate the data row displaying one or more values that you want to use as filter criteria. Then, click a value in one of the cells to filter by that value only or by a combination of values in the row. Whether a single or combination filter is applied depends on the data type selected and how the aggregations are configured - you can experiment with different visualization types and data sets to see how the filters are applied.
   Note: Not all columns in the grid can be used as a filter. Those columns that can will typically display a plus cursor ✥ when you hover over a cell in the column.
   b. If required, click Apply Now in the filter area beneath the search bar of the dashboard to apply that filter across all visualizations in the dashboard. (If multiple check boxes are displayed for the filter criteria in the filter area, you can optionally clear a check box to change the criteria before you click Apply Now.)
   Note: In some cases, the filter is automatically applied when you click the value in a cell, so this step might not be required.
   The following examples illustrate how filters can be applied from two of the default visualizations.

Example 1:
In the following illustration, the filter is being selected from the 5 Most Active Products bar chart visualization, which is time-based. To filter for events that were generated on August 3rd 2016 and which are associated with the surfing-products Product, click August 3rd 2016 under the datetime per day column and surfing-products row (as depicted by a). Then, from the filter area, click Apply Now (b) to filter for events relating to that Product and time period only.

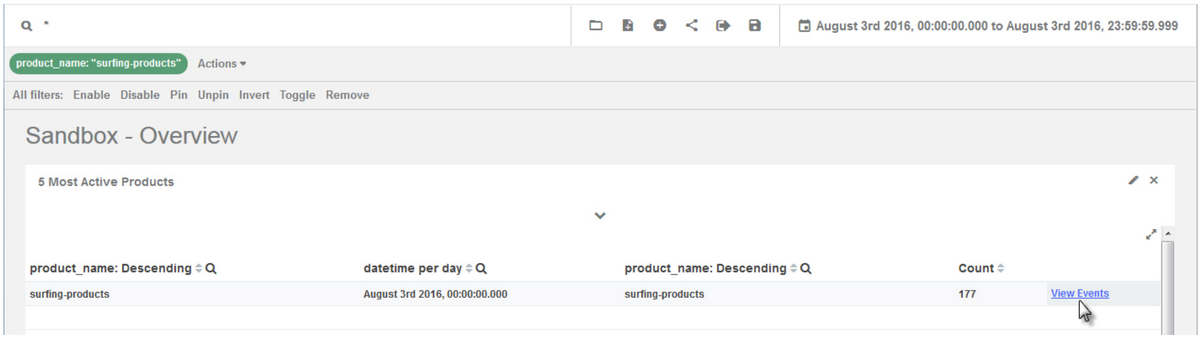

Apply these filters? ☑ product_name: surfing-products ☑ Change time to: August 3rd 2016, 00:00:00.000 to August 3rd 2016, 23:59:59.999 Apply Now or Cancel

Sandbox - Overview

5 Most Active Products

product_name: Descending	datetime per day	product_name: Descending	Count	
surfing-products	August 3rd 2016, 00:00:00.000	surfing-products	72	View Events
surfing-products	August 4th 2016, 00:00:00.000	surfing-products	159	View Events

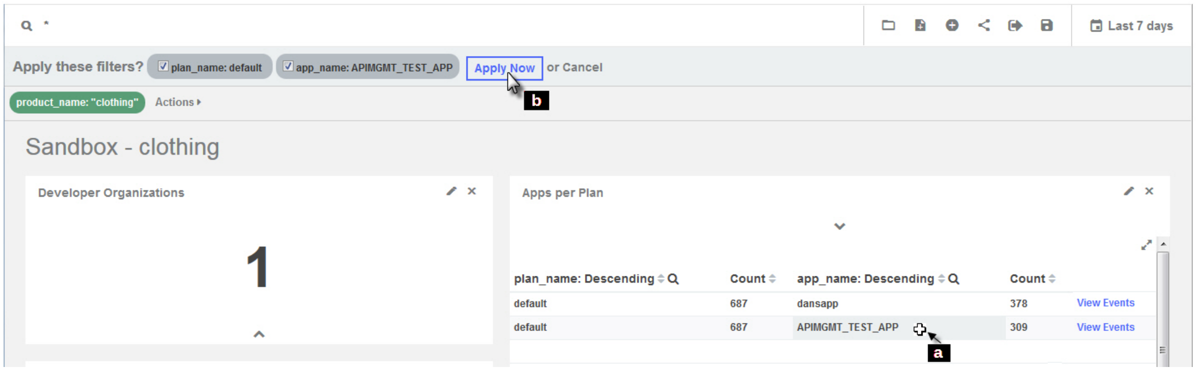
The applied filter is depicted as follows. Notice that the Time Picker now shows August 3rd 2016 as the time filter, and the green filter oval indicates that surfing-products is applied as an inclusion filter.


*
August 3rd 2016, 00:00:00.000 to August 3rd 2016, 23:59:59.999
product_name: "surfing-products"
Actions
All filters: Enable Disable Pin Unpin Invert Toggle Remove
Sandbox - Overview
5 Most Active Products
View Events

product_name: Descending	datetime per day	product_name: Descending	Count	
surfing-products	August 3rd 2016, 00:00:00.000	surfing-products	177	View Events

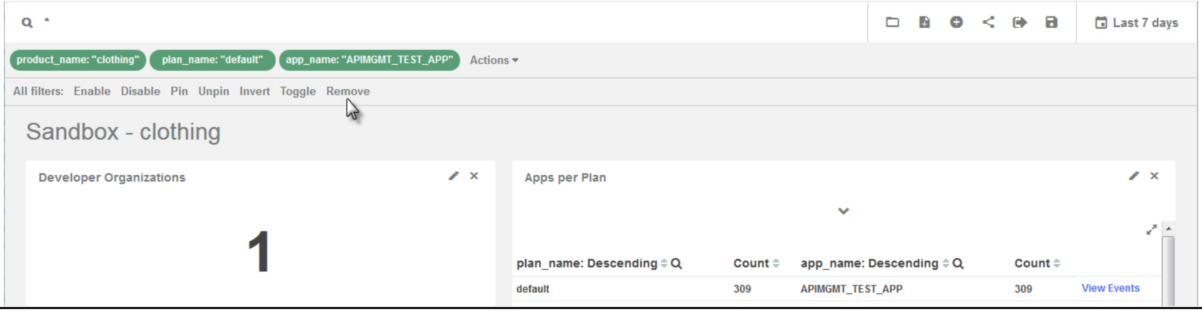
Tip: When setting the filter, if you clicked surfing-products under the first product_name column (instead of August 3rd 2016 in the second column), all Products except surfing-products would be filtered out of the visualizations, but the default Last 7 days time filter would still apply.

Example 2:
In the following illustration, the filter is being selected from the Apps per Plan pie chart visualization. (Notice that a clothing Product filter has been pre-applied to the dashboard.) To filter for events that were generated for the APIMGMT_TEST_APP App in the default Plan, click APIMGMT_TEST_APP under the app_name column and default row (as depicted by a). Then, from the filter area, click Apply Now (b) to filter for events relating to that App and Plan.



plan_name: Descending	Count	app_name: Descending	Count	
default	687	dansapp	378	View Events
default	687	APIMGMT_TEST_APP	309	View Events

The applied filter is depicted as follows. Notice that there is no change to the time filter.



plan_name: Descending	Count	app_name: Descending	Count	
default	309	APIMGMT_TEST_APP	309	View Events

c. From the visualization grid where the filter was selected, click View Events if you want to see the list of related events.
V5.0.7+ You can also click the caret icon ⌃ on another visualization container to display the filtered raw data behind that visualization, and then complete step 2 to step 4 to view the event records.

V5.0.4 V5.0.5 V5.0.6 You can also click the caret icon ⌃ on another visualization container to display the filtered raw data behind that visualization, and then complete step 2 to step 4 to view the event records.

d. To remove the filter, complete one or more of these steps based on the filter criteria that was specified:
   i. If applicable, remove the inclusion filter for one or more values by clicking Remove in the filter area. If you do not see the Remove option, click the Actions twistie to expand the menu.
   ii. If applicable, change to the previous or a different time range, by clicking the Time Picker icon (V5.0.7+ 🕘 V5.0.4 V5.0.5 V5.0.6 📅). Then, use the Quick, Relative, or Absolute option to set the required time filter. For example, if your default time period was Last 7 days, click Quick and then click Last 7 days. For more information, see Specifying a time period and auto-refresh rate for the data in your visualizations.
   iii. If you see Apply these filters? under the search bar, click Cancel.

7. V5.0.7+ To close the "raw data" view of the visualization, hover over its container and then click the toggle icon ⌄ in the lower left corner.

V5.0.4 V5.0.5 V5.0.6 To close the "raw data" view of the visualization, click the toggle icon ⌄ on the visualization container.

Related tasks

- Exporting analytics and API event data to CSV files
- Obtaining analytics data by using REST API calls

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Obtaining cluster health information by using REST API calls

You can obtain analytics data for your clusters by using Elasticsearch API calls.

Before you begin

You must have administrative permissions on the cmc to access these REST APIs.

About this task

There are times when you need to make sure that your clusters are working correctly. Elasticsearch provides a REST API that enables you to view a red, yellow, or green status for your cluster.

Procedure

- To obtain health data for your cluster, issue the following call:

  ```
  curl -k -X GET -u 'cmc/username' 'https://api-manager-server.company.com/v1/es/elasticsearch_api'
  ```

  where
  - *username* is your username that has cmc.admin permissions.
  - *elasticsearch_api* is the name of the Elasticsearch REST API that you are calling. The API must be a GET method of the following types of REST API calls:
    - _cat
    - _cluster
      The cluster API calls include the following nodes API calls, which provide information about the nodes in a cluster:
      - _nodes_info
      - _nodes_stats

  Important: The API call must be authenticated (you must pass Cloud Manager admin credentials to the Management server). The credentials that are required must be prefixed with `cmc/`. An authentication error occurs if you do not apply the `cmc/` prefix.
- If you notice that some Elasticsearch shards cannot be assigned due to a retry limit being exceeded, you can clear the retry limit by issuing the following call:

  ```
  curl -k -X POST -u 'cmc/username' 'https://api-managerserver.company.com/v1/es/_cluster/reroute?
  retry_failed=true&pretty=true' -H 'Content-type: application/json'
  ```

  where *username* is your username that has cmc.admin permissions.

  Clearing the retry limits enables the shards to be assigned.

Sample calls and responses

The following examples show sample formats for calls that are issued from a browser address bar to obtain analytics data for a cluster. In the URL, the host address of the API Manager server being queried is specified. A secure login form will prompt for a user name and password for authenticating to the API Manager server. (The API Manager server details are those used for accessing the API Manager UI.)

```
https://api-manager-server.company.com/v1/es/_cat/indices
```

```
https://api-manager-server.company.com/v1/es/_cluster/health
```

```
https://api-manager-server.company.com/v1/es/_nodes
```

The following examples show sample formats for calls that are issued as **curl** commands to obtain analytics data for a cluster. In the commands, the Internet protocol string specifies the host address of the API Manager server being queried, and any query parameters are appended with a leading `?` and separated with an ampersand (`&`). The -u option specifies only the user name, rather than the user name and password combination (*user_name*:*password*), to prevent the password from being shown in plain text.

Linux

```
curl -k -X GET -u 'cmc/username' 'https://api-manager-server.company.com/v1/es/_cat/indices'
```

Windows

```
curl -k -X GET -u "cmc/username" "https://api-manager-server.company.com/v1/es/_cat/indices"
```

Related tasks

- Viewing API event data from visualizations
- Exporting analytics and API event data to CSV files

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Exporting analytics and API event data to CSV files

You can export the aggregated analytics data from individual visualizations to a comma-separated values (CSV) file. V5.0.4 + You can also export API event data that is associated with an entire dashboard to a CSV file.

- Exporting analytics data from a visualization
- V5.0.4 + Exporting API event data from a dashboard

Exporting 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

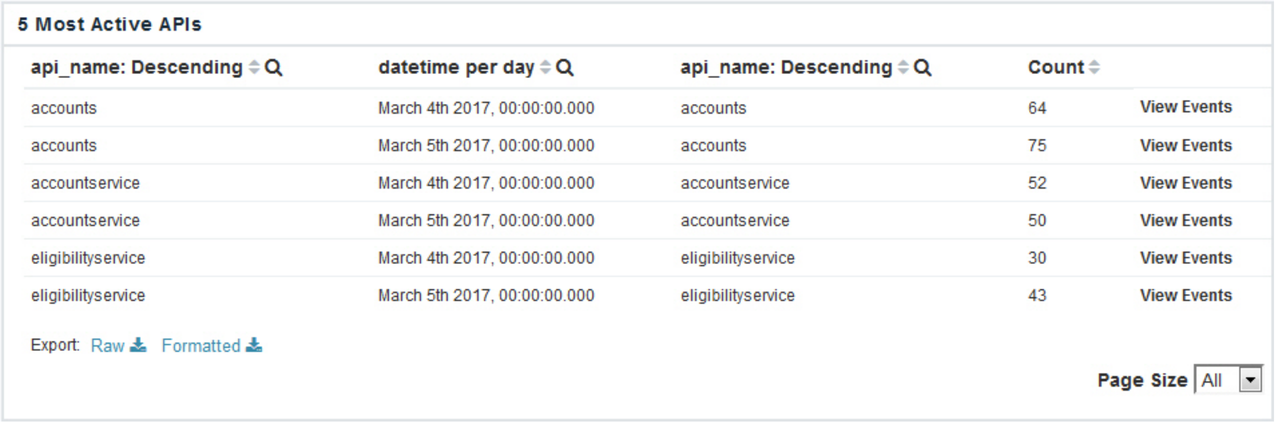
To export analytics data from a visualization, complete the following steps:

1. V5.0.7 + From a dashboard, view the raw data behind the visualization by hovering over the visualization container and then clicking the caret icon ^ that is displayed in the lower left corner.

   V5.0.6 and earlier From a dashboard, click the caret icon ^ on the visualization container to display the raw data behind 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.

   V5.0.7 +



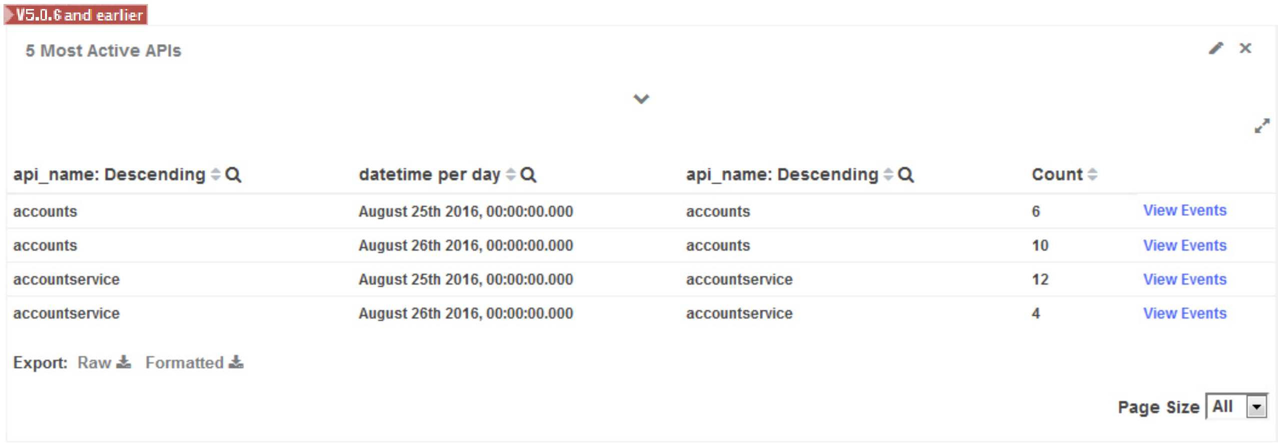
5 Most Active APIs

api_name: Descending	datetime per day	api_name: Descending	Count	
accounts	March 4th 2017, 00:00:00.000	accounts	64	View Events
accounts	March 5th 2017, 00:00:00.000	accounts	75	View Events
accountservice	March 4th 2017, 00:00:00.000	accountservice	52	View Events
accountservice	March 5th 2017, 00:00:00.000	accountservice	50	View Events
eligibilityservice	March 4th 2017, 00:00:00.000	eligibilityservice	30	View Events
eligibilityservice	March 5th 2017, 00:00:00.000	eligibilityservice	43	View Events

Export: Raw Formatted

Page Size All

V5.0.6 and earlier



5 Most Active APIs

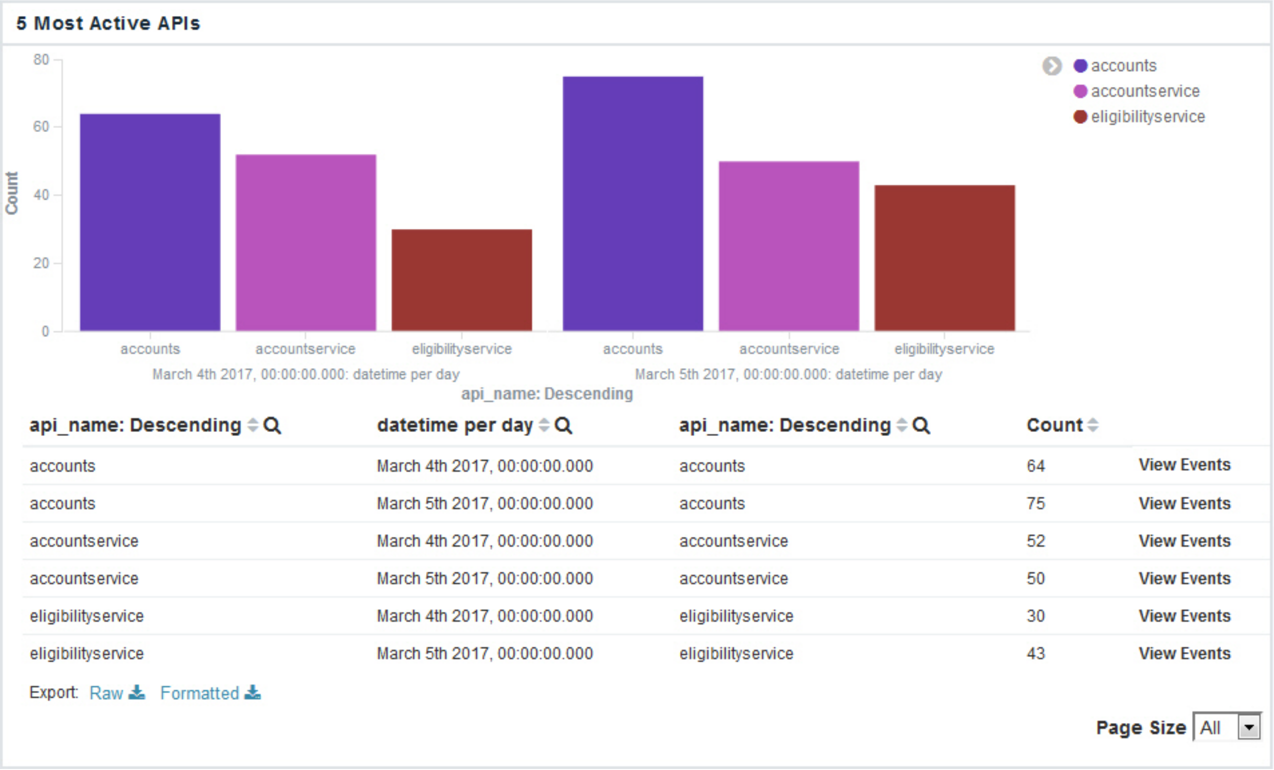
api_name: Descending	datetime per day	api_name: Descending	Count	
accounts	August 25th 2016, 00:00:00.000	accounts	6	View Events
accounts	August 26th 2016, 00:00:00.000	accounts	10	View Events
accountservice	August 25th 2016, 00:00:00.000	accountservice	12	View Events
accountservice	August 26th 2016, 00:00:00.000	accountservice	4	View Events

Export: Raw Formatted

Page Size All

Tip: If you scroll vertically or resize the container to eliminate the vertical scroll bar, you can see the Raw and Formatted links that can be used to export data, and also see the Page Size drop-down list, which determines how many rows are visible at a time. Notice also that if you continue to increase the height of the container, you will be able to see both the graphical representation and raw data at the same time, as shown in the following image. (V5.0.7 + You can hide the graph again by decreasing the container height. V5.0.6 and earlier You can hide the graph again by decreasing the container height or clicking the arrow icon ⤢ .)

V5.0.7 +



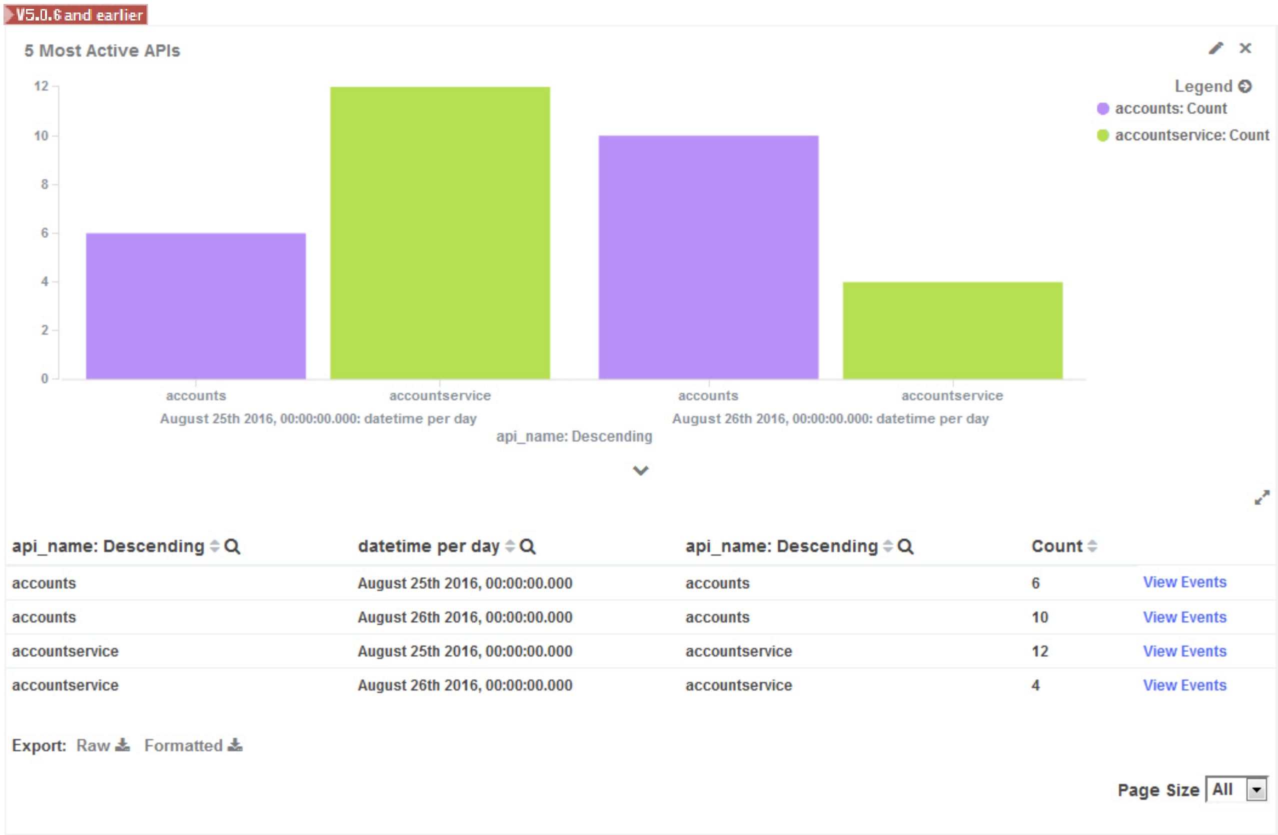
5 Most Active APIs

api_name: Descending	datetime per day	api_name: Descending	Count	
accounts	March 4th 2017, 00:00:00.000	accounts	64	View Events
accounts	March 5th 2017, 00:00:00.000	accounts	75	View Events
accountservice	March 4th 2017, 00:00:00.000	accountservice	52	View Events
accountservice	March 5th 2017, 00:00:00.000	accountservice	50	View Events
eligibilityservice	March 4th 2017, 00:00:00.000	eligibilityservice	30	View Events
eligibilityservice	March 5th 2017, 00:00:00.000	eligibilityservice	43	View Events

Export: Raw Formatted

Page Size All

V5.0.6 and earlier



api_name: Descending	datetime per day	api_name: Descending	Count	
accounts	August 25th 2016, 00:00:00.000	accounts	6	View Events
accounts	August 26th 2016, 00:00:00.000	accounts	10	View Events
accountservice	August 25th 2016, 00:00:00.000	accountservice	12	View Events
accountservice	August 26th 2016, 00:00:00.000	accountservice	4	View Events

2. 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.
3. V5.0.7+ Choose to save the file, which is named *visualization_name*.csv by default.
   V5.0.6 and earlier Choose to save the file, which is named table.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 to help you differentiate between them.

4. V5.0.7+ To close the "raw data" view of the visualization, hover over its container and then click the toggle icon ⌄ in the lower left corner.
   V5.0.6 and earlier To close the "raw data" view of the visualization, click the toggle icon ⌄ on the visualization container.

V5.0.4+ Exporting API event data from a dashboard

You can export the API event data that is collectively generated for the aggregated data sets in all visualizations, and across the defined time frame, in a dashboard. When you export the event data, a CSV file is created. The CSV file contains one API event record for each API invocation in the Gateway server.

Procedure

To export API event data from all visualizations in a dashboard, complete the following steps:

1. Open the dashboard from which you want to export data, as described in Opening saved dashboards.
2. If required, apply filters to change the sampling of data displayed in the visualizations. For example, specify a different time period, or apply inclusion or exclusion filters.
   For more information, see Applying filters to change the sampling of data displayed in visualizations.
   Note: Only API event records for the currently filtered data set are exported to the CSV file.
3. V5.0.7+ From the dashboard, click the Export icon ⤓.
   V5.0.4 V5.0.5 V5.0.6 From the dashboard, click the Export icon ⇨.
   V5.0.7+ The Export Analytics Events panel opens to display a message, which informs you that a CSV file of the exported event data will be generated in the background. V5.0.4 V5.0.5 V5.0.6 You see a message on the screen, which informs you that a downloadable CSV file of the collected data will be generated in the background.

4. V5.0.7+ Click the Export button to start the export. You can view the status bar to monitor the progress as the file downloads to the server. You can select Download
   V5.0.4 V5.0.5 V5.0.6 Click the Export button to close the message and start the export.
   Important:

- V5.0.7+ While the file is being generated, you can continue to work within the analytics dashboards under the Analytics tab. If you switch from Analytics to another tab (such as Products or Settings), or to another component in the API Manager user interface, it cancels the download step of your Export operation. If you leave the tab, the initial export process to the server continues in the background. If it is within the time that the Exported files are retained (usually 1 hour by default) and no other Exports are started, you can return to the Analytics tab and select Download to download the file from the last Export.
- V5.0.4 V5.0.5 V5.0.6 While the file is being generated, you can continue to work within the analytics dashboards under the Analytics tab. Do not, however, switch from Analytics to another tab (such as Products or Settings), or to another component in the API Manager user interface because this will cancel the export operation.
- If your browser is configured to block pop-ups, the download of the CSV file 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 attempt the export again.

5. If prompted, choose to save the file, which is named analytics_export.csv by default.
   The file might also be automatically downloaded depending on your browser.
   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.
   The CSV file contains a row for each API event record. You can import this CSV file into other software tools to further analyze your API Connect usage. For information about the fields that are included in the CSV file, see API event records.

   Note: 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.
6. V5.0.7+ Click the Export icon ⬇ to close the Export Analytics Events message panel.

Related tasks

- Viewing API event data from visualizations
- Obtaining analytics data by using REST API calls

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining analytics data by using REST API calls

You can obtain analytics data for a specific Catalog and API provider organization by using API Manager REST API calls.

About this task

The REST API calls return JSON objects that contain analytics data. The JSON objects contain the same information as the API event records that you see when you use the API Manager user interface to view the raw data for visualizations.

Procedure

To obtain analytics data for a Catalog and API provider organization, issue the following call:

```
GET /v1/orgs/{orgId}/environments/{envId}/events
```

where

- *{orgId}* is either the URL path segment or the ID of the API provider organization.
- *{envId}* is either the URL path segment or the ID of the Catalog.

The following table details the query parameters for this call:

Table 1. Query parameters for the analytics REST API call

Parameter	Description	Default
after	The start of the time period for which you want to obtain analytics data. Specify the date/time in UTC format, without a timezone offset; for example, `yyyy-MM-dd` or `yyyy-MM-ddTHH:mm:ss.sss`.	Current time minus 24 hours
before	The end of the time period for which you want to obtain analytics data. Specify the date/time in UTC format, without a timezone offset; for example, `yyyy-MM-dd` or `yyyy-MM-ddTHH:mm:ss.sss`.	Current time

Parameter	Description	Default
fields	A list of the event fields that you want included in each event. You can use a comma as a separator when specifying the fields. Use this parameter if you require only a subset of the event fields rather than the full set.	Returns all fields by default
limit	The number of analytics events that you want to be returned in a single call. If the total number of events is greater than the specified limit, then to obtain the remaining events you issue further calls and supply the next parameter. Note: The maximum allowed value for the limit parameter is 10000. If this value is exceeded, the API call fails and returns error code 500.	100
next	An ID that is returned on the first call to the API. Supply this ID on subsequent calls to obtain the next set of events. The number of events that is returned in each set of events is determined by the value of the limit parameter.	(Not applicable)
V5.0.8 + timeZoneOffset	V5.0.8 + **From API Connect version 5.0.8.5 onward:** The amount of time, in minutes, by which you want the returned `datetime` and `timestamp` fields to be adjusted, compared to the original server based values. To add time to the original values, supply a positive value. To subtract time from the original values, supply a negative value. The value must be an integer. Note: `datetime` is set by the server, whereas `timestamp` is set by the analytics engine and is very slightly later.	V5.0.8 + 0

Requests are authenticated using HTTP Basic Authentication. For more information, see Authenticating requests.

Sample calls and responses

The following examples show sample formats for calls that are issued from a browser address bar to obtain analytics data for a Catalog and API provider organization. In the URL, the host address of the API Manager server being queried is specified, and any query parameters are appended with a leading `?` and separated with an ampersand (`&`). A secure login form will prompt for a user name and password for authenticating to the API Manager server. (The API Manager server details are those used for accessing the API Manager UI.)

```
https://api-manager-server.company.com/v1/orgs/macs-shack/environments/sb/events
```

```
https://api-manager-
server.company.com/v1/orgs/57cd809ce4b09c68a2183832/environments/57cd80a6e4b09c68a218387b/events
```

```
https://api-manager-server.company.com/v1/orgs/smallorg/environments/dev/events?limit=2
```

```
https://api-manager-server.company.com/v1/orgs/macs-shack/environments/sb/events?after=2016-08-
01T00:00:00.000&before=2016-09-30T23:59:59.999
```

The following examples show sample formats for calls that are issued as **curl** commands to obtain analytics data for a Catalog and API provider organization. In the commands, the Internet protocol string specifies the host address of the API Manager server being queried, and any query parameters are appended with a leading `?` and separated with an ampersand (`&`). The -u option specifies only the user name, rather than the user name and password combination (***user_name*:*password***), to prevent the password from being shown in plain text.

Linux

```
curl -k -X GET -u 'username@company.com' 'https://api-manager-server.company.com/v1/orgs/macs-shack/
environments/sb/events?after=2016-08-01T00:00:00.000&before=2016-09-30T23:59:59.999'
```

Windows

```
curl -k -X GET -u "username@company.com" "https://api-manager-server.company.com/v1/orgs/macs-shack/
environments/sb/events?after=2016-08-01T00:00:00.000&before=2016-09-30T23:59:59.999&limit=2"
```

V5.0.7 +

Restriction: The `billing` object fields are only supported in Version 5.0.7.2, and later.

The following example shows a call that is issued from a browser and then shows all the event fields in the corresponding response. The call is issued to return two analytics events (`limit=2`), and includes a next parameter for requesting further events.

```
https://api-manager-server.company.com/v1/orgs/macs-shack/environments/sb/events?next&limit=2
```

```
{
  "totalCalls": 5,
  "next": "5792250fe4b0838bc6abfc0a",
  "nextHref": "https://api-manager-server.company.com/v1/orgs/macs-shack/environments/sb/events?
next=5792250fe4b0838bc6abfc0a",
  "calls": [
    {
      "datetime": "2016-07-22T13:51:08.658Z",
      "timestamp": "2016-07-22T13:51:08.748Z",
      "apiName": "fins",
      "apiVersion": "1.0.0",
      "appName": "Surfboards",
      "catalogName": "sb",
      "planName": "default",
      "planVersion": "1.0.0",
      "productName": "surfing-products",
      "productVersion": "1.0.0",
      "devOrgName": "bluebottle-boards",
      "resourceName": "/stocklist",
      "timeToServeRequest": 243,
      "bytesSent": 115,
      "requestProtocol": "https",
      "requestMethod": "GET",
      "uriPath": "/macs-shack/sb/surf/stocklist",
      "queryString": "",
      "statusCode": "200 OK",
      "requestHeaders": "",
      "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) 
Chrome/51.0.2704.106 Safari/537.36",
      "requestBody": "",
      "responseHeaders": "",
      "responseBody": "",
      "latency": "Initialization=0ms : Start=8ms : PlanRateLimits=4ms : Plan Limit=2ms : 
Finservice=221ms : PlanRateLimits=8ms",
      "rateLimit": {
        "per-minute": {
          "limit": "100",
          "count": "5",
          "period": "60",
          "reject": "false",
          "shared": "true"
        }
      },
      "logPolicy": "activity",
      "orgName": "macs-shack",
      "resourcePath": "get",
      "productTitle": "Surfing Products",
      "clientGeoIp": {
        "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
        ]
      },
      "billing": {
        "trial_period_days": "0",
        "amount": "0",
        "currency": "USD",
        "model": "free",
        "provider": "none"
      }
      "gatewayGeoIp": {
        "ip": "9.20.98.109",
        "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
        ]
      }
    },
    {
      "datetime": "2016-07-22T13:51:07.375Z",
      "timestamp": "2016-07-22T13:51:08.465Z",
      "apiName": "fins",
      "apiVersion": "1.0.0",
      "appName": "Surfboards",
      "catalogName": "sb",
      "planName": "default",
      "planVersion": "1.0.0",
      "productName": "surfing-products",
      "productVersion": "1.0.0",
      "devOrgName": "bluebottle-boards",
      "resourceName": "/stocklist",
      "timeToServeRequest": 248,
      "bytesSent": 115,
      "requestProtocol": "https",
      "requestMethod": "GET",
      "uriPath": "/macs-shack/sb/surf/stocklist",
      "queryString": "",
      "statusCode": "200 OK",
      "requestHeaders": "",
      "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) 
Chrome/51.0.2704.106 Safari/537.36",
      "requestBody": "",
      "responseHeaders": "",
      "responseBody": "",
      "latency": "Initialization=0ms : Start=10ms : PlanRateLimits=4ms : Plan Limit=2ms : 
Finservice=225ms : PlanRateLimits=7ms",
      "rateLimit": {
        "per-minute": {
          "limit": "100",
          "count": "4",
          "period": "60",
          "reject": "false",
          "shared": "true"
        }
      },
      "logPolicy": "activity",
      "orgName": "macs-shack",
      "resourcePath": "get",
      "productTitle": "Surfing Products",
      "clientGeoIp": {
        "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
        ]
      },
      "gatewayGeoIp": {
        "ip": "9.20.98.109",
        "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
        ]
```

```
            }
        }
    ]
}
```

Related tasks

- Viewing API event data from visualizations
- Exporting analytics and API event data to CSV files

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing notifications of activities and alerts

You can use the Notifications view to track all user activity in the API Manager user interface; for example, the creation of a Product, or the deletion of an API.

About this task

V5.0.7+ You use the Notifications icon 🔔 to view notifications. V5.0.6 and earlier You use the Notifications icon 🔔 to view notifications. This icon is displayed in the primary banner of every page in the API Manager user interface.

V5.0.7+ Restriction: If the Notifications view displays no data or shows 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 cloud administrator for confirmation. (For more information, see Configuring destination targets for API Connect analytics data.)

Procedure

To view details of notifications, complete the following steps:

1. V5.0.7+ Click the Notifications icon 🔔.

   V5.0.6 and earlier Click the Notifications icon 🔔.
   A log of activities is displayed in the Notifications view. On initial access, all activities are listed. On subsequent access, the level of detail shown might change based on the last set of activities accessed.
2. View the activities by category as follows:
   - To view all user activity, all API activity, and all alerts, click the Show all activity icon ☰. The activities are listed in reverse chronological order, with the most recent first.
   - To view all of the user activity and to find out whether the activity was successful, click the Show user activity icon 👤.
   - To view notifications of issues, click the Show alerts icon ⚠.
3. To check for new notifications while the Notifications page is open, click the Refresh icon ↻.
   The page refreshes to show the most recent activity.
4. When your review is complete, exit the Notifications view by clicking the Back button on your browser.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Administering user access

If you have permission to administer users, 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 followings tasks:

- **Viewing your user information**
  You can view the users that can access your organization.
- **Adding provider organization users and assigning roles**
  If you have the permissions that are required to edit users, you can add users to a provider organization, remove users, assign roles and perform other user administration tasks.
- **Creating custom roles**
  If you have permission to edit roles, 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 an organization**
  If you have permission to edit users, you can remove a user from an organization.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing your user information

You can view the users that can access your organization.

Procedure

To view your organization user information, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the Navigation pane, click Admin, then click the Members tab.
   A list of users is displayed, showing their email addresses, user role authorizations, status, and when they last logged in.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding provider organization users and assigning roles

If you have the permissions that are required to edit users, you can add users to a provider organization, remove users, assign roles and perform other user administration tasks.

Procedure

To add users and assign user roles for your provider organization, complete the following steps:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.

2. V5.0.4 and earlier In the Navigation pane, click Admin, then click Users.
3. V5.0.5 + In the Navigation pane, click Admin, then click Members.
4. Click Add. The Add User window is displayed.

   For information on how to create and configure registries for API Manager users, see Specifying the cloud settings.
5. Specify a user and assign the user a role:
   a. Optional: To authenticate API Manager users with a local registry, complete one of the following steps:
      -
        - To add an existing user , click Existing User, then enter a search string and select the user from the Search Results scroll window.
        - To add a new user, click New User, then enter the user's email into the text field.

   b. Optional: To authenticate API Manager users with LDAP, enter a search string and select the user from the Search Results scroll window.

      Note: When you use a local registry or LDAP, if the search returns more than 100 results, refine your filter string.

   c. Optional: If the provider is an authentication URL, enter the email address of the intended invitee.

   d. Assign the user a role by completing the following steps:

      i. V5.0.3 and earlier Click the ⊕ icon. The Assign Role window displays a list of available roles. Roles that are shown are those available to the provider organization and any custom roles created. As new roles are created, they are displayed in the Assign Role window.

         Note: If you assign a user more than one role, the user obtains permissions for both roles.

      ii. In the Assign Role section, select the user role. The assigned role is displayed if you navigate to the Roles column of the Users window. The following roles are available and displayed:

         Administrator
            The Administrator can create and edit APIs, operations, Products, developer organizations, users, identity providers, and Catalogs.

         Product Manager
            The Product Manager is responsible for commissioning APIs and tracking their business adoption. The Product Manager can perform the same tasks as the developer as well as being able to create and edit Developer organizations.

         API Developer
            API Developers create and configure APIs, Products, and policies for provider organizations of 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.

         V5.0.4 and earlier Publisher
            V5.0.4 and earlier The Publisher manages the lifecycles of APIs and publishes Products to selected communities of application developer organizations.

         V5.0.5 + API Administrator
            V5.0.5 + The API Administrator manages the lifecycles of APIs and publishes Products to selected communities of application developer organizations.

         Custom
            You can create custom roles and specify the permissions and functionality. The role appears in the list only after you have created it. For more information on creating custom roles, see Creating custom roles.

         Note: When a provider organization is created, the specified owner is automatically given the Owner role, and is granted all permissions. The Owner role for a provider organization cannot be assigned to another user.

         For full details of the permissions that are assigned to the various API Manager roles, see API Connect user roles.

6. Click Add.

   The user's name is added to the Name column and the email invitation is sent. `Invitation Pending` is displayed in the Status column of the Users window.

   Important: When a user is invited to join a provider organization that is secured by using LDAP, the user name that is created is case-sensitive regardless of whether the LDAP registry is configured to be case sensitive. Therefore, if a user is sent an invitation that includes a user name of USERA@mail.com, this user must sign in to the API Manager user interface with USERA@mail.com (and not usera@mail.com). If the user enters the Username incorrectly due to using the wrong case, the user cannot sign in and an error is displayed.

Results

The API Connect user account is created and is activated when the invitee opens the email and clicks the activation link.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating custom roles

If you have permission to edit roles, 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

The following tables provide a list of available permissions and what they represent. For a listing of the default API Manager roles and permissions, see Adding provider organization users and assigning roles.

V5.0.5+ Permissions can be applied on an organizational or Catalog level.

V5.0.4 and earlier

Table 1. User role descriptions

Permissions	Actions	Permits the user to
Roles	View	View the Roles tab
	Edit	Create, edit, and delete roles
Users	View	View the users that are in an organization
	Edit	Add and update users to, and delete users from, an organization
TLS Profiles	View	View TLS profiles
	Edit	Create, edit, and delete TLS profiles
User Registries	View	View user registries
	Edit	Create, edit, and delete user registries
Draft APIs	View	View draft APIs
	Edit	Create, update, and delete draft APIs
Draft Products	View	View draft Products
	Edit	Create, update, and delete draft Products
Subscriptions	View	View Plan subscriptions
	Approve	Approve Plan subscriptions
Catalog Administration	View	View Catalogs
	Edit	Create, update, and delete Catalogs
Developers	View	View developers and developer organizations
	Manage	Add, update, and delete developers and developer organizations
Analytics	View	View Catalog analytics

V5.0.5+

Table 2. Organization permissions

Permissions	Action	Permits the user to
Draft APIs	View	View draft APIs
	Edit	Edit draft APIs
Organization Settings	View	View organization's configuration settings Note: A user with Organization Settings > View permission can view Roles, TLS Profiles, and User Registries.
	Manage	Manage organization's configuration settings Note: A user with Organization Settings > Manage permission can manage Roles, TLS Profiles, and User Registries.
Catalogs	Create	Create Catalogs in the organization

Permissions	Action	Permits the user to
	View	View all Catalogs in the organization
	Manage	Manage all Catalogs in the organization
Draft Products	View	View draft Products
	Edit	Edit draft Products
Organization Members	View	View organization's members
	Manage	Manage organization's members

V5.0.5+

Table 3. Catalog Permissions

Permissions	Action	Permits the user to
Catalog Members	View	View Catalog members
	Manage	Manage Catalog members
Catalogs Settings	View	View the Catalog configuration settings
	Manage	Manage the Catalog configuration settings
Subscriptions	View	View subscriptions
	Manage	Manage subscriptions
API Products	Stage	Stage Products in a Catalog
	View	View Products in a Catalog
	Manage	Manage Products in a Catalog
Subscription Approvals	View	View subscription approvals
	Manage	Manage subscription approvals
V5.0.7+ Subscription and Application Approvals	V5.0.7+ View	V5.0.7+ View subscription and application upgrade approvals
V5.0.7+	V5.0.7+ Manage	V5.0.7+ Manage subscription and application upgrade approvals
Analytics	View	View analytics
	Manage	Manage analytics
Applications	View	View applications
	Manage	Manage applications
Developer Organizations and Developers	View	View developer organizations and developers
	Manage	Manage developer organizations and developers
Product Lifecycle Approvals	View	View Product lifecycle changes
	Stage	Stage Products
	Publish	Publish Products
	Deprecate	Deprecate Products
	Retire	Retire Products
	Replace	Replace Products
	Supersede	Supersede Products
Spaces	Create	Create Spaces
	View	View Spaces
	Manage	Manage Spaces

V5.0.5+

Table 4. Space Permissions

Permissions	Action	Permits the user to
Space Members	View	View Space members
	Manage	Manage Space members
Space Settings	View	View the Space configuration settings
	Manage	Manage the Space configuration settings
Subscriptions	View	View subscriptions
	Manage	Manage subscriptions
API Products	Stage	Stage Products in a Space
	View	View Products in a Space
	Manage	Manage Products in a Space
Subscription Approvals	View	View subscription approvals
	Manage	Manage subscription approvals
V5.0.7+ Subscription and Application Approvals	V5.0.7+ View	V5.0.7+ View subscription and application upgrade approvals
V5.0.7+	V5.0.7+ Manage	V5.0.7+ Manage subscription and application upgrade approvals
Analytics	View	View analytics

Permissions	Action	Permits the user to
	Manage	Manage analytics
Applications	View	View applications
	Manage	Manage applications
Developer Organizations and Developers	View	View developer organizations and developers
	Manage	Manage developer organizations and developers
Product Lifecycle Approvals	View	View Product lifecycle changes
	Stage	Stage Products
	Publish	Publish Products
	Deprecate	Deprecate Products
	Retire	Retire Products
	Replace	Replace Products
	Supersede	Supersede Products

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

You can create custom roles by following the procedure:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
2. In the Navigation pane, click Admin > Roles.
   The Roles page opens.
3. Click Add.
   A new role appears in the list of roles.
4. V5.0.4 and earlier Click on the new role, then enter the name and description of the custom role in the corresponding Role Name and Description text fields.
5. V5.0.5 + Click on the new role, then enter the display name, name, and description of the custom role in the corresponding Display Name, Name and Description text fields.
   Note:
   - The Display Name field can contain a maximum of 25 characters.
   - The value that you enter for the Name can be used to reference the role through the CLI.
6. Use the check boxes to assign permissions to the new role.
7. When you are finished, click Save.
8. To delete a role, click the Delete icon alongside the required role.
   Note: You can delete a role only at the provider organization level. You cannot delete a role at the Catalog level. V5.0.5 + 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. V5.0.5 + 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 IBM 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 concepts

- Creating and configuring Catalogs

Related tasks

- Adding provider organization users and assigning roles

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing a user from an organization

If you have permission to edit users, you can remove a user from an organization.

About this task

After you have removed the user, their resources are deleted and the user cannot access any of the organization's artifacts. The user account remains in API Manager.
Note: You cannot remove an Owner from an organization.

Procedure

You can remove a user from an organization by following the procedure:

1. If you have not previously pinned the UI navigation pane then click the Navigate to icon ☰.

   The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon .
2. V5.0.4 and earlier In the Navigation pane, click Admin, then click Users.
3. V5.0.5 + In the Navigation pane, click Admin, then click Members.
4. Click the Delete icon that is adjacent to the user you want to delete.
   Note: You cannot delete the owner of an organization.

Results

The user is removed from the organization, and the user account remains in API Manager.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your API Manager password

You can change your API Manager password.

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 My Account.
3. In the Change password section, enter your current password and your new password, and confirm the new password.
   The password must consist of at least six characters and must contain at least three of the following types of characters:
   - Lowercase letters
   - Uppercase letters
   - Numbers
   - Special characters; the following characters are allowed:

     ```
     !
     \"
     #
     $
     %
     ```

```
&
\
(
)
*
+
,(comma)
- (dash/hyphen)
.(period)
/
:(colon)
;(semi-colon)
<
=
>
?
@
[
\\
]
^
_ (underscore)
` (back quote/grave accent)
{
| (pipe)
}
~ (tilde)
```

The password cannot contain more than two consecutive repeating characters.

4. Click Save.

Results

Your password is changed.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reference

Reference information for the API Manager component in API Connect.

- **Named queries for discovering a service in WSRR**
  If you have a customized data model in WebSphere® Service Registry and Repository (WSRR) then, depending on the degree of customization, your WSRR administrator might have to define a series of named queries in WSRR so that the API Connect system software can discover services in your registry. However, if only the lifecycles or classifications have been modified then this step is not required.
- **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.
- V5.0.8 + **Troubleshooting your billing configuration**
  Some issues can occur when you are setting up your billing plan for subscriptions in IBM® API Connect.

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Named queries for discovering a service in WSRR

If you have a customized data model in WebSphere® Service Registry and Repository (WSRR) then, depending on the degree of customization, your WSRR administrator might have to define a series of named queries in WSRR so that the API Connect system software can discover services in your registry. However, if only the lifecycles or classifications have been modified then this step is not required.

Overview

When API Connect connects to WSRR to discover services, a series of built-in search queries are used to carry out those searches. If your organization has a customized data model that is used inside WSRR, it is possible that the built-in queries are no longer valid, or do not return the results that you would expect.

To handle this situation, API Connect provides the option for you to overrule the built-in queries by defining a specific named query that is used to match your modified data model. When you search for services through API Connect, the system software looks to see whether the appropriate named queries exist in WSRR, and, if it finds them, they are used rather than the built-in queries.

The following table shows details about the various built-in queries that API Connect uses to interact with WSRR. You can work with your WSRR administrator to override that default behavior, if required. Depending on the customizations in your data model, you might have to override only one or two of the specific named queries that are described here, or, if extensive customizations exist, you might have to override all of them.

For each named query, the following table provides details of the specific name that you must use in WSRR to override the named query, and the XML query object that your WSRR administrator can copy into the WSRR named query as the starting point for the modifications to match the data model for your organization.

Named queries

The following table shows the named queries that you can define in WSRR so that you can override the default named query behavior. For information about how to load a named query into WSRR, see the IBM® WebSphere Service Registry and Repository Version 8.0 documentation (https://www-01.ibm.com/support/knowledgecenter/SSWLGF_8.0.0/maps/product_landing.html), and search for *Loading a named query configuration file* in WebSphere Service Registry and Repository (WSRR).

Note: The name of the query in WSRR must match the name that is shown in the table.

Table 1. Named queries. Named queries that you can define in WSRR

Name	Summary of named query	Link to information about the named query
`apimgmt_listServicesWithSOAP11Bindings`	This named query lists all of the capability versions that have SOAP 1.1 bindings with a name or description that matches %1, and a version number that matches %2.	Query to list all of the capability versions that have SOAP 1.1. bindings.
`apimgmt_listServicesWithSOAP12Bindings`	This named query lists all of the capability versions that have SOAP 1.2. bindings with a name or description that matches %1, and a version number that matches %2.	Query to list all of the capability versions that have SOAP 1.2. bindings.
`apimgmt_listOperationsOnCapabilityVersion`	This named query lists all of the operations on the provided web service of a capability version, where **bsrURI** equals %1.	Query to list all of the operations on a capability version.
`apimgmt_listSOAP11EndpointsOnCapabilityVersion`	This named query lists all of the endpoints on capability versions that have SOAP 1.1 bindings where **bsrURI** equals %1.	Query to list all of the endpoints on a capability version that have SOAP 1.1 bindings.
`apimgmt_listSOAP12EndpointsOnCapabilityVersion`	This named query lists all of the endpoints on capability versions that have SOAP 1.2 bindings where **bsrURI** equals %1.	Query to list all of the endpoints on a capability version that have SOAP 1.2 bindings
`apimgmt_wsdlDocumentOnCapabilityVersion`	This named query gets the WSDL document that represents the service that is provided by a capability version where **bsrURI** equals %1.	Query to get the WSDL document that represents the service that is provided by a capability version.
`apimgmt_artifactsOnCapabilityVersion`	This named query gets the artifacts that are related to a capability version where **bsrURI** equals %1.	Query to get the artifacts that are related to a capability version.

- **Query to list all of the capability versions that have SOAP 1.1. bindings**
  If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listServicesWithSOAP11Bindings`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a property query that lists all of the capability versions that have SOAP 1.1 bindings that match the specific parameters.
- **Query to list all of the capability versions that have SOAP 1.2. bindings**
  If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you

might have to add a named query, **`apimgmt_listServicesWithSOAP12Bindings`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a property query that lists all of the capability versions that have SOAP 1.2 bindings that match the specific parameters.

- **Query to list all of the operations on a capability version**
  If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listOperationsOnCapabilityVersion`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a property query that lists all of the operations on the provided web service of a specific capability version.
- **Query to list all of the endpoints on a capability version that have SOAP 1.1 bindings**
  If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listSOAP11EndpointsOnCapabilityVersion`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a graph query that lists all of the endpoints on a capability version with SOAP 1.1 bindings that match the specific parameters.
- **Query to list all of the endpoints on a capability version that have SOAP 1.2 bindings**
  If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listSOAP12EndpointsOnCapabilityVersion`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a graph query that lists all of the endpoints on a capability version with SOAP 1.2 bindings that match the specific parameters.
- **Query to get the WSDL document that represents the service that is provided by a capability version**
  If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_wsdlDocumentOnCapabilityVersion`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a graph query to get the Web Service Definition Language (WSDL) document that represents the service that is provided by a capability version.
- **Query to get the artifacts that are related to a capability version**
  If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_artifactsOnCapabilityVersion`**, to WebSphere Service Registry and Repository (WSRR). The named query runs a graph query that gets the artifacts that are related to a capability version.

Related tasks

- Adding a SOAP API definition by discovering a service from a registry
- Adding a WSRR registry for a SOAP API

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to list all of the capability versions that have SOAP 1.1. bindings

If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listServicesWithSOAP11Bindings`**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a property query that lists all of the capability versions that have SOAP 1.1 bindings that match the specific parameters.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name **`apimgmt_listServicesWithSOAP11Bindings`**

Description

This named query lists all of the capability versions that have SOAP 1.1 bindings with a name or description that matches %1, and a version number that matches %2.

If the capability version is related to a web service that has a SOAPBinding that is defined with the `transport` property set to 'http://schemas.xmlsoap.org/soap/http', the capability version is considered to have a SOAP 1.1 binding.

Returns

The named query returns a list of capability versions that have SOAP 1.1 bindings that match the specific input parameters. The list must contain all of the properties that are listed in the Properties section.

Standard query

When you create a named query, you must ensure that the query has the same number of parameters as the ones listed. In this case, the %1 parameter represents the search term to match to the name or description, and the %2 parameter represents the version number.

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>

<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration
../../schemas/NamedQueryConfiguration.xsd"
       type="propertyQuery">


<xpath>//GenericObject[(classifiedByAnyOf(.,'http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/
GovernanceEnablementModel#CapabilityVersion') and
         (matches(@name,'.*%1.*','i') or matches(@description,'.*%1.*','i')) and
          matches(@version,'%2'))
/gep63_providedWebServices(.)/sm63_ports(.)/sm63_binding(.)/sm63_wsdlBindings(.)/SOAPBinding[@transport=
'http://schemas.xmlsoap.org/soap/http']]
  </xpath>

  <properties>
    <property>bsrURI</property>
    <property>name</property>
    <property>lastModified</property>
    <property>version</property>
    <property>description</property>
    <property>owner</property>
    <property>creationTimestamp</property>
  </properties>

</query>
```

Properties

The query returns information about the following properties:

bsrURI
: A unique identifier for the resource.

name
: The name of the resource in WSRR.

lastModified
: The date the resource was modified.

version
: The version number of the document in WSRR.

description
: A description of the resource in WSRR.

owner
: The owner of the resource.

creationTimestamp
: The time that the resource was created.

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource>
  <properties>
```

```
   <property name="bsrURI" value="ebe4dfeb-136e-4e73.a3ad.98cac598ada8"/>
   <property name="name" value="ServiceVersion1a"/>
   <property name="lastModified" value="1385466638574"/>
   <property name="version" value="1.2"/>
   <property name="description" value="service version 1a"/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1378115342058"/>
  </properties>
 </resource>
 <resource>
  <properties>
   <property name="bsrURI" value="5bb15f5b-7eb4-44d0.afc2.0f26430fc296"/>
   <property name="name" value="ServiceVersion4"/>
   <property name="lastModified" value="1386762664335"/>
   <property name="version" value=""/>
   <property name="description" value="Service for Round4XSDTest. Contains SOAP 1.1 and 1.2 bindings"/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1378115518547"/>
  </properties>
 </resource>
 <resource>
  <properties>
   <property name="bsrURI" value="e6328be6-ebb5-4514.a332.b887efb832a2"/>
   <property name="name" value="ServiceVersionAccCrt"/>
   <property name="lastModified" value="1384875705711"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1384858293227"/>
  </properties>
 </resource>
 <resource>
  <properties>
   <property name="bsrURI" value="3b37a13b-bec8-485b.b8bc.85f86185bc46"/>
   <property name="name" value="ServiceVersionKaiser1"/>
   <property name="lastModified" value="1385034623940"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1385034623246"/>
  </properties>
 </resource>
</resources>
```

Related reference

- Query to list all of the capability versions that have SOAP 1.2. bindings

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to list all of the capability versions that have SOAP 1.2. bindings

If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listServicesWithSOAP12Bindings`**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a property query that lists all of the capability versions that have SOAP 1.2 bindings that match the specific parameters.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name **`apimgmt_listServicesWithSOAP12Bindings`**

Description

This named query lists all of the capability versions that have SOAP 1.2. bindings with a name or description that matches %1, and a version number that matches %2.

If the capability version is related to a web service that has a WSDL binding that is defined with an extension that has the **`extensionNamespace`** property set to 'http://schemas.xmlsoap.org/wsdl/soap12/', the capability version is considered to have a SOAP 1.2 binding.

Returns

The named query returns a list of capability versions that have SOAP 1.2 bindings that match the specific input parameters. The list must contain all of the properties that are listed in the Properties section.

Standard query

When you create a named query, you must ensure that the query has the same number of parameters as the ones listed. In this case, the %1 parameter represents the search term to match to the name or description, and the %2 parameter represents the version number.

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>

<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration
../../schemas/NamedQueryConfiguration.xsd"
       type="propertyQuery">

  
<xpath>//GenericObject[(classifiedByAnyOf(.,'http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/
GovernanceEnablementModel#CapabilityVersion') and
         (matches(@name,'.*%1.*','i') or matches(@description,'.*%1.*','i')) and
          matches(@version,'%2'))
/gep63_providedWebServices(.)/sm63_ports(.)/sm63_binding(.)/sm63_wsdlBindings(.)/extensions[@extensionNa
mespace='http://schemas.xmlsoap.org/wsdl/soap12/']]
  </xpath>

  <properties>
    <property>bsrURI</property>
    <property>name</property>
    <property>lastModified</property>
    <property>version</property>
    <property>description</property>
    <property>owner</property>
    <property>creationTimestamp</property>
  </properties>

</query>
```

Properties

The query returns information about the following properties:

bsrURI
  A unique identifier for the resource.

name
  The name of the resource in WSRR.

lastModified
  The date the resource was modified.

version
  The version number of the document in WSRR.

description
  A description of the resource in WSRR.

owner
  The owner of the resource.

creationTimestamp

The time that the resource was created.

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource>
  <properties>
   <property name="bsrURI" value="5bb15f5b-7eb4-44d0.afc2.0f26430fc296"/>
   <property name="name" value="ServiceVersion4"/>
   <property name="lastModified" value="1386762664335"/>
   <property name="version" value=""/>
   <property name="description" value="Service for Round4XSDTest. Contains SOAP 1.1 and 1.2 bindings"/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1378115518547"/>
  </properties>
 </resource>
</resources>
```

Related reference

- Query to list all of the capability versions that have SOAP 1.1. bindings

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to list all of the operations on a capability version

If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **apimgmt_listOperationsOnCapabilityVersion**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a property query that lists all of the operations on the provided web service of a specific capability version.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name `apimgmt_listOperationsOnCapabilityVersion`

Description

This named query lists all of the operations on the provided web service of a capability version, where **bsrURI** equals %1.

Returns

The named query returns a list of the operations on the provided web service of a specific capability version. The list must contain all of the properties that are listed in the Properties section.

Standard query

When you create a named query, you must ensure that the query has the same number of parameters as the ones listed. In this case, the %1 parameter represents the **bsrURI** of the capability version.

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>

<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration 
../../schemas/NamedQueryConfiguration.xsd"
       type="propertyQuery">


<xpath>//GenericObject[@bsrURI='%1']/gep63_providedWebServices(.)/sm63_serviceInterfaces(.)/sm63_operati
ons(.)</xpath>

  <properties>
    <property>bsrURI</property>
    <property>name</property>
    <property>lastModified</property>
    <property>version</property>
    <property>description</property>
    <property>owner</property>
    <property>creationTimestamp</property>
  </properties>

</query>
```

Properties

The query returns information about the following properties:

bsrURI
    A unique identifier for the resource.
name
    The name of the resource in WSRR.
lastModified
    The date the resource was modified.
version
    The version number of the document in WSRR.
description
    A description of the resource in WSRR.
owner
    The owner of the resource.
creationTimestamp
    The time that the resource was created.

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource>
  <properties>
   <property name="bsrURI" value="2e67122e-6922-4241.9dc9.75500675c99d"/>
   <property name="name" value="echoVoid"/>
   <property name="lastModified" value="1384274449578"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1384189590531"/>
  </properties>
 </resource>
 <resource>
  <properties>
   <property name="bsrURI" value="dac997da-68fe-4ec7.a74d.1460ae144d54"/>
   <property name="name" value="echoStringMultiOccurs"/>
   <property name="lastModified" value="1384274449580"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="creationTimestamp" value="1384189590739"/>
  </properties>
 </resource>
</resources>
```

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to list all of the endpoints on a capability version that have SOAP 1.1 bindings

If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listSOAP11EndpointsOnCapabilityVersion`**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a graph query that lists all of the endpoints on a capability version with SOAP 1.1 bindings that match the specific parameters.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name **`apimgmt_listSOAP11EndpointsOnCapabilityVersion`**

Description

This named query lists all of the endpoints on capability versions that have SOAP 1.1 bindings where **`bsrURI`** equals %1.

Given the **`bsrURI`** of a service, this query returns the SOAP endpoints that are associated with that service.

Returns

This named query returns a resources element that contains multiple resource items, whose type is SOAPAddress.

Standard query

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration 
../../schemas/NamedQueryConfiguration.xsd" 
       type="graphQuery">
 <xpath>//GenericObject[@bsrURI='%1']/gep63_provides(.)/gep63_availableEndpoints(.)/sm63_soapAddress(.)
</xpath>
 <depth>0</depth>
</query>
```

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource bsrURI="9e38129e-56a9-49a7.8501.43636e43016b" type="SOAPAddress">
  <properties>
   <property name="bsrURI" value="9e38129e-56a9-49a7.8501.43636e43016b"/>
   <property name="name" value="Round4XSDTestSoap_SOAPAddress"/>
   <property name="namespace" value="http://soapinterop.org/"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="lastModified" value="1384189269191"/>
   <property name="creationTimestamp" value="1384189269191"/>
   <property name="lastModifiedBy" value="wasadmin"/>
   <property name="location" value="http://www.example.com/interop/r4/wsdl-xsd"/>
  </properties>
  <relationships>
   <relationship name="_container" targetBsrURI="1d20531d-4ef1-4137.bc98.b9d5abb998ef" 
targetType="WSDLPort"/>
  </relationships>
  <classifications>
   <classification 
uri="http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Offline"/>
  </classifications>
 </resource>
</resources>
```

Related reference

- Query to list all of the endpoints on a capability version that have SOAP 1.2 bindings

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to list all of the endpoints on a capability version that have SOAP 1.2 bindings

If you are using a customized Governance Enablement Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_listSOAP12EndpointsOnCapabilityVersion`**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a graph query that lists all of the endpoints on a capability version with SOAP 1.2 bindings that match the specific parameters.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name **`apimgmt_listSOAP12EndpointsOnCapabilityVersion`**

Description

This named query lists all of the endpoints on capability versions that have SOAP 1.2 bindings where **`bsrURI`** equals %1.

Given the **`bsrURI`** of a service, this query returns the SOAP endpoints that are associated with that service.

Returns

This named query returns a resources element that contains multiple resource items, whose type is ExtensionLogicalObject.

Standard query

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration 
../../schemas/NamedQueryConfiguration.xsd" 
```

```
        type="graphQuery">

<xpath>//GenericObject[@bsrURI='%1']/gep63_provides(.)/gep63_availableEndpoints(.)/sm63_extensionLogical
Object(.)</xpath>
 <depth>0</depth>
</query>
```

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource bsrURI="1506d115-7bbe-4e4e.a983.d3be7bd383ff" type="ExtensionLogicalObject">
  <properties>
   <property name="bsrURI" value="1506d115-7bbe-4e4e.a983.d3be7bd383ff"/>
   <property name="name" value="Round4XSDTestSoap12"/>
   <property name="namespace" value="http://soapinterop.org/"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="lastModified" value="1384189269191"/>
   <property name="creationTimestamp" value="1384189269191"/>
   <property name="lastModifiedBy" value="wasadmin"/>
   <property name="extensionNamespace" value="http://schemas.xmlsoap.org/wsdl/soap12/"/>
   <property name="address_location" value="http://www.example.com/interop/r4/wsdl-xsd12"/>
   <property name="address" value=""/>
   <property name="address_xmlns_soap12" value="http://schemas.xmlsoap.org/wsdl/soap12/"/>
  </properties>
  <relationships>
   <relationship name="_container" targetBsrURI="f92ba6f9-7c8c-4cf1.9e87.d999ccd9877b" 
targetType="WSDLPort"/>
  </relationships>
  <classifications>
   <classification 
uri="http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Offline"/>
  </classifications>
 </resource>
</resources>
```

Related reference

- Query to list all of the endpoints on a capability version that have SOAP 1.1 bindings

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Query to get the WSDL document that represents the service that is provided by a capability version

If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **apimgmt_wsdlDocumentOnCapabilityVersion**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a graph query to get the Web Service Definition Language (WSDL) document that represents the service that is provided by a capability version.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name `apimgmt_wsdlDocumentOnCapabilityVersion`

Description

This named query gets the WSDL document that represents the service that is provided by a capability version where `bsrURI` equals %1.

Returns

This named query returns information about the WSDL document that represents the service that is provided by a capability version.

Standard query

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration 
../../schemas/NamedQueryConfiguration.xsd" 
        type="graphQuery">
 <xpath>//GenericObject[@bsrURI='%1']/gep63_providedWebServices(.)/sm63_wsdlServices(.)/document(.)
</xpath>
 <depth>0</depth>
</query>
```

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource bsrURI="e00968e0-3e36-4663.bdb5.da8608dab5f8" type="WSDLDocument" 
governanceRootBsrURI="9df9239d-1fd4-44aa.9708.3d3e453d089f">
  <properties>
   <property name="bsrURI" value="e00968e0-3e36-4663.bdb5.da8608dab5f8"/>
   <property name="name" value="round4R4.wsdl"/>
   <property name="namespace" value="http://soapinterop.org/"/>
   <property name="version" value=""/>
   <property name="description" value=""/>
   <property name="owner" value="wasadmin"/>
   <property name="lastModified" value="1384189601398"/>
   <property name="creationTimestamp" value="1384189269191"/>
   <property name="lastModifiedBy" value="wasadmin"/>
   <property name="encoding" value="utf-8"/>
   <property name="location" value="round4R4.wsdl"/>
   <property name="contentSize" value="47355"/>
   <property name="xmlns_s0" value="http://soapinterop.org/"/>
   <property name="xmlns_s2" value="http://soapinterop.org"/>
   <property name="xmlns_http" value="http://schemas.xmlsoap.org/wsdl/http/"/>
   <property name="xmlns_tm" value="http://microsoft.com/wsdl/mime/textMatching/"/>
   <property name="xmlns_s1" value="http://soapinterop.org/xsd"/>
   <property name="xmlns_s3" value="http://soapinterop.org/echoheader/"/>
   <property name="xmlns_soapenc" value="http://schemas.xmlsoap.org/soap/encoding/"/>
   <property name="xmlns_soap" value="http://schemas.xmlsoap.org/wsdl/soap/"/>
   <property name="xmlns_soap12" value="http://schemas.xmlsoap.org/wsdl/soap12/"/>
   <property name="xmlns_s" value="http://www.w3.org/2001/XMLSchema"/>
   <property name="xmlns_mime" value="http://schemas.xmlsoap.org/wsdl/mime/"/>
   <property name="xmlns" value="http://schemas.xmlsoap.org/wsdl/"/>
   <property name="xsdTargetNamespaces" value="http://soapinterop.org/"/>
   <property name="xsdTargetNamespaces" value="http://soapinterop.org"/>
   <property name="xsdTargetNamespaces" value="http://soapinterop.org/echoheader/"/>
   <property name="xsdTargetNamespaces" value="http://soapinterop.org/xsd"/>
  </properties>
  <relationships/>
  <classifications>
   <classification 
uri="http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Offline" 
governanceState="true"/>
  </classifications>
 </resource>
</resources>
```

Related information

Query to get the artifacts that are related to a capability version

If you are using a customized Governance Enablment Profile (GEP) model then, depending on the degree of customization, you might have to add a named query, **`apimgmt_artifactsOnCapabilityVersion`**, to WebSphere® Service Registry and Repository (WSRR). The named query runs a graph query that gets the artifacts that are related to a capability version.

Note: You have to add the named query to WSRR only if you have to update the standard query, detailed in this topic, to match your customized GEP. If the standard query is valid unchanged then no action is required.

Query name

When you load the named query into WSRR, the query must have the name **`apimgmt_artifactsOnCapabilityVersion`**

Description

This named query gets the artifacts that are related to a capability version where **`bsrURI`** equals %1.

Returns

This named query returns information about the artifacts that are related to a capability version.

Standard query

Use the following example query as a basis for the named query to load into WSRR:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.ibm.com/xmlns/prod/serviceregistry/7/0/NamedQueryConfiguration 
../../schemas/NamedQueryConfiguration.xsd" 
        type="graphQuery">
 <xpath>//GenericObject[@bsrURI='%1']/ale63_artifacts(.)</xpath>
 <depth>0</depth>
</query>
```

Example result XML

The following code is an example of the XML that is returned by the query:

```
<resources>
 <resource bsrURI="d6fe85d6-83c9-4967.ae49.e32b99e349d6" type="WSDLDocument" 
governanceRootBsrURI="9bd9849b-bca4-44f5.b101.0e5c350e0189">
  <properties>
   <property name="bsrURI" value="d6fe85d6-83c9-4967.ae49.e32b99e349d6"/>
   <property name="name" value="accountDetail.wsdl"/>
   <property name="namespace" value="http://www.example.org/accountDetail/"/>
   <property name="version" value=""/>
   <property name="description" value="Dupe"/>
   <property name="owner" value="wasadmin"/>
   <property name="lastModified" value="1385027376142"/>
   <property name="creationTimestamp" value="1385027262195"/>
   <property name="lastModifiedBy" value="wasadmin"/>
   <property name="encoding" value="UTF-8"/>
   <property name="location" value="accountDetail.wsdl"/>
   <property name="contentSize" value="8671"/>
   <property name="xmlns_xsd" value="http://www.w3.org/2001/XMLSchema"/>
   <property name="xmlns_wsdl" value="http://schemas.xmlsoap.org/wsdl/"/>
   <property name="xmlns_soap" value="http://schemas.xmlsoap.org/wsdl/soap/"/>
   <property name="xmlns_tns" value="http://www.example.org/accountDetail/"/>
   <property name="xmlns_bons1" value="http://retailAccount"/>
   <property name="xsdTargetNamespaces" value="http://www.example.org/accountDetail/"/>
  </properties>
  <relationships/>
  <classifications>
   <classification 
uri="http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Offline" 
governanceState="true"/>
  </classifications>
 </resource>
</resources>
```

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 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	• The required client identification has not been successfully provided. • User authentication failed or did not take place. • The application is not registered with the plan that is used.
403 Forbidden	The application is not active.
404 Not Found	• Information for the provider organization or environment was not found. • The API URL was not found in the organization or environment.
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 Managing your Products in the API Manager UI and Managing API Products using the developer toolkit.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Troubleshooting your billing configuration

Some issues can occur when you are setting up your billing plan for subscriptions in IBM® API Connect.

You can use the following links to browse the issues:

1. I cannot see the Billing tab when I look at my Product when the Admin section selected.
2. I cannot subscribe to a billing Plan with my Stripe account.
3. My customer payment is not completing with Stripe, but the Product is still being accessed.
4. My Product with billing is returning an error when I try to configure it with my Stripe account.
5. My customer is not able to access the Product with billing, but was previously subscribed to it with a Stripe account.

I cannot see the Billing tab when I look at my Product when the Admin section selected.

You must have an Enterprise license for API Connect to use the billing features.You must have an Enterprise IBM Cloud license to use the billing features.

- Upgrade your license to the correct level to view the Billing tab and access the billing features.

I cannot subscribe to a billing Plan with my Stripe account.

- Stripe limits each customer to a maximum of 25 subscriptions. Ensure that you have not exceeded this limit. If so, you can add this subscription only if you remove another subscription.
  Tip: You can exceed the 25-subscription limit by creating an extra organization, if you do not want to remove a subscription.

My customer payment is not completing with Stripe, but the Product is still being accessed.

When payment issues occur with the credit card payment provider, the credit card subscription is canceled, but the API Connect subscription is not canceled. You can suspend an application until the payment issues are resolved by completing the following steps:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Select Dashboard in the navigation.
3. Select the catalog for the Product.
4. Select Community tab.
5. Select Applications.
6. Select the Manage icon (three vertically aligned dots) > Suspend application for the application that you want to stop. The owner receives a notification that the application is suspended.

To restart a suspended application, complete the following steps:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Select Dashboard in the navigation.
3. Select the catalog for the Product.
4. Select Community tab.
5. Select Applications.
6. Select the Manage icon (three vertically aligned dots) > Resume application for the subscription that you want to stop. The owner receives a notification that the subscription is resumed.

My Product with billing is returning an error when I try to configure it with my Stripe account.

The most common cause of this is that your Developer Portal cluster servers or your Management cluster servers cannot communicate to Stripe.
To resolve this, ensure that the Stripe API is enabled with HTTPS communication with your Management cluster servers and the Developer Portal on port 443. See Firewall requirements for more information about this requirement.

My customer is not able to access the Product with billing, but was previously subscribed to it with a Stripe account.

There is more than one possible cause and resolution for this situation.

- The credit card transaction might not be completing correctly. If this situation happens, the subscription might be canceled with Stripe, but the API Connect subscription is not canceled. If you want to cancel an account in API Connect, you can complete that process in the Community Management area of your Developer Portal administration interface.
    - By default, Stripe attempts to charge a credit card for a subscription three times when billing attempts fail. The subscription with the Stripe account is canceled after the three unsuccessful attempts.
      You can track credit card issues more easily by changing your Stripe settings to mark the account as unpaid. To change the behavior of the Stripe account, complete the following steps:
        1. Log in to your Stripe account.
        2. Select Subscriptions in the navigation.
        3. Select the Settings tab.
        4. In the drop-down menu that follows *Then finally*, select Mark subscription as unpaid.
        5. Select Save to save your changes.
    - The account might be in suspended state. An account can be put in suspended state within the Community Management area of the API Manager if you want to temporarily disable it. This account might be in this state because someone changed the account to this state after payment issues. You can restore a suspended account in the Community Management area of the API Manager.
    - Stripe might have an issue with billing that occurred after the Product was migrated. The subscriptions that fail to migrate are put in the category of "Blocked migrations."
- A recent change to a Product with billing might have triggered an action that was not completed correctly. Seemingly minor changes to the Product can start a number of actions that can overload a server. For example, when you replace a version of the Product with a new version, it starts an update to the Stripe server for every account that is subscribed to that Product. If the action fails to complete, the plan's subscription status might not be reflected correctly with Stripe.
  You can identify this situation by viewing the Billing tab in the Community Management area of the API Manager. In the Issues section, it shows a number of blocked jobs for the Stripe integration. You can retry the job later to determine whether it was a temporary issue by completing the following steps:
    1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
    2. Select Dashboard in the navigation.
    3. Select the catalog for the Product.
    4. Select Community Management in the navigation.
    5. Select the Billing tab.
    6. Locate the job that was blocked.
    7. Select the Manage icon (three vertically aligned dots) > Retry Blocked Jobs for the blocked job.

  The blocked job is retried. The number of blocked jobs resets to 0, and the Product is accessible if the job completes successfully.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

API Manager tutorials

Tutorials for using API Manager.

Before you begin

Because you are working online, using API Manager, to complete the following tutorials you must be the Organization Manager of an IBM® API Connect account. Or, you received an invite to join API Connect as an Administrator. See Administering Developer 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.

About this task

In addition to its other APIs, as defined in Tutorials for working with API definitions that call an existing endpoint, BankA has existing SOAP APIs that it wants to secure and expose.

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.

- V5.0.7+ **API Manager tutorials for V5.0.7 and later**
  Tutorials for using the API Manager in IBM API Connect Version 5.0.7 and later.
- V5.0.6 and earlier **API Manager tutorials for V5.0.6 and earlier**
  Tutorials for using the API Manager in IBM API Connect Version 5.0.6 and earlier.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

API Manager tutorials for V5.0.7 and later

Tutorials for using the API Manager in IBM® API Connect Version 5.0.7 and later.

Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.6 and earlier, see API Manager tutorials for V5.0.6 and earlier.

- **Tutorial for creating a SOAP API**
  This tutorial shows you how to create an API definition by using a SOAP service's Web Service Definition Language (WSDL) in IBM API Connect Version 5.0.7 and later. This API definition allows simplified access to, and management of, access for 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, in IBM API Connect Version 5.0.7 and later.
- V5.0.8+ **Tutorial: Defining a subscription Plan with pricing for your API Product**
  This tutorial shows you how to define a Plan with pricing that consumers can use to subscribe to your API Product, in IBM API Connect Version 5.0.8 and later.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7+

Tutorial for creating a SOAP API

This tutorial shows you how to create an API definition by using a SOAP service's Web Service Definition Language (WSDL) in IBM® API Connect Version 5.0.7 and later. This API definition allows simplified access to, and management of, access for the SOAP service.

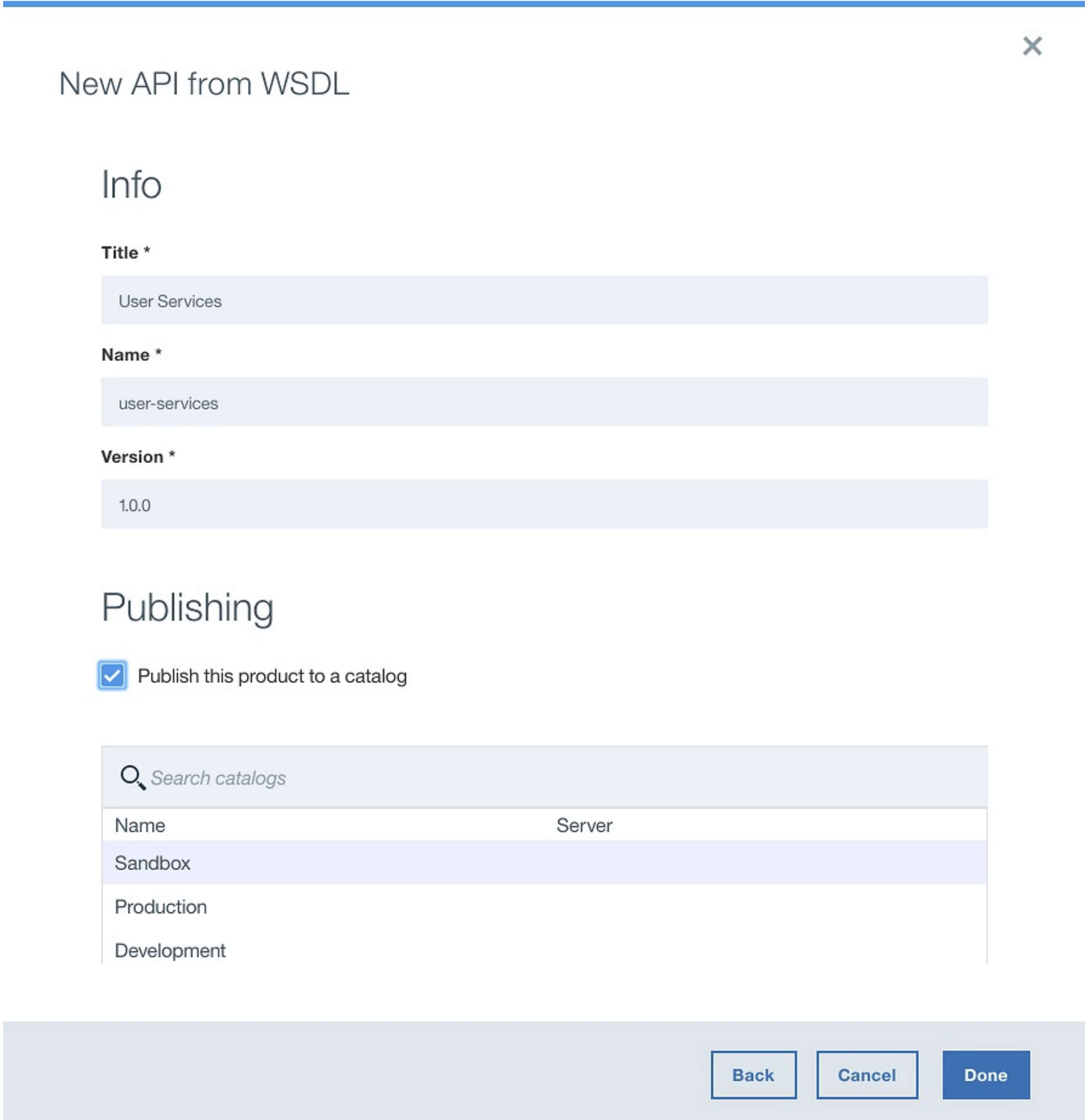
About this tutorial

Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.6 and earlier, see API Manager tutorials for V5.0.6 and earlier.
In API Manager, you create an API by importing the WSDL for an existing SOAP service. 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.

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. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane and then click the APIs tab. The APIs tab opens.
4. Click Add > New OpenAPI from SOAP service.
5. Click Upload file.
6. Select the `AccountService.wsdl` file from your file system
7. Specify the service to include, and then create the API definition.
   a. Select the AccountService SOAP service and then click Add a product.
   b. In the Title field, enter `User Services`.
   c. Leave the Name and Version fields unchanged.
   d. Ensure that the Publish this product to a catalog check box is selected and then select Sandbox as the target Catalog.



New API from WSDL

Info

Title * : User Services

Name * : user-services

Version * : 1.0.0

Publishing

☑ Publish this product to a catalog

Name	Server
Sandbox	
Production	
Development	

   e. Click Done. The Design tab for the draft of your API definition opens.

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

To add your SOAP API to a Product and Plan, and then test it, complete the following steps:

1. Click the Assemble tab. The assemble view opens.
2. Click the Test icon ▶. The test tool opens.
3. If you have used the test tool before, click Change setup.
4. In the Catalog field, select your Sandbox Catalog.
5. In the Product field, select your User Services Product and then click Republish product to publish your Product so that it can be tested.
6. Click Next.
   Notice that the Setup section indicates that automatic subscription is used: **`Sandbox, User Services 1.0.0, using automatic subscription`**. Automatic subscription is enabled by default for the Sandbox Catalog; so you do not have to specify a Plan or application when testing. A test application which is automatically subscribed to all the Plans in the Catalog will be used, with a pre-supplied client ID and client secret.
7. In the Operation field, select getBalance.
8. 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 "3". However, unlike the SOAP service, the API requires identification using a Client ID, which is enforced by the gateway server.
9. 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

Related information

- Tutorial: Creating a REST API definition that invokes an existing SOAP service
- Adding a SOAP API definition
- Adding a SOAP API definition by using a WSDL file
- Adding a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

DataPower Gateway only V5.0.7+

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, in IBM® API Connect Version 5.0.7 and later.

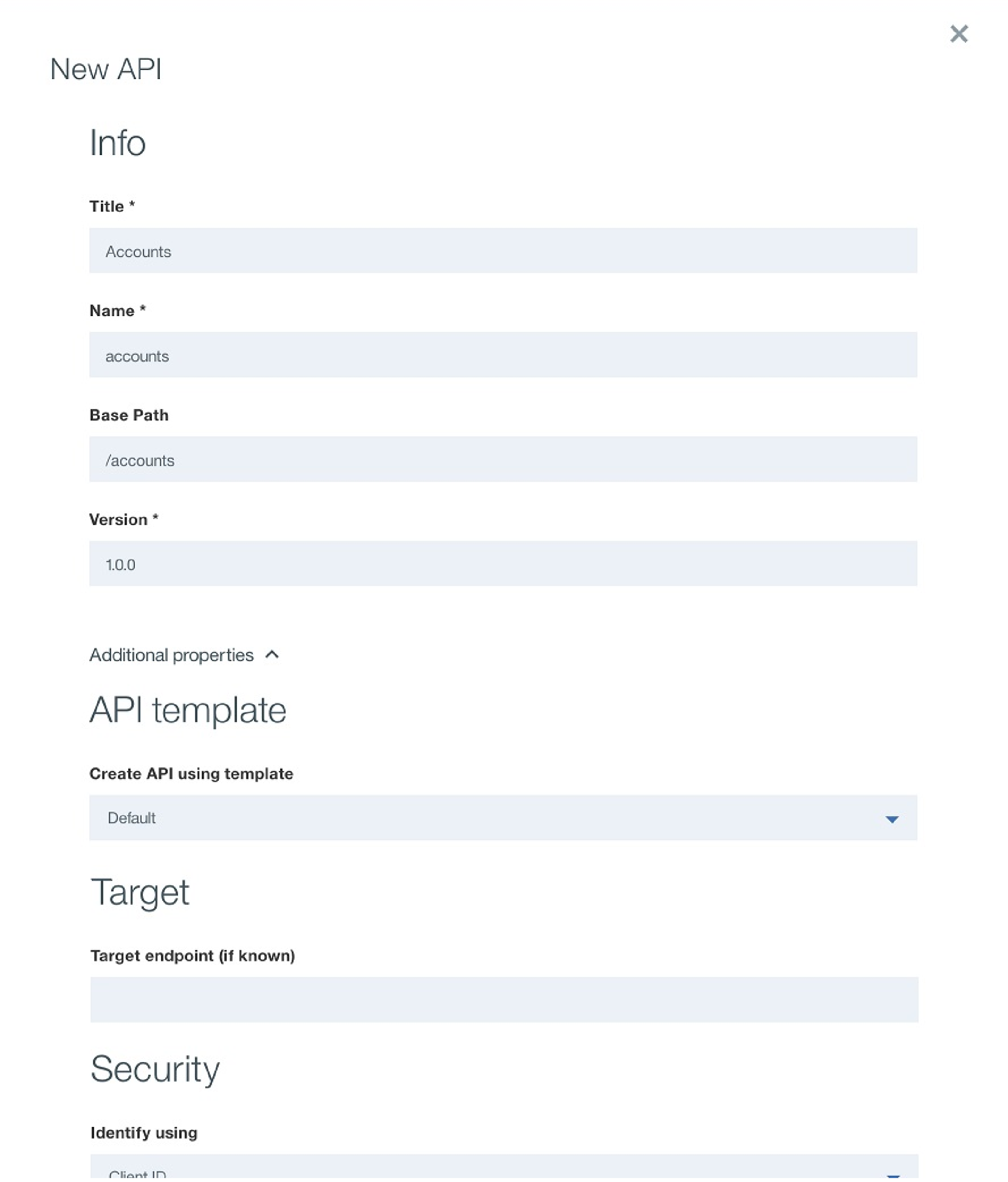
About this tutorial

Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.6 and earlier, see API Manager tutorials for V5.0.6 and earlier.
In API Manager, you will create a REST API that accesses a SOAP API to make data from the existing SOAP service available. This tutorial uses the same SOAP service as the Tutorial for creating a SOAP API tutorial, but exposes it in a different way.

Setting up a REST API definition

To set up a REST API, complete the following steps:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane and then click the APIs tab. The APIs tab opens.
3. Click Add > New API.
4. Specify basic information about the API.
   a. In the Title field, enter `Accounts`.
   b. Leave the Name field as `accounts` when it is filled while you enter your title.
   c. Leave the Base Path field as `/accounts`.
   d. Leave the Version field as `1.0.0`.
5. Expand Additional properties to specify additional properties for the API.
   a. From the API template field, select Default to indicate that you want to use the default template to create the API definition.
   b. Leave the remaining fields unchanged.



New API

Info

Title *: Accounts

Name *: accounts

Base Path: /accounts

Version *: 1.0.0

Additional properties

API template

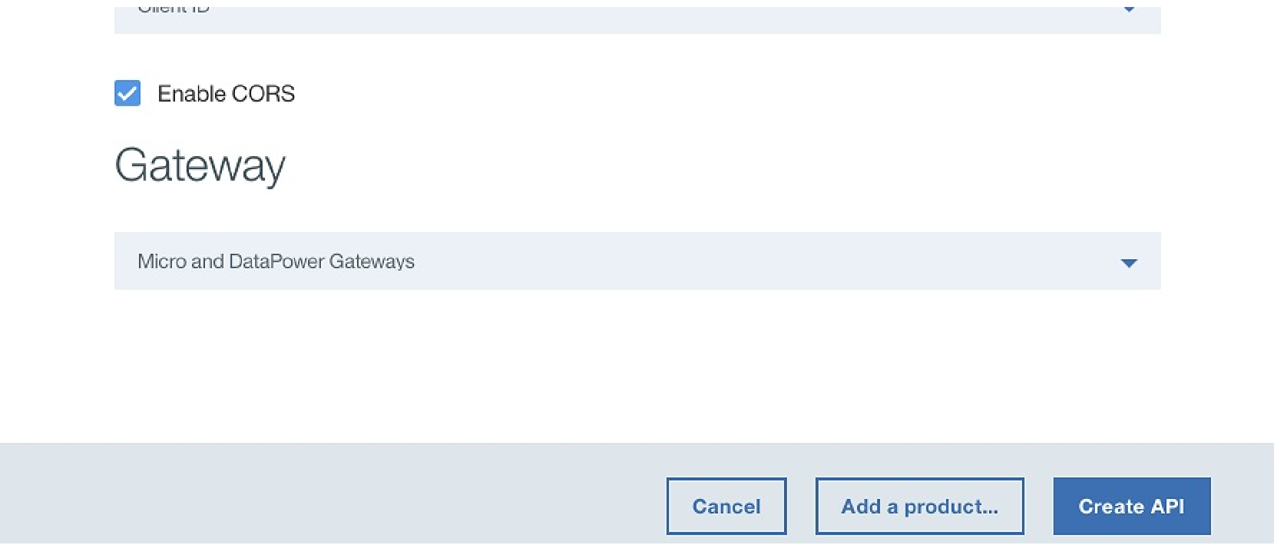
Create API using template: Default

Target

Target endpoint (if known):

Security

Identify using: Client ID



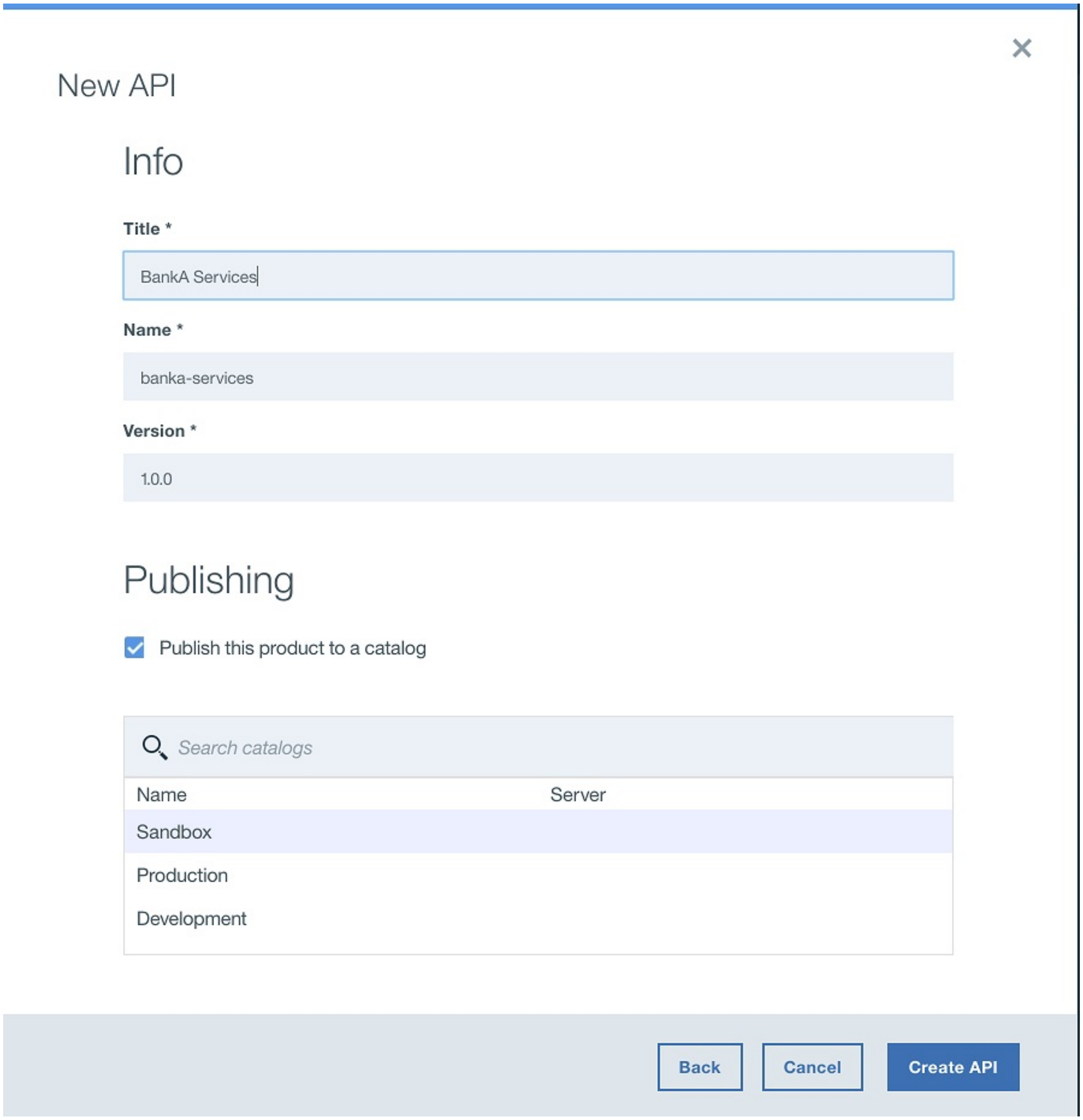
Client ID

☑ Enable CORS

Gateway

Micro and DataPower Gateways

6. Add your API to a new Product and then create the API definition.
   a. Click Add a product.
   b. In the Title field, enter `BankA Services`.
   c. Leave the Name and Version fields unchanged.
   d. Ensure that the Publish this product to a catalog check box is selected and then select Sandbox as the target Catalog.


New API

Info

Title *

BankA Services

Name *

banka-services

Version *

1.0.0

Publishing

☑ Publish this product to a catalog

Search catalogs

Name	Server
Sandbox	
Production	
Development	

e. Click Create API. The Design tab for the draft of your API definition opens.

7. In the Definitions section, click the Add Definition icon ⊕ and then expand the new Definition by clicking it.
8. In the Name field rename the Definition to `Account Output`.
9. In the Properties subsection of the Definition is a single property called new-property-1. Rename the property to `Balance`, and in the TYPE column select double. Select the check box in the * column to mark the property as required.
   Marking a property as required is indicated to users in the OpenAPI (Swagger 2.0) definition of the API and can be enforced by a validate policy.

10. In the Paths section, click the Add Path icon ⊕, and then expand the new Path by clicking it.
11. In the Path field of your newly created Path, replace the contents with `/balance`.
12. In the Parameters subsection of the Path, expand the GET /balance operation by clicking it.
13. In the Parameters subsection for the GET /balance operation, click Add Parameter and then click Add new parameter.
14. Name your new parameter `customer_id`, and in the TYPE column select number-double. Select the check box in the REQUIRED column to mark the property as required.
    Marking a property as required is indicated to users in the OpenAPI (Swagger 2.0) definition of the API, is enforced by validate policies, and results in the test tool always generating the parameter as part of a sample API call.

15. In the Schema column of the 200 OK response in the Responses subsection, select your Account Output definition.

16. Click the Save icon 💾 to save your changes.

Adding and configuring your web service invocation

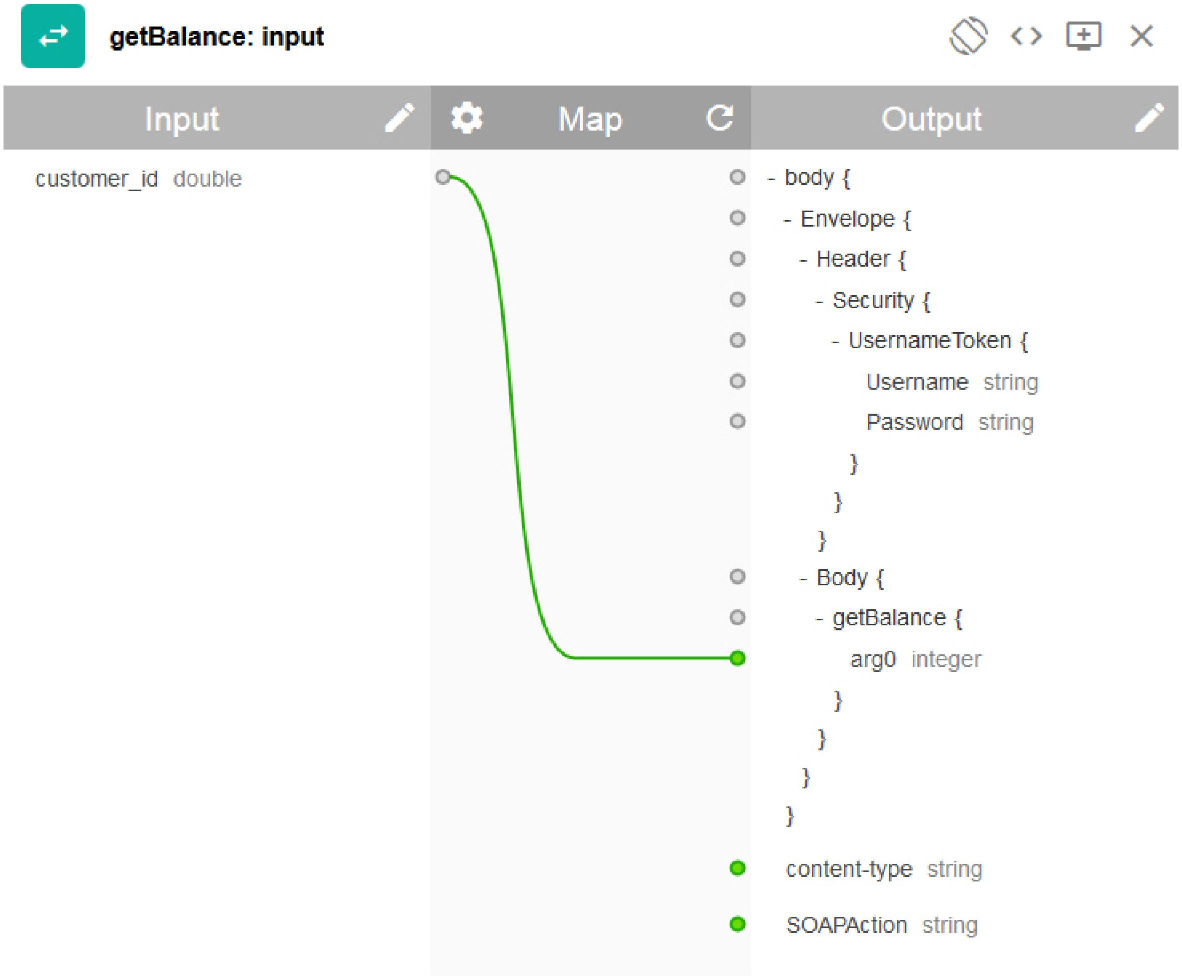
To add and configure the invoke and map policies that integrate your web service into your API definition, complete the following steps:

1. Download the SOAP WSDL file AccountService.txt. Rename this file `AccountService.wsdl`.
2. In the Services section, click the Add service icon ⊕. The "Import web service from WSDL" window opens.
3. Click Upload file.
4. Select the `AccountService.wsdl` file from your file system
5. Select the AccountService SOAP service and then click Done. In the Services section, the web service is listed.
6. Click the Assemble tab and then ensure that DataPower Gateway policies is selected.
7. Delete the existing invoke policy on the canvas by hovering your cursor over the policy and then clicking the Delete policy icon 🗑.
8. From the palette, drag the getBalance web service onto the dashed box that is displayed on the canvas. An invoke policy and two map policies are placed in the assembly. The first map policy assigns variables to the input of your web service invocation, while the second policy assigns outputs of your web service invocation to variables. The outputs of the first map and the inputs of the second map are generated from the WSDL that you uploaded.
9. Click the getBalance: input map policy and then click the Edit inputs icon ✎ in the Input column of the property sheet.
10. Click + input.
11. Configure the input according to the following table:

Table 1. Web service input

Property	Value
Context variable	`request.parameters.customer_id`
Name	`customer_id`
Content type	`none`
Definition	`double`

12. Click Done.
13. Click the circle corresponding to customer_id double on the input side and then click the circle corresponding to arg0 integer on the output side.



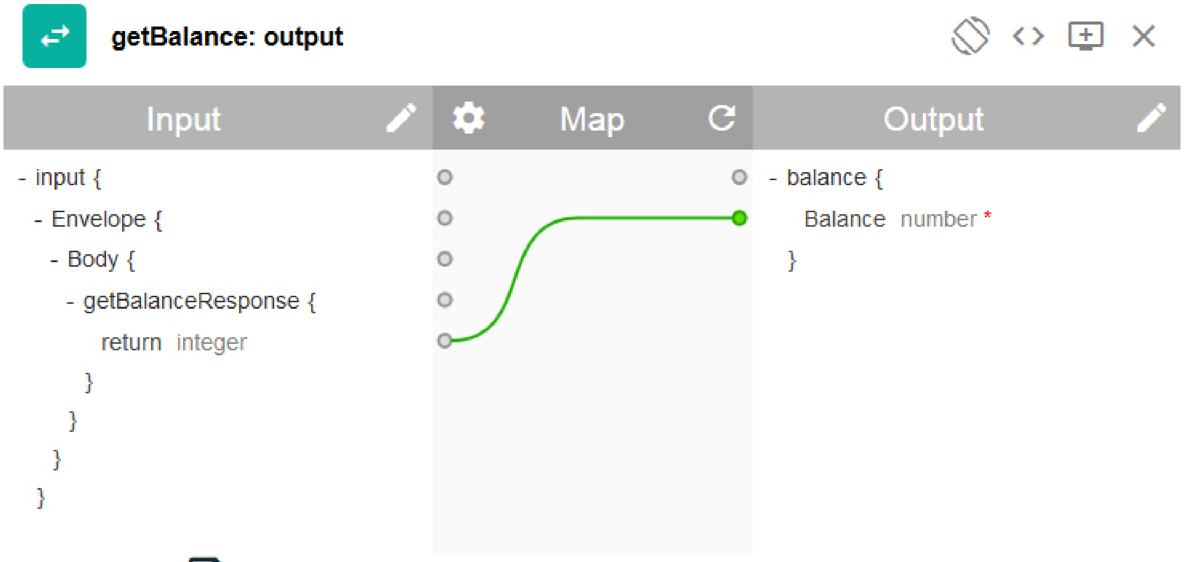
14. Close the property sheet.
15. Click the getBalance: output map policy in the palette and then click the Edit outputs icon ✎ in the Output column of the property sheet.
16. Click + output.

17. Configure the input according to the following table:

Table 2. Web service output

Property	Value
Context variable	`message.body`
Name	`balance`
Content type	`none`
Definition	`#/definitions/Account Output`

18. Click Done.
19. Click the circle corresponding to return integer on the input side and then click the circle corresponding to Balance number on the output side.



20. Click the Save icon to save your changes.

You have included the web service invocation in your assembly and mapped an input parameter to the appropriate part of the SOAP request and mapped the appropriate part of the SOAP response to a JSON output.

Testing your API definition

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. If you have used the test tool before, click Change setup.
3. In the Catalog field, select your Sandbox Catalog.
4. In the Product field, select your BankA Services Product and then click Republish product to publish your Product so that it can be tested.
5. Click Next.
6. In the Operation field, select get /balance.
7. In the customer_id field, enter `12345`.
8. 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:

- 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

- Adding a SOAP API definition
- Adding a SOAP API definition by using a WSDL file

- Adding a REST API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Tutorial: Defining a subscription Plan with pricing for your API Product

This tutorial shows you how to define a Plan with pricing that consumers can use to subscribe to your API Product, in IBM® API Connect Version 5.0.8 and later.

About this tutorial

In API Manager, you will create a subscription Plan with pricing for your API Product.

Prerequisites

To complete this tutorial, ensure that you have the following prerequisites:

- User ID with Administrative privileges to your API Manager account.
- IBM API Connect version 5.0.8.0, or later, for the on-premises platform.
- APIs that are being managed in an API Connect Developer Portal.
- Internet access that allows your API Connect management server to communicate with your Stripe account. See Firewall requirements for more information about this requirement.
- A Stripe test account for the credit card setup. If you do not have a Stripe account, you can create one during this tutorial. Stripe accounts (including test accounts) can be created at https://dashboard.stripe.com/register. Stripe provides the credit card payment processing for the API Connect transactions, and integrates into the API Connect Developer Portal through an iFrame.
  Important:
  There is no sharing of data between Stripe accounts, and there is no sharing of data between a single account's test keys and production keys. If you plan to use test keys for a Stripe account, make sure that you use them on a test provider organization. Only use the live API keys of your production Stripe account with your production provider organizations.

  The Stripe subscription plans, customers, payment methods, and subscriptions created using a test key are not visible when using a production key or when using the key to another account. If you change your Stripe keys after publishing a monetized product (for example, to a different account, or from test keys to production keys), the references that link APIC resources to the corresponding Stripe resources cannot be resolved. Subsequent operations might fail, resulting in inconsistent data between the APIC and Stripe systems.

Set up your environment

To define and apply payment options to your subscription Plan, complete the following steps:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. In your API Connect Designer interface, select Navigate to ☰) > Admin.
3. Select the Billing tab.
4. Select Add + > Stripe integration.
5. Select api keys to link to your Stripe account.
6. If you have a Stripe account, log into it on the Stripe website and skip to step 8.
7. If you do not have a Stripe account, create one now. You can complete this tutorial with a test account or a standard account.
    a. Select Don’t have an account? Sign up.
    b. Create an account with an email address and password. You need a test account for this procedure.
    c. Log in to your new Stripe account.
8. In the Stripe dashboard, select API in the navigation. You will need the two keys to copy into your API Connect interface to join the two accounts.
9. Copy your *Stripe Publishable key* from your Stripe dashboard, and paste it in the corresponding field on the Add Stripe Account window of API Connect.
10. Copy your *Stripe Secret Key* from the Stripe dashboard, and paste it in the corresponding field on the Add Stripe Account window of API Connect.

Note: Select the option to display the key if it is not already displayed.

11. Select Create to complete the Stripe integration. Your new integration should be displayed in the Billing Integrations table. A status of Connected means that the connection was verified.

Define the Plan

1. Create an API in the API Connect Designer if you don’t already have an existing one that you want to use.
   a. In the API Connect Designer interface select the Navigate to ☰) > Drafts .
   b. Select the APIs tab to view the existing APIs.
   c. Select Add + to create an API or to import one.
   d. Enter the required information for the API.
   e. Leave the Add a product box unselected.
2. Add the API to a Product.
   a. Select the Products tab to view the list of Products.
   b. Select Add + to create or import a Product.
   c. Complete the required information for the Product.
   d. Select Create product to create your Product.
   e. With the Design tab selected for your new Product, select APIs in the navigation.
   f. Select + beside the APIs entry to add your API to the Product.
   g. Select your API, and then Apply.
3. Create the Plan with pricing for your Product.
   a. With your Product open, select Plans in the navigation to define the monetization Plan for your Product. The Plans are the different subscription options that your customers choose from to use your APIs.
   b. Delete the Default Plan by selecting its trash icon.
   c. Select + to create your first Plan, which is titled New Plan 1 by default.
   d. Change the *Title* and *Name* to Basic.
   e. Select Monthly Subscription in the *Billing Model* field.
   f. Select USD in the Currency field.
   g. Set the price per month to 9.99.
   h. Set the Free Trial Days to 30.
   i. Set your rate limits to what you want for the Product on that Plan. For this example, set it to 7 calls per minute.
   j. Leave the rest of the information as it is.
   k. Select the Save icon (💾) to save the new Plan.

Publish your Product to catalog

1. Stage your Product to a catalog by selecting the stage icon and selecting the catalog to stage the product to.
2. Select the Navigate to... icon (☰) > Dashboard to view your catalogs.
3. Select the catalog where you staged your product.
4. Select the Settings tab, and ensure that the Development mode slider is off.
5. Select the Save icon (💾) to save your settings, if you changed them.
6. Select the Products tab. Your staged product should be in the list of Products. If not, you might have staged it to a different catalog or not staged it.
7. Select the icon of the three vertically aligned dots next to your product, and select Publish to publish your product.
8. Confirm your visibility settings and select Publish. The Public (Developer Portal) setting indicates that anyone with access to your Developer Portal can see this Product.

Verify that the subscription is created in your Stripe account

1. Log in to your Stripe account with your API Connect management server, if you are not already logged in.
2. Select Events and Logs in the navigation.
3. Verify that there is an event that was created when you published the Plan.

What you did in this tutorial

In this tutorial, you completed the following activities:

- Connected your API with a credit card processing service (in this tutorial, we used Stripe)
- Created a Plan with pricing for a Product
- Staged the Product to your Developer Portal so your customers can subscribe to it

Related information

- Tutorial: Subscribing to a plan with pricing

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 and earlier

API Manager tutorials for V5.0.6 and earlier

Tutorials for using the API Manager in IBM® API Connect Version 5.0.6 and earlier.

Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.7 and later, see API Manager tutorials for V5.0.7 and later.

- **Tutorial: Creating a SOAP API**
  This tutorial shows you how to create an API definition by using a SOAP service's Web Service Definition Language (WSDL) in IBM API Connect Version 5.0.6 and earlier. This API definition will allow simplified access to and management of access for 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, in IBM API Connect Version 5.0.6 and earlier.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 and earlier

Tutorial: Creating a SOAP API

This tutorial shows you how to create an API definition by using a SOAP service's Web Service Definition Language (WSDL) in IBM® API Connect Version 5.0.6 and earlier. This API definition will allow simplified access to and management of access for the SOAP service.

About this tutorial

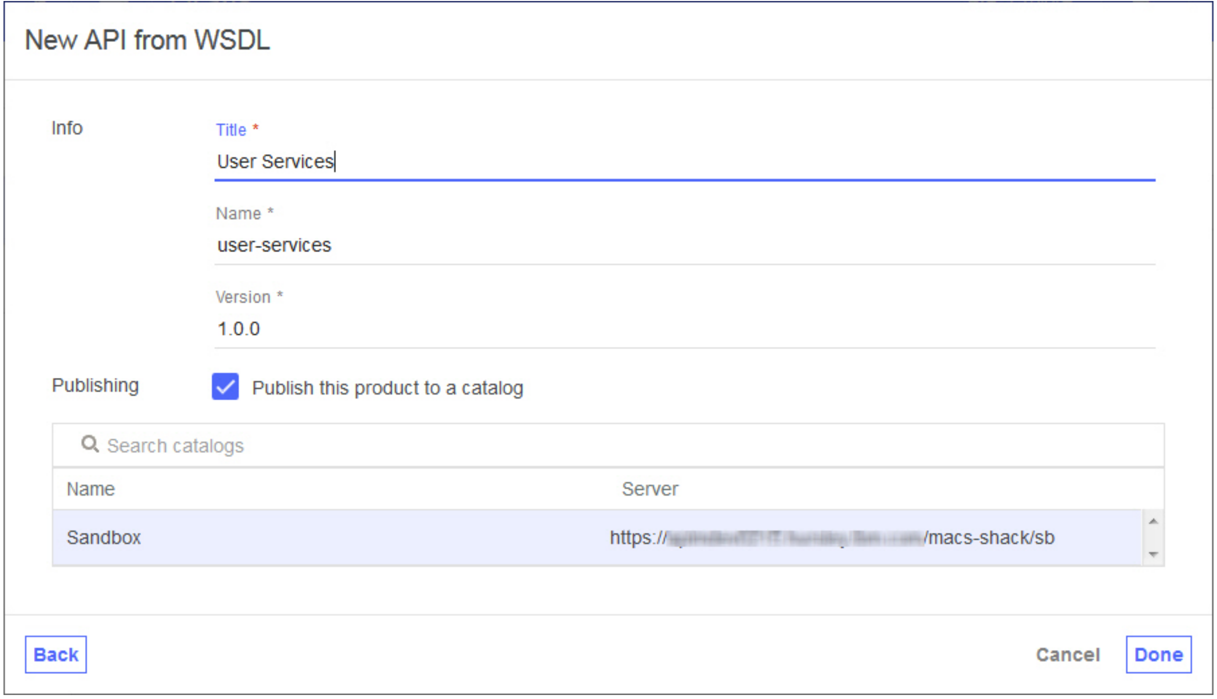
Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.7 and later, see API Manager tutorials for V5.0.7 and later.
In API Manager, you will create an API by importing the WSDL for an existing SOAP service. When called, the API takes a SOAP request from the API caller and 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.

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. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ≡. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
3. Click Drafts in the UI navigation pane and then click the APIs tab. The APIs tab opens.
4. V5.0.4 + Click Add > New OpenAPI from SOAP service.
   V5.0.3 and earlier Click Add > API from WSDL.
5. Click Upload file.
6. Select the `AccountService.wsdl` file from your file system
7. V5.0.3 and earlier Specify the service to include, and then create the API definition.
   a. Select the AccountService SOAP service then click Next.
   b. Select Create product and then name the Product `User Services`

    c. Click Done.
8. V5.0.4 + Specify the service to include, and then create the API definition.
    a. Select the AccountService SOAP service and then click Add a product.
    b. In the Title field, enter `User Services`.
    c. Leave the Name and Version fields unchanged.
    d. Ensure that the Publish this product to a catalog check box is selected and then select Sandbox as the target Catalog.



New API from WSDL

Info

Title *
User Services

Name *
user-services

Version *
1.0.0

Publishing ☑ Publish this product to a catalog

Search catalogs

Name	Server
Sandbox	https://[redacted]/macs-shack/sb

    e. Click Done. The Design tab for the draft of your API definition opens.

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

To add your SOAP API to a Product and Plan, and then test it, complete the following steps:

1. V5.0.3 and earlier In the APIs tab of the Drafts window, click your AccountService SOAP API.
2. Click the Assemble tab. The assemble view opens.
3. Click the Test icon ▶. The test tool opens.
4. If you have used the test tool before, click Change setup.
5. In the Catalog field, select your Sandbox Catalog.
6. In the Product field, select your User Services Product and then click Republish product to publish your Product so that it can be tested.
7. V5.0.3 and earlier In the Application field, select APIMGMT_TEST_APP and then click Subscribe.
8. Click Next.
   V5.0.4 + Notice that the Setup section indicates that automatic subscription is used: **`Sandbox, User Services 1.0.0, using automatic subscription`**. Automatic subscription is enabled by default for the Sandbox Catalog; so you do not have to specify a Plan or application when testing. A test application which is automatically subscribed to all the Plans in the Catalog will be used, with a pre-supplied client ID and client secret.
9. In the Operation field, select getBalance.
10. 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 "3". However, unlike the SOAP service, the API requires identification using a Client ID, which is enforced by the gateway server.
11. 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

Related information

- Adding a SOAP API definition
- Adding a SOAP API definition by using a WSDL file
- Adding a REST API definition

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 and earlier

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, in IBM® API Connect Version 5.0.6 and earlier.

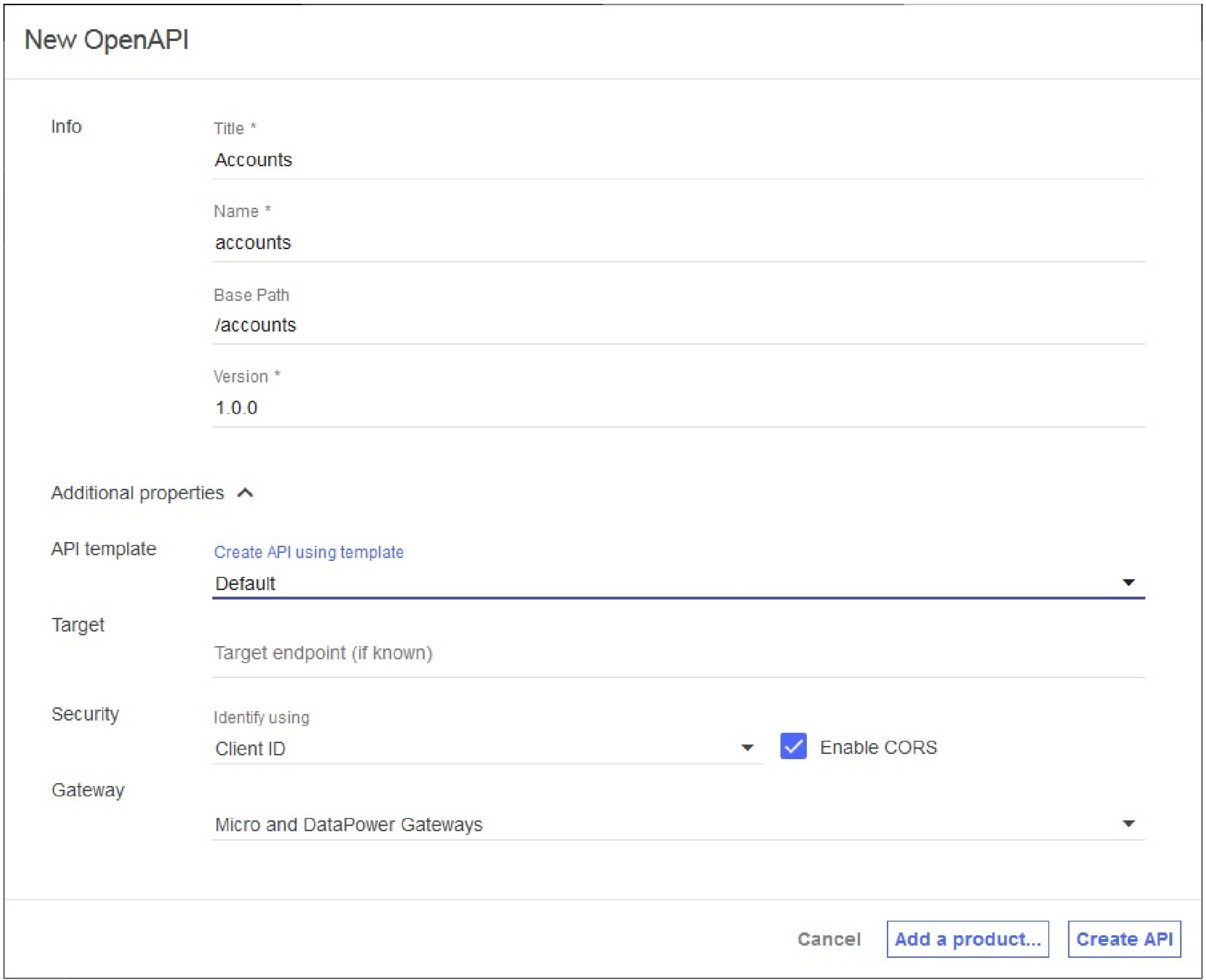
About this tutorial

Note: For tutorials about working with the API Manager in IBM API Connect Version 5.0.7 and later, see API Manager tutorials for V5.0.7 and later.
In API Manager, you will create a REST API that accesses a SOAP API to make data from the existing SOAP service available. This tutorial uses the same SOAP service as the Tutorial for creating a SOAP API tutorial, but exposes it in a different way.

Setting up a REST API definition

To set up a REST API, complete the following steps:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Click Drafts in the UI navigation pane and then click the APIs tab. The APIs tab opens.
3. V5.0.6 + Click Add > New API.
   V5.0.4 + Click Add > New OpenAPI from scratch.
   V5.0.3 and earlier Click Add > API.
4. Specify basic information about the API.
   a. In the Title field, enter `Accounts`.
   b. Leave the Name field as `accounts` when it is filled while you enter your title.
   c. V5.0.4 + Leave the Base Path field as `/accounts`.
   d. Leave the Version field as `1.0.0`.
5. V5.0.4 + Expand Additional properties to specify additional properties for the API.
   a. From the API template field, select Default to indicate that you want to use the default template to create the API definition.
   b. Leave the remaining fields unchanged.



Title * Accounts

Name * accounts

Base Path /accounts

Version * 1.0.0

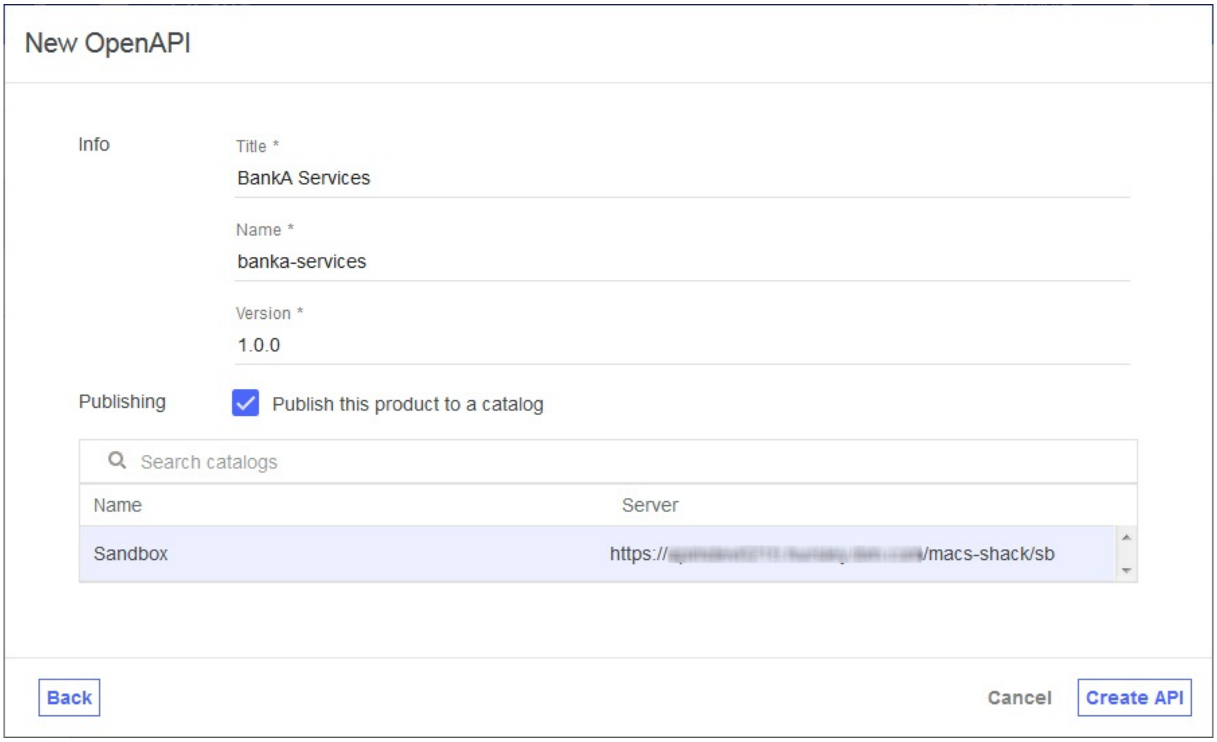
API template: Default

Target: Target endpoint (if known)

Security: Identify using Client ID ☑ Enable CORS

Gateway: Micro and DataPower Gateways

6. V5.0.3 and earlier Add your API to a new Product and then create the API definition.
   a. Select Add to a new Product and, in the Title field, enter `BankA Services`.
   b. Click Add.
7. V5.0.4 + Add your API to a new Product and then create the API definition.
   a. Click Add a product.
   b. In the Title field, enter `BankA Services`.
   c. Leave the Name and Version fields unchanged.
   d. Ensure that the Publish this product to a catalog check box is selected and then select Sandbox as the target Catalog.



New OpenAPI

Info

Title *
BankA Services

Name *
banka-services

Version *
1.0.0

Publishing ☑ Publish this product to a catalog

Name	Server
Sandbox	https://[redacted]/macs-shack/sb

e. Click Create API. The Design tab for the draft of your API definition opens.

8. V5.0.3 and earlier In the Base Path field, enter `/accounts`.
9. In the Definitions section, click the Add Definition icon ⊕ and then expand the new definition by clicking it.
10. Name the definition `Account Output`.
11. The definition contains a single property. Rename the property to `Balance` and, in the Type column, select double. Mark the property as required in the * column by selecting the check box. Marking a property as required is reflected to customers in the OpenAPI (Swagger 2.0) definition of the API and can be enforced by a validate policy.
12. In the Paths section, click the Add Path icon ⊕.
13. In the Path field of your newly created Path, replace the contents with `/balance`.
14. Expand the GET /balance operation by clicking it.
15. For your GET /balance operation, click Add Parameter and then click Add new parameter.
16. Name your new parameter `customer_id` and mark it as required by selecting the check box. Marking a parameter as required is indicated to users in the OpenAPI (Swagger 2.0) definition of the API, is enforced by validate policies, and will result in the test tool always generating the parameter as part of a sample API call.
17. In the Schema column of the 200 OK response in the Responses section, select your Account Output definition.
18. Click the Save icon 💾 to save your changes.

Adding and configuring your web service invocation

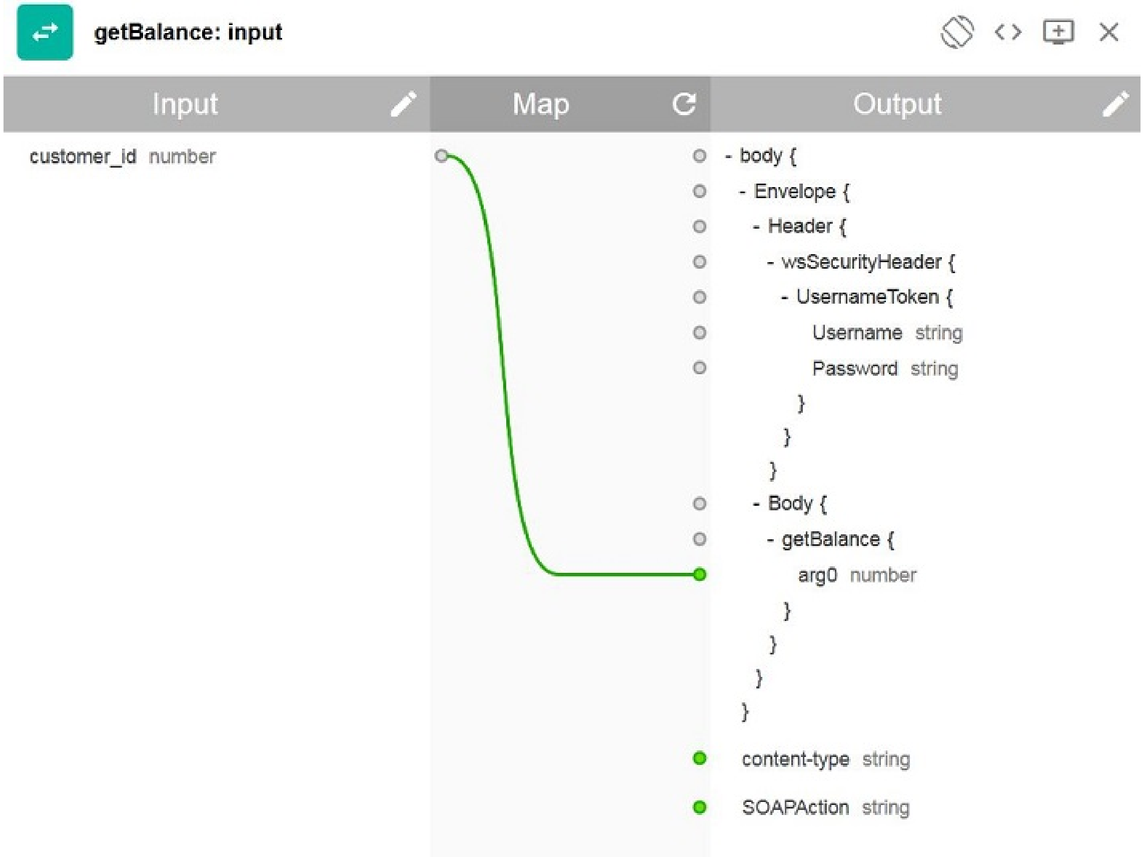
To add and configure the invoke and map policies that integrate your web service into your API definition, complete the following steps:

1. Download the SOAP WSDL file AccountService.txt. Rename this file `AccountService.wsdl`.
2. In the Services section, click the Add service icon ⊕. The "Import web service from WSDL" window opens.
3. Click Upload file.
4. Select the `AccountService.wsdl` file from your file system
5. Select the AccountService SOAP service and then click Done. In the Services section, a web service is listed.
6. Click the Assemble tab and then ensure that DataPower Gateway policies is selected.
7. Delete the existing invoke policy on the canvas by hovering your cursor over the policy and then clicking the Delete policy icon 🗑.
8. From the palette, drag the getBalance web service onto the dashed box that is displayed on the canvas. An invoke policy and two map policies are placed in the assembly. The first map policy assigns variables to the input of your web service invocation, while the second policy assigns outputs of your web service invocation to variables. The outputs of the first map and the inputs of the second map are generated from the WSDL that you uploaded.
9. Click the getBalance: input map policy and then click the Edit inputs icon ✎ in the Input column of the property sheet.
10. Click + input.
11. Configure the input according to the following table:

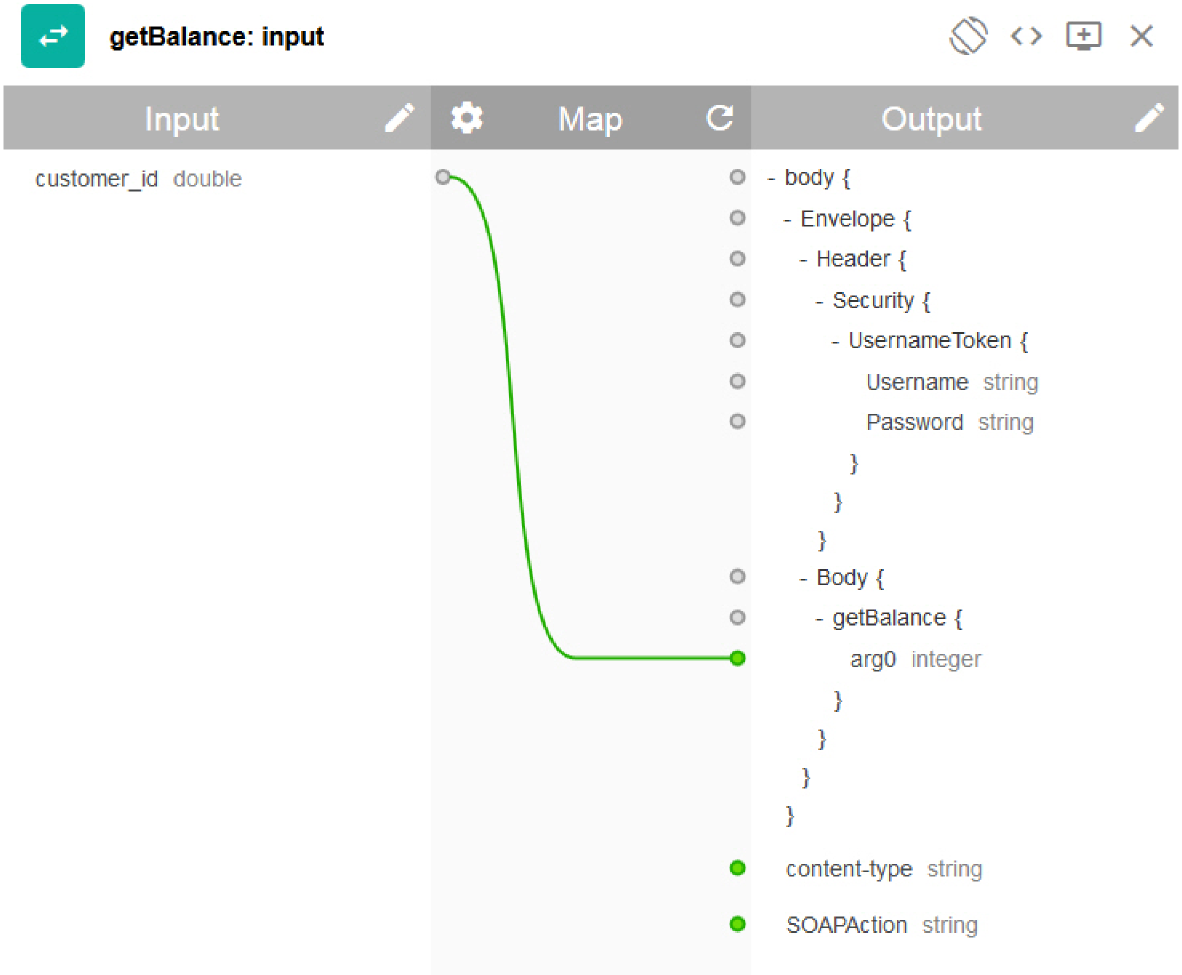
Table 1. Web service input

Property	Value
Context variable	`request.parameters.customer_id`
Name	`customer_id`
Content type	`none`
Definition	`double`

12. Click Done.
13. V5.0.3 and earlier Click the circle corresponding to customer_id string on the input side and then click the circle corresponding to arg0 number on the output side.



14. V5.0.4 + Click the circle corresponding to customer_id double on the input side and then click the circle corresponding to arg0 integer on the output side.



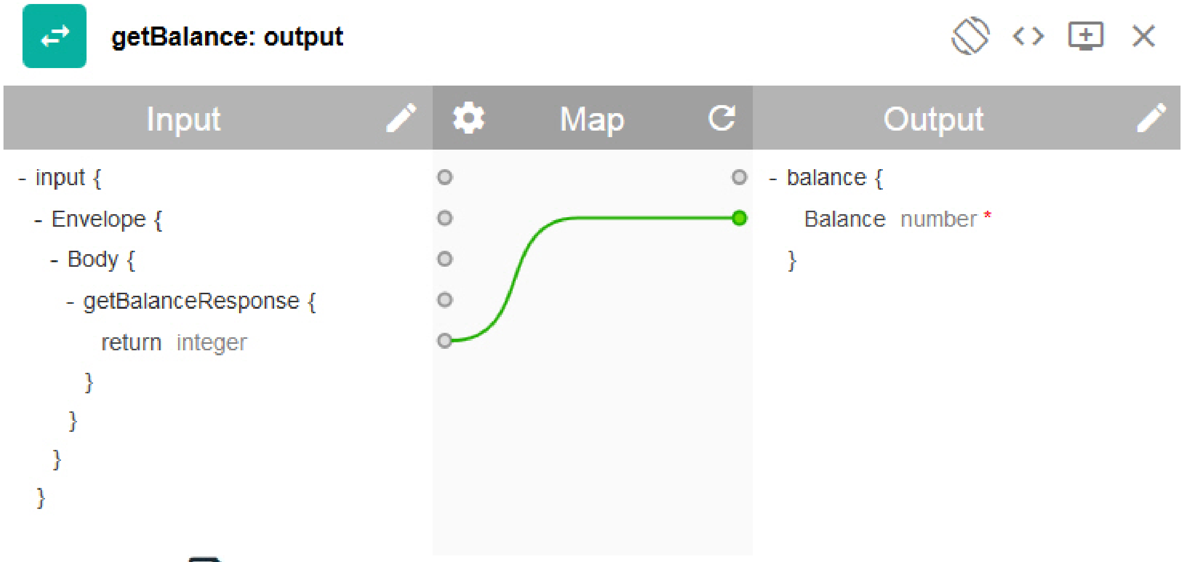
15. Close the property sheet.
16. Click the getBalance: output map policy in the palette and then click the Edit outputs icon in the Output column of the property sheet.
17. Click + output.
18. Configure the input according to the following table:

Table 2. Web service output

Property	Value
Context variable	`message.body`
Name	`balance`
Content type	`none`
Definition	`#/definitions/Account Output`

19. Click Done.
20. V5.0.3 and earlier Click the circle corresponding to return number on the input side and then click the circle corresponding to balance number on the output side.

    The output map policy
21. V5.0.4 + Click the circle corresponding to return integer on the input side and then click the circle corresponding to Balance number on the output side.



22. Click the Save icon to save your changes.

You have included the web service invocation in your assembly and mapped an input parameter to the appropriate part of the SOAP request and mapped the appropriate part of the SOAP response to a JSON output.

Testing your API definition

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. If you have used the test tool before, click Change setup.
3. In the Catalog field, select your Sandbox Catalog.
4. In the Product field, select your BankA Services Product and then click Republish product to publish your Product so that it can be tested.
5. V5.0.3 and earlier In the Application field, select APIMGMT_TEST_APP and then click Subscribe.
6. Click Next.
7. In the Operation field, select get /balance.
8. In the customer_id field, enter `12345`.
9. 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:

- 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

- Adding a SOAP API definition
- Adding a SOAP API definition by using a WSDL file
- Adding a REST API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 Designer user interface (UI). The API Designer 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, or Micro 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.

See the following topics for more information about user-defined policies, and how to create and import them into IBM API Connect:

- **User-defined policies in IBM API Connect**
  Create user-defined policies 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.
- **Describing your policy**
  Describe your user-defined policy by creating a definition file in YAML format to document metadata about the policy.
- **Authoring policies for the DataPower Gateway**
- **Authoring policies for the Micro Gateway**
- **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.

Related information

- IBM API Connect Overview
- API policies

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

User-defined policies in IBM API Connect

Create user-defined policies 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.

When would you need a user-defined policy?

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.

How do user-defined policies work?

A user-defined policy is implemented in one of the following ways depending on the platform:

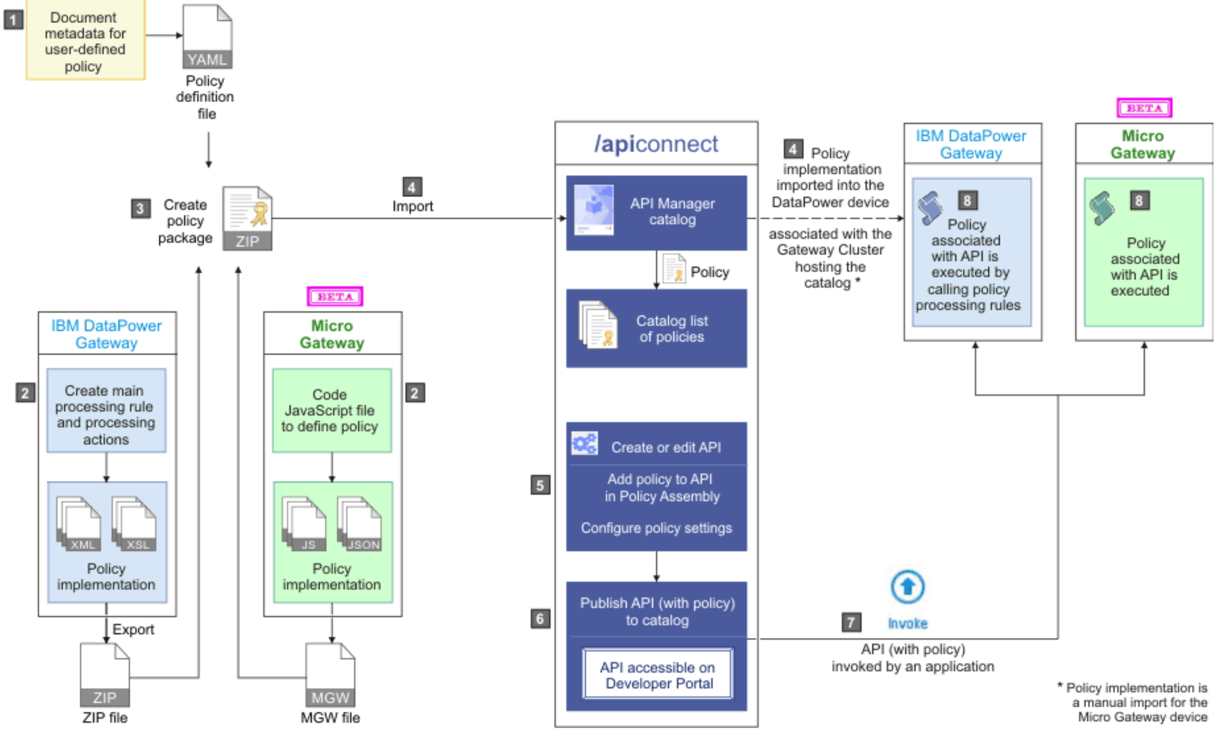
- For an IBM DataPower Gateway, as a DataPower processing rule.
- BETA For a Micro Gateway, as a Node.js module.

Note: User-defined policies for the Micro Gateway are marked as BETA to indicate that the implementation might change in a future release. IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
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 depending on the platform:

- An API Gateway that is running in DataPower calls the IBM DataPower Gateway processing rules that those policies implement.
- BETA A Micro Gateway runs the Node.js module that implements the policy.

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


1
Document metadata for user-defined policy
YAML
Policy definition file
3
Create policy package
ZIP
4
Import
/apiconnect
API Manager catalog
Policy
Catalog list of policies
4
Policy implementation imported into the DataPower device
associated with the Gateway Cluster hosting the catalog *
IBM DataPower Gateway
8
Policy associated with API is executed by calling policy processing rules
BETA
Micro Gateway
8
Policy associated with API is executed
IBM DataPower Gateway
2
Create main processing rule and processing actions
XML
XSL
Policy implementation
Export
ZIP
ZIP file
BETA
Micro Gateway
Code JavaScript file to define policy
2
JS
JSON
Policy implementation
MGW
MGW file
5
Create or edit API
Add policy to API in Policy Assembly
Configure policy settings
6
Publish API (with policy) to catalog
API accessible on Developer Portal
7
Invoke
API (with policy) invoked by an application
* Policy implementation is a manual import for the Micro Gateway device

Related information

- Getting started with the API Manager
- User-defined policies

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Micro Gateway only Note: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

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 (Swagger 2.0) names of the built-in policies, otherwise it will cause a policy conflict in the API assembly definition. For a list of the OpenAPI (Swagger 2.0) built-in policy names, see execute.
     Micro Gateway only 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*.
     Micro Gateway only 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. DataPower Gateway only 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.
   Micro Gateway only Note: The **`attach:`** property has no effect in the Micro Gateway.
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 which gateway that your policy is for.
   - DataPower Gateway only Enter the following code for a DataPower Gateway:

     ```
     gateways:
       - datapower-gateway
     ```

   - Micro Gateway only Enter the following code for a Micro Gateway:

     ```
     gateways:
       - micro-gateway
     ```

   - You can also target a policy for both gateways:

     ```
     gateways:
       - datapower-gateway
       - micro-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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Authoring policies for the DataPower® Gateway

- **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.
- **Reference**
  Reference information for authoring policies in IBM API Connect.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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
- V5.0.8 + 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 -->
  <xsl:import 
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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 left 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_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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, 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.

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_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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, 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 -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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:include
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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:include
```

```
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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.

V5.0.8 +

(From API Connect version 5.0.8.8 onward) 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"
  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_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_apim.custom.xsl" />

  <xsl:template match="/">

    <xsl:variable name="exception" select="apim:getError()"/>
    <!-- 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: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()` function returns an XML node set; for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<error>
    <name>RuntimeError</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_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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/apimanagement" extension-element-prefixes="dp func apim">

  <!-- Contains the APIM functions -->
  <xsl:import
href="#_home_markdown_jenkins_workspace_Transform_in_SSMNED_5.0.0_com.ibm.apic.policy.doc_local:_isp_pol
icy_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.

Related concepts

- Authoring policies

Related tasks

- Implementing your policy

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

BETA

Authoring policies for the Micro Gateway

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

- **Creating a user-defined policy for the Micro Gateway**
  Create a user-defined policy as a Node.js module that includes a package description, a policy definition file, and JavaScript, and add the module to the API Connect user policy array.
- **Accessing policy properties**
  Policy properties values that are defined in the policy definition are passed to the policy implementation at run time.
- **Reference**
  Reference information for accessing the context object when authoring policies in JavaScript.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

BETA

Creating a user-defined policy for the Micro Gateway

Create a user-defined policy as a Node.js module that includes a package description, a policy definition file, and JavaScript, and add the module to the API Connect user policy array.

Important: IBM® API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
The Micro Gateway handles user-defined policies that are packaged as Node.js modules. The module contains the following files:

- package.json
- policy.yaml
- index.js, or other JavaScript file
- README.md, optional

The following directory tree represents a basic user-defined policy. This structure is identical to a typical Node.js module except for the addition of the policy.yaml. This description does not preclude other types of files from being added.

```
<package root directory>
   ¦  package.json
   ¦  policy.yaml
   ¦  README.md
   ¦
   +-- lib
   ¦     index.js
   ¦
   +-- node_modules
   ¦     +-- <dependent modules>
   ¦
   +-- test
         +-- <test files>
```

Note: When you write a user-defined policy, the dependencies in the package.json might be modified by npm when you use the `--save` option or when manually added. Although the dependencies of package.json support many formats to specify the version of a dependency, specify the exact version that you are using when you develop the policy. Other annotations, such as `>=version`, `>version`, `^version`, or `~version`, might cause some compatible issues.
Review the following topics to learn details about configuring a user-defined policy for the Micro Gateway.

- **Coding the JavaScript file**
  Code the JavaScript file for a user-defined policy with the required parameters.
- **Deploying user-defined policies in the developer toolkit environment**
  In the IBM API Connect developer toolkit environment, deploy your user-defined policy to the Micro Gateway and configure the user policies array object.

Related information

- IBM API Connect Overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Coding the JavaScript file

Code the JavaScript file for a user-defined policy with the required parameters.

The main JavaScript file of a Node.js module is defined by the package.json file. This same construct applies for user-defined policies. For the purposes of this example, assume that the JavaScript file is index.js and that it is in the lib/ directory.

The index.js file shows the exported function that is the factory for the policy run-time implementation. It is called once during the policy setup phase each time the policy is enforced. The function must call `flow.proceed()` or `flow.fail(error)` for the error case.

The source of the index.js file looks like this:

```
/**
 *
 * @param config - Static configuration information that may apply to this policy
 */
module.exports = function (config) {
    ...
    // do global setup, initialization, and config parsing here.
    ...

    /**
     * This is the policy runtime implementation. It is called each time the
     * policy is enforced.
     * @param props - property values as defined in policy.yaml
     * @param context - context values
     * @param flow - a flow instance and includes all of the flow APIs
     */
    return function (props, context, flow) {
      ...
      //do "property parse" and "policy enforce"
      ...

      flow.proceed(); // you must call flow.proceed() or flow.fail(error) for the error case
    }
}
```

where:

- The `config` parameter of the factory function initially is an empty JSON object. This value is populated with values from `policy-providers.json`.
- The values in the `props` parameter are defined in the policy definition file, policy.yaml, and set in the flow assembly.
- The `context` values support the API Gateway context variables.
- The `flow` parameter is a flow instance and contains a set of APIs that a policy can use to interact with the flow-engine. The APIs includes:
    - `.proceed()`: A policy must call this API when it finishes its job successfully. Then, the next policy is then invoked.
    - `.fail(error)`: A policy fails, and an error is thrown. Then, the flow-engine handles the error.
    - `.stop()`: A policy finishes all its logic, and a flow ends. This function ends the whole flow, and the transaction finishes.
    - `.subscribe(event, cb)`: A policy can subscribe to a specific event that flow-engine supports. The callback function is invoked when the events are triggered. The interface of `cb` is : `function(event, next)`. The callback function must call `next()` when it is done. The following events are supported:
        - `FINISH`: A transaction finished.
        - `ERROR`: A transaction failed with an error.
        - `pre:policyName`: Before a specific policy is executed.
        - `post:policyName`: After a specific policy is executed.
    - `.unsubscribe(event, cb)`: Unsubscribe an event that a policy subscribes before.
    - `.logger`: A Bunyan logger object that can be used for logging.

Related information

- API Gateway context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

BETA

Deploying user-defined policies in the developer toolkit environment

In the IBM® API Connect developer toolkit environment, deploy your user-defined policy to the Micro Gateway and configure the user policies array object.

Before you begin

Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower® Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.
You must create a policy YAML file before you deploy your user-defined policy. For more information, see Describing your policy.

You must code the JavaScript file before you deploy your user-defined policy. For more information, see Coding the JavaScript file.

About this task

Deploy your user-defined policy to the Micro Gateway by copying the package to a directory that is accessible to Node.js. Then, populate the user policies array object with the absolute path to the package. When you start the API Designer assembly editor, the policy is available in the policy palette.

The following conditions apply:

- When you edit the file, it might be null except for empty braces, `{}`.
- User-defined policies can be stored in any directory hierarchy on the file system that is accessible to Node.js.
- Only one policy is permitted per leaf directory, for example, ~/mypolicies/policya and ~/mypolicies/policyb each contain a single policy.
- The `userPolicies: []` array object contains a list of absolute paths to the policies. The values can point to a parent location, for example, `userPolicies: ["~/mypolicies"]`, and the Micro Gateway recursively searches the directory for leaf directories. Alternatively, each policy location can be specified in the array, for example `userPolicies: ["~/mypolicies/policya", ~/mypolicies/policyb"]`.
  Note: 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.
- If you specify the full path to the policy file, such as `~/foo/policya/policy.yml`, the policy is ignored, and a warning is logged.
- File paths that include wildcards, such as `~/foo/*`, are ignored with a warning.
- If two policy files are found in the same directory, such as `~/foo/policy.yml` and `~/foo/policy.yaml`, the Micro Gateway uses the first one encountered, and the rest are ignored with a warning.

Procedure

1. Create a policy directory that is accessible to Node.js.
2. Copy the user-defined policy package to the policy directory.
3. Add the `userPolicies: [ ]` array object to the existing JSON object in ~/<project>/.apiconnect or ~/.apiconnect/config.
4. Edit the `userPolicies: [ ]` array object, and enter the absolute path to the policy or policy directory.
5. Start the UI. The user-defined policy is available on the policy palette, and you can add the policy to your API.

Results

You implemented for your user-defined policy on the Micro Gateway. When you test the user-defined policy, the Micro Gateway automatically loads and enforces it.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

BETA

Accessing policy properties

Policy properties values that are defined in the policy definition are passed to the policy implementation at run time.

Policy properties are defined in the policy definition. The values for these properties are set to default values or overridden in the API Manager with the UI or other means. The values are passed to the policy implementation at run time so that each invocation of a given policy implementation uses different policy property values.

For user-defined policies, the values for policy properties are passed in as a JSON object in the **`props`** variable in the index.js file. For example, assume the following policy.yml excerpt from an encryption policy:

```
enc:
  label: "Encryption Algorithm"
  description: "Select an encryption algorithm"
  type: string
enc-alg:
  label: "Key Encryption Algorithm"
  description: "Select a key encryption algorithm"
  type: string
```

The property values can be accessed by the policy implementation at run time by interrogating the **`props`** var:

```
var myEnc     = props.enc;
var myEncAlg  = props['enc-alg'];
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reference

Reference information for accessing the context object when authoring policies in JavaScript.

- **Reference information for accessing the context object**
  Example code snippets access the context object in user-defined policies.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reference information for accessing the context object

Example code snippets access the context object in user-defined policies.

You can reference context variables in user-defined policies with the **`getContext()`** and **`setContext()`** functions and with JavaScript dot notation.

The **`context`** object is like a whiteboard for storing transactional information. Every policy in the same assembly gets the same **`context`** object for a single transaction. All of the policies can populate information into the context object and retrieve information from it. Data is categorized based on JavaScript properties. For example, all of the request related data are stored under **`request`** and **`response`** related data go to **`message`**. Pre-populated data might be read-only properties. Code that attempts to override read-only values causes an exception.

Reading payload

Request and response payload are stored in a context object. Request payload is populated into the **`request.body`** and response payload is stored in the **`message.body`**. The data type of the **`request.body`** and the **`message.body`** is determined by the value of

request.content-type.

Here are examples that access the request payload.

Use dot notation to access the **request.body**:

```
var requestPayload  = context. request. body;
```

Use the **get** function to get the **request.body**:

```
var payload  = context. get(' request.body ');
```

Writing payload

The **message.body** starts with same value as the **request.body**, but the value can changed as the flow continues. After the policy implementation runs, the **message.body** becomes the payload for subsequent policies, if any, in the flow. The following table shows the correlation between the **message.body** data type and the **request.content-type** data type:

Table 1. Request body and message body data types

request.content-type	message.body
*/json	JSON
*/+json	JSON
text/*	String
*/x-www-form-urlencoded	JSON (key or value represents the HTTP form data)
*/xml	XML
*/+xml	XML
All others	Buffer

Reading headers

Request and response headers are stored in the **context** object. You can retrieve the headers as in the following examples:

```
//get header
var requestHdr = context.get('request.headers.req-header-name');
var responseHdr = context.get('message.headers.resp-header-name');
```

Writing headers

Request headers are read-only headers, therefore, you cannot change them. However, you can modify headers in **message.headers**. The **message.headers** are the headers that are sent back to the client.

```
//set header
context.set('message.headers.x-custom-header', 'value');
context.message.headers.myheader ='myheader';
```

If the header name contains a hyphen, use the **get()** function to retrieve it. Use brackets, **[]**, to access this type of variable with dot-notation. For example:

```
var contentType = context.message.headers['content-type'];
//or
contentType = context.get('message.content-type');
```

Reading context

Context is transactional-wide shared variable space. For Micro Gateway policies, a context object provides access to context variables. The **get()** function and dot-notation are used to retrieve context values.

For example:

```
var verb = context.get('request.verb');
var uri = context.get('request.uri');
var path = context.request.path;
```

where `context` object is passed to the policy handler function by default.

Writing context

Similar to the `get()` function that reads context variables, the `set()` method writes context variables. For example:

```
context.set('acme.itemnumber', 'item1');
```

JavaScript dot-notation also is supported, as in this example:

```
context.acme.itemnumber2 = 'item2';
```

/com.ibm.apic.toolkit.doc/rapim_context_var.html

Related information

- API Gateway context variables

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

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. DataPower Gateway only Implement your policy by using DataPower processing rules and actions.
3. Micro Gateway only Code the JavaScript file for your policy.
Important: IBM API Connect Micro Gateway is deprecated in IBM API Connect Version 5.0.8 in favor of DataPower Gateway. From 1 April 2020, Micro Gateway, and associated toolkit CLI commands, will no longer be supported. Existing users can migrate their API definitions to IBM DataPower Gateways. For information on supported API policies, see Built-in policies.

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
              mypolicy_mgw.mgw
```

where

- *<policy-name>*.yaml is your policy definition file. The YAML file must use the name that is specified in its `name` property.
- DataPower Gateway only 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.
- Micro Gateway only implementation/*mypolicy_mgw*.mgw is your policy implementation for a policy that is deployed to the Micro Gateway. The policy implementation file contains all of the source files (package.json, .js files, README.md, and so on). The package.json file and its peer files and folders are at the root path of the compressed file.

The name of the .zip file or .mgw 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 or a .mgw policy implementation file, but can contain both.

2. Use the API Manager user interface to import your policy into a Catalog. When successfully imported, the policy appears on the policy palette of the API Designer assembly editor. 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.
3. Micro Gateway only Deploy the imported policies to the Micro Gateway collective members.
   a. Create a directory on the Micro Gateway named package/userPolicies.
   b. Under package/userPolicies, create a directory for each policy to be deployed, such as package/userPolicies/micro-gw-policy-sample.
   c. Expand the contents of the compressed policy file into the policy directory. For example, the compressed file micro-gw-policy-sample.mgw is expanded into the package/userPolicies/micro-gw-policy-sample directory.
   d. Under the package/ directory, edit or create a file named .apiconnect.
   e. Edit or add the `{ userPolicies: [<userPolicies>] }` array object, and enter the absolute path to the policy directory.

What to do next

Micro Gateway only 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5 Developer Portal: discover and use APIs

Application developers discover and use APIs by using the V5 Developer Portal. You can customize the Developer Portal for your application developers.

In addition to allowing application developers to find and use both free and paid APIs, the Developer Portal provides additional features including forums, blogs, comments, and ratings, together with an administrative interface for customizing the Developer Portal. For tutorials on using the Developer Portal, see Tutorials.

For information about deploying, using, and configuring the Developer Portal, as well as best practices for administrators, see the following sections:

- **Use the Developer Portal**
  Users of the API Manager UI can publish Products and APIs to the Developer Portal for Application Developers to access and use. The tasks and concepts that are described in this section are for authorized users.
- **Configure the Developer Portal site**
  Depending on the permissions that the role of a Developer Portal user possesses, you can configure features of the Developer Portal.
- **Developer Portal best practices for administrators**
  You are recommended to follow the best practices that are listed in the next sections, to ensure you get the best experience from the Developer Portal appliance.
- V5.0.8 + **Migration support for your Developer Portal**
  From IBM® API Connect Version 5.0.8.6 onwards, migration support is provided for exporting certain Developer Portal users and content.
- **Troubleshooting the Developer Portal**
  This page describes how to troubleshoot common problems that can occur when using the Developer Portal. The steps outlined apply to IBM API Connect version 5.0.8.4 and later; some of the troubleshooting commands might be missing from earlier releases. It is recommended that you regularly upgrade to the latest fix pack or interim fix.
- **Developer Portal tutorials**
  Tutorials for using the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Use the Developer Portal

Users of the API Manager UI can publish Products and APIs to the Developer Portal for Application Developers to access and use. The tasks and concepts that are described in this section are for authorized users.

Application developers can learn more about an organization's APIs and register applications so that they can use APIs.

Use the following tasks to work with APIs and Plans.

- **The Developer Portal user interface**
  Use the menu items in the Developer Portal user interface to register applications, work with APIs, interact with other application developers, and obtain support.
- **APIs in the Developer Portal**
  You can view, edit, and configure APIs in the Developer Portal.
- **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).
- **Developer organizations in the Developer Portal**
  You can add, remove, and configure Developer organizations in the Developer Portal.
- **User accounts, passwords, and support in the Developer Portal**
  You can change user properties such as `timezone` and `passwords` in the Developer Portal, for your own user.
- **Configure bookmarks in the Developer Portal**
  You can configure, add, and remove bookmarks in the Developer Portal.

Related information

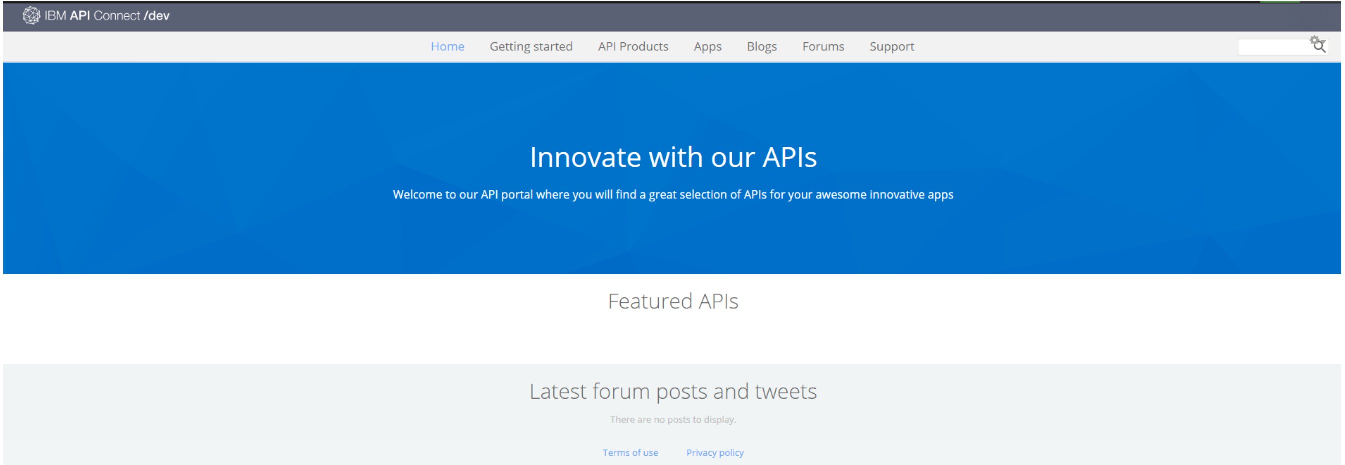
- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

The Developer Portal user interface

Use the menu items in the Developer Portal user interface to register applications, work with APIs, interact with other application developers, and obtain support.

The following sections detail the items in the main menu of the Developer Portal user interface, and summarize the activities that you can complete on each of the associated pages.



Innovate with our APIs

Welcome to our API portal where you will find a great selection of APIs for your awesome innovative apps

There are no posts to display.

Note: If your Developer Portal has been customized, the options available might differ from those described here.

Home

The Home page summarizes new and active forum topics, and recent comments, and links directly to the APIs page.

Getting Started

The Getting Started page describes the tasks that you complete to begin working with APIs in the Developer Portal.

API Products

The API Products page shows all the APIs that are available for you to work with. Click the name of an API to see the details of that API, and to test it.

Apps

From the Apps page, you register new applications so that you can use the APIs that are available to you. The Apps page shows all the applications that you have registered. Click the name of an application to see the details of that application, and to modify it.

Blogs

Use the Blogs page to view blog entries, and associated comments, and to add your own comments.

Forums

The Forums page lists all of the available forums. Click the name of a forum to read the posts in that forum, and to add your own posts.

Support

From the Support page, you can search and participate in the developer forum, view frequently asked questions, raise a support ticket, and submit a question or feedback by email.

APIs in the Developer Portal

You can view, edit, and configure APIs in the Developer Portal.

To learn about APIs in the Developer Portal, use the following topic links:

- **Browsing available APIs, code snippets, and example operations**
  You can browse available APIs and their code snippets through the Developer Portal.
- **Selecting a Plan**
  A Plan is a collection of REST API operations and SOAP API WSDL operations. In the Developer Portal you can browse and select the most appropriate Plan to use with your application.
- **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.
- **Signing up to use an API**
  You can sign up to use an API in the Developer Portal.
- **Testing an API 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 Developer Portal and click Invoke to see the response.
- **Downloading a WSDL file for a SOAP API**
  You can download a WSDL file so that you can use it to generate a client stub to invoke the SOAP API. For example, by using **wsdl2java** or a similar approach.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Browsing available APIs, code snippets, and example operations

You can browse available APIs and their code snippets through the Developer Portal.

About this task

Code snippets are sample code fragments that are generated automatically from REST APIs. They demonstrate how an API consumer can invoke and use an API operation.

V5.0.3 + If you have administrator permissions, you can specify the default programming language that the code snippets are displayed in, and enable code snippets for SOAP APIs. For more information, see Enabling code languages for code snippets.

Examples of code for operations can be displayed in the Developer Portal. For more information on how example code is generated, see Example code for operations.

Procedure

1. From the Developer Portal home screen, click API Products.
2. To see more details, click APIs for the Product that contains the required API, then click the API.
   You can see the following information:
   - Any additional documentation available for you to read.
   - If you need to provide a Client ID or Client Secret to access this API.
   - The authentication requirements that are required to access the API.
   - The operations or WSDL operations that belong to the API.
   - V5.0.8 + The Plans that the operations belong to, including any pricing details.
3. Optional: More details of the individual operations associated with an API are available under the API Operations heading.
   If the API is secured with OAuth, the following parameters are displayed:

- The authorization endpoint URL for the operation owner. For more information, see *Section 3.1 Authorization Endpoint* in The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749).
- The token endpoint URL. For more information, see *Section 3.2 Token Endpoint* The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749).
- The client type. For more information, see *Section 2.1 Client Types* in The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749).
- The grant types. For more information, see *Section 1.3. Authorization Grant* in The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749).
- The default scope. For more information, see *Section 3.3 Access Token Scope* in The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749).

4. Optional: To view the example code of an operation:
   a. In the API Operations section, click Paths, and click the operation that you want to see the example code for.
   b. Click the operation's tab to expand and show its details, including code examples.
   You can view the example code for your operation in several languages:
      - cURL
      - Ruby
      - Python
      - PHP
      - Java
      - Node
      - Go
      - Swift
      - C
      - C#

   Note: The languages that are available for the code example might vary as they are dependent on the languages that the administrator has enabled. For more information on how to enable and display languages, see Enabling code languages for code snippets.

- **Request and response examples**
  Examples for request and response operations can be displayed in the Developer Portal.
- V5.0.4 + **Selecting the default code snippet language**
  Any user of the Developer Portal can select the default programming language that their code snippets are displayed in.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Request and response examples

Examples for request and response operations can be displayed in the Developer Portal.

Note: Request and response examples are different to code snippets for APIs. For more information on code snippets for APIs, see Browsing available APIs, code snippets, and example operations.
The examples can be created from response objects, which are defined in the OpenAPI (Swagger 2.0) documentation. By using the OpenAPI (Swagger 2.0) documentation, an object is created. The object that is created is used to create a schema.

If examples are defined in the OpenAPI (Swagger 2.0) documentation, they are used in the schema. However, if examples are not defined in the OpenAPI (Swagger 2.0) documentation, they are generated based on the contents of the field type or names of the properties.

Examples can be either based on the schema, default properties, or example properties (in that order).

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.4 +

Selecting the default code snippet language

Any user of the Developer Portal can select the default programming language that their code snippets are displayed in.

Procedure

1. After you have logged in to the Developer Portal, click on your username.
2. In the Account settings window, click the Edit tab.
3. From the Code Snippet Language drop-down list in the Edit tab, select your default code snippet language.
4. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Selecting a Plan

A Plan is a collection of REST API operations and SOAP API WSDL operations. In the Developer Portal you can browse and select the most appropriate Plan to use with your application.

Before you begin

You must register an application to use with a Plan. For more information about registering an application, see Registering an application.
V5.0.8 + Plans that you can subscribe to might be free plans or paid plans. If you subscribe to a paid plan, you must have an account with a supported credit card processor, such as Stripe. Your account administrator can create the account. See Tutorial: Defining a subscription Plan with pricing for your API Product for more information about setting up an account with a credit card system provider.

About this task

A Plan is a collection of API operations or subsets of operations from one or more API. A Plan can contain a mixture of HTTP GET, PUT, POST, OPTIONS, HEAD, PATCH, and DELETE verbs from different APIs or it can contain all the GET verbs from various APIs. A Plan can have a shared rate limit for all the operations, or each operation can have a different rate limit. Rate limits specify how many requests an application is allowed to make during a specified time interval.
V5.0.2 + In addition, Plans can also have multiple rate limits set per Plan and per operation, at second, minute, hour, day, and week time intervals.

DataPower Gateway only 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.

Use the Developer Portal to browse the different Plans that are available to you and select a Plan that is most suitable for your requirements. Some Plans have restricted access that you must request access to use. When you submit your request, the organization is notified, the API administrator assesses your request and they might contact you for more details. Other Plans are available to use straight away.

Procedure

To select a Plan, complete the following steps:

1. In the Developer Portal, click API Products.
2. Click the Product that contains the Plan that you want to work with.
3. In the Plan section of the Product, click the Plan on the left that you want to use.
   The details of the selected Plan are displayed.
4. After you have identified the plan that you want to use, click Use this plan.
   The Use this Plan dialog box is displayed.
5. Select the application that you want to use with this Plan, and click Save.
   The application details are displayed.
6. Optional: To view the operations for the APIs that are included in the Plans to which the application is subscribed, click the name of the API.
7. If the Plan is not restricted, you can use it immediately. If the Plan is restricted, the Plan is shown as Pending Approval, and you cannot use the requested Plan until the administrator approves your request.

Results

You have selected a Plan.

What to do next

Manage and monitor your API and application usage.

Related tasks

- Registering an application

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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:

1. Click API Products, then click API under the name of the Product that contains the API you want to work with.
2. Click the API that you want to work with.
   The API overview page opens.
3. Under the API Operations heading, copy the URL.
   This is the URL that the application calls, and its structure is defined:

Type of API	URL
REST APIs	**`https://`*host*`/`*org*`/`*catalog*`/`*api*`/`*operation***
SOAP APIs	**`https://`*host*`/`*org*`/`*catalog*`/`*api*** 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:

5. To find the client ID, complete the following steps:
   a. Click Apps, then click the application name that you want to work with.
   b. Select the Show check box for Client ID.
      The client ID is displayed.
   c. Provide the client ID with the query 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
```

6. 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.

7. 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
   ```

8. 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Calling an API from a CORS-enabled browser is as simple as setting the Origin header on the API request to the value of the origin server of your application.
CORS is supported in the following browsers:

- Chrome 3+
- Firefox 3.5+
- Internet Explorer V11, or later
- Opera 12+
- Safari 4+

Procedure

1. In your application, set the headers according to the following sample request.

```
OPTIONS /org/env/api/resourceHTTP/1.1
User-Agent: useragent details
X-IBM-Client-Id=1234
Host: x.xx.xxx.xx
Origin: http://example/example/testui.html
Accept: */*
```

2. Your browser handles the preflight check.

Results

The following sample response is received:

```
HTTP/1.1 200 OK
X-Backside-Transport: FAIL FAIL
Connection: Keep-Alive
Transfer-Encoding: chunked
Access-Control-Allow-Origin: http://example/example/testui.html
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers:accept, accept-language, content-type, x-ibm-client-id
Access-Control-Allow-Method: <methods allowed on the resource>
Vary: Origin
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Signing up to use an API

You can sign up to use an API in the Developer Portal.

Before you begin

You must have registered an application in order to sign up to use an API.

Procedure

1. From the Developer Portal home page, click API Products.
2. Click APIs, then click the name of the API you want to work with, and click Use this API.
3. In the Plans section, identify the Plan that you want to work with from the list of Plans available on the left, then click Use this plan.
4. Use the radio buttons to select the application with which you want to use this Plan, then click Save.
   You have signed up to use an API.

- **Viewing application subscriptions**
  You can view which APIs your application is subscribed to in the Developer Portal.
- V5.0.8 + **Adding credit card information for paid subscriptions**
  Before you can subscribe to a Plan with billing in a Developer Portal, you must add your credit card information.
- V5.0.8 + **Changing to a different Plan**
  You can switch to a different Plan for the same Product in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing application subscriptions

You can view which APIs your application is subscribed to in the Developer Portal.

Procedure

1. From the Developer Portal home page, click Apps.
2. Click the name of the target application.
   Details of the Plans and APIs subscribed to by your application are displayed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Adding credit card information for paid subscriptions

Before you can subscribe to a Plan with billing in a Developer Portal, you must add your credit card information.

Before you begin

You might want to subscribe to a Plan in the Developer Portal that requires a paid subscription. If this is the case, you must provide your credit card information before you can subscribe to the Plan.
Remember: You must have organization owner permission to complete this procedure. If you do not have owner access, the billing information is not displayed.

About this task

To enter your credit card information in your API Connect account so you can subscribe to Plans with billing, complete the following steps:

Procedure

1. Log in to the Developer Portal that contains the Product that you want to subscribe to.
2. Select your user name in the title bar, and select My organization.
3. Select the Billing tab.
   If you do not see the Billing tab, you do not have owner privileges to your organization.
4. Select Update to add your credit card information.
5. Complete the fields with your Billing Information, and select Payment Info.
6. Enter your credit card information in the required fields, and select Update Billing Details.

Results

Your credit card information is saved in your account. Your credit card will be billed according to the paid Plans that you subscribe to, following any free trial days that are included in the Plans. Monthly subscriptions are billed the same time each month.

Related information

- Tutorial: Defining a subscription Plan with pricing for your API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Changing to a different Plan

You can switch to a different Plan for the same Product in the Developer Portal.

Before you begin

There are times when you have to switch Plans after you have been using a Plan. The following provide some examples of why you might need to change Plans:

- The Plan changed from a free Plan to a paid Plan in the Developer Portal.
- There was a change in the cost of the Plan.
- You want to change from your current Plan to one that provides you with a different service level.

If you change from a free plan to a paid Plan, you must provide your credit card information before you can subscribe to the Plan. Remember: You must have organization owner permission to complete this procedure. If you do not have owner access, the billing information is not displayed.

About this task

To change to another Plan, complete the following steps:

Procedure

1. Log in to the Developer Portal that contains the Product that you want to change.
2. If you are changing to a paid Plan, make sure that your credit card information is updated. See Adding credit card information for paid subscriptions for more information.
3. Select Products.
4. Select the name of the Product that you want to change.
5. If a migration path is recommended or required by the owner of the Product, select the Migrate button that is beside the name of the Product.
   The path for migration is already determined, so there is no selection that is required.
6. If no migration path was recommended and you are changing to a Plan that is not on the migration path, select Unsubscribe next to the Product that you want to change.
7. Return to the list of Plans for that product and register for the new Plan.

Results

You are changed to the new Plan at the new price, if it is a Plan with billing. There are no free trial days when you change, and billing starts on the day of the change. Monthly billing Plans are billed at the same time every month.

Related information

- Tutorial: Defining a subscription Plan with pricing for your API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Testing an API 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 Developer Portal and click Invoke to see the response.

Before you begin

To use the Developer Portal test tool with an API that requires the application client ID, complete the following tasks first.

- Registering an application
- Selecting a Plan

Important:

- To test an API in an application within the Developer Portal, the Allow this API to be tested check box must be selected in API Manager. For more information, see Creating API definitions.

- You can only test an unenforced API if `testable=true`, and if the existing API implements CORS and is using HTTPS.

About this task

The Developer Portal test tool is an interactive API document test tool. You can use the Developer Portal test tool without the need to sign in. If the operation you want to interact with requires a client ID, then you must sign in.

Note that 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 you click invoke within the minute interval. This caveat affects the quota of the application that is selected to use with the Developer Portal test tool, but it does not affect the quota of the other applications using the same operation or Plan.

Restriction:

- You cannot use the Developer Portal test tool to test a public scheme in an OAuth provider API, you can only test a confidential scheme. For more information about OAuth schemes, see Tutorial: Securing an API by using OAuth 2.0.
- Also, you cannot use 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.
- You cannot use the Developer Portal test tool with suspended applications.

Procedure

1. To use the Developer Portal test tool with an API that does not require the client ID of an application, complete the following steps:
    a. Click APIs.
    All of the APIs that can be used by application developers are displayed.
    b. Click the name of the API that you want to test.
    c. Find the operation you want to work with under the API Operations heading.
    d. Click Try this operation.
    e. Supply the values for required headers or parameters.
    f. If the operation is secured with Basic Authentication, supply the credentials.
    g. 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 you click Try this operation, you are 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.
2. To use the Developer Portal test tool with an API that requires the client ID of an application, complete the following steps:
    a. Sign in to the Developer Portal.
    b. You must register an application to test an API that requires the client ID of an application. For more information, see Registering an application.
    c. Ensure that your application is registered to use a Plan that contains the API you want to test. For more information, see Selecting a Plan.
    d. Click APIs.
    All of the APIs that can be used by application developers are displayed.
    e. Click the *API* that you want to test.
    f. Under the API Operations heading, and the Security subheading, click Provide credentials for Client ID (API Key).
    The API Key Identification window displays.
    g. Use the Register an application in order to select a client ID drop-down list to select an application to test the API with.
    h. Click Save credentials.
    i. If a client secret is required, enter the value in the Client Secret field.
    j. Find the operation you want to test under the API Operations heading, then click Try this operation.
    k. Supply the required parameters and values.
    l. If the operation is secured with Basic Authentication, supply the credentials.
    m. Click Send Request.
    The result is displayed in the Response section. You can continue to test different field values as necessary.
    Note: The first time you click Invoke, you are 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Downloading a WSDL file for a SOAP API

You can download a WSDL file so that you can use it to generate a client stub to invoke the SOAP API. For example, by using **wsdl2java** or a similar approach.

Before you begin

Sign in to use the Developer Portal. For more information, see The Developer Portal user interface.

About this task

A client stub is a piece of code that is generated automatically from the service definition (WSDL). A client stub makes it easy to call the service by wrapping up all the technical details into a simple set of method calls. In Java, you can use the command **wsdl2java** to turn the WSDL file into a set of Java classes, called the Java client stub.

You can download a WSDL file from your organization's Developer Portal, or by using a URL.

Procedure

To download a WSDL file from the Developer Portal, complete the following steps:

1. Click API Products.
   All of the API Products that can be used by application developers are displayed.
2. Click the API Product title that contains the API for which you want to download the WSDL file.
3. Click the WSDL file name to download the file.

Results

The WSDL file is downloaded.

What to do next

Use the WSDL to generate a client stub that you can use to invoke the SOAP API.

Related information

- IBM API Connect overview
- Adding a SOAP API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Applications in the Developer Portal

You can view, edit, configure, and create applications in the Developer Portal.

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.
- V5.0.7+ **Upgrading a Development application to Production status**
  If development application workflow is enabled for the Catalog associated with your Developer Portal, any new application that you register initially has Development status, and can call APIs in the Catalog only through dedicated development endpoints. When you

have completed your application testing, you can request to upgrade your application to Production status. After the upgrade request is approved, the application can call APIs through production endpoints.

- **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
- **Setting up notifications for an application**
  You can enable notifications for applications so that you are alerted when the usage of an API is nearing its rate limit. If you notice that an application is reaching its rate limit, you can take action such as changing the Plan to avoid impacting the users of your application.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. V5.0.5 + Click Create new App.
   V5.0.4 and earlier Click Register new Application.
   V5.0.7 + Note: If development application workflow is enabled for the catalog that is associated with your Developer Portal, a Development label is displayed alongside the title of your application. Your application can call APIs in the catalog only through dedicated development endpoints. When you complete your application testing, you can request to upgrade your application to Production status. After the upgrade request is approved, your application can call API calls through production endpoints. For more information, see Upgrading a Development application to Production status.
3. Complete the displayed fields.
   Note: Do not include a double quotation mark, ", or backslash, \, character in the `Title` field. Including these characters causes errors when you generate an access token in the OAuth process.
4. Optional: V5.0.8 + In the Certificate field, paste the X509 certificate for your application, in PEM format.
   The Certificate field is available only if the APIs that are published to the Developer Portal include at least one API that is secured with TLS mutual authentication. You must complete this field if you want to call an API that is secured with TLS mutual authentication. For more information, see Composing a REST API definition.

5. Click Submit.
   Your application is displayed.
6. Make a note of your client secret because it is only displayed 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.

7. Optional: The client ID is hidden, so to display the client ID for your application, select the Show checkbox for Client ID.
   The client ID is displayed and can be hidden again by clearing the checkbox .
8. Optional: To verify your client secret, click Verify next to Client Secret, enter your client secret in the Secret field, then click Submit.
   You 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 Add alongside Client Credentials.
      The "Request additional client credentials" window opens.
   b. Enter an optional description, and click Submit.
   c. Select the Show Client ID or Show Client Secret check box to display the client ID or client secret for the new credentials.
   d. To add a description to a set of client credentials, or to change the current description, click Update alongside the required credentials.
   e. 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. An application can have at most 20 client ID/client secret pairs.
   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. Optional: To add an image, click Update under the default image.
    A new window opens; click Browse, select an image from your directory, and click Submit.
11. Optional: To specify or change the URL that authenticated OAuth flows for this application should be redirected to, click the Edit icon and then update the OAuth Redirect URI field.
    V5.0.8 + From API Connect Version 5.0.8.1, you can specify multiple OAuth redirect URLs by separating them with a comma (there is a strict limit of 2048 characters for this field). For example, `https://abc.redirect.com,https://def.acme.redirect.com`. 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.

12. Optional: To change the application name or description, or verify or reset the client secret, click the Edit icon.

Results

Your application is registered.

What to do next

Select a Plan to use with your application, see Selecting a Plan.

Related information

- IBM API Connect overview
- Setting application identification requirements

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.7 +

Upgrading a Development application to Production status

If development application workflow is enabled for the Catalog associated with your Developer Portal, any new application that you register initially has Development status, and can call APIs in the Catalog only through dedicated development endpoints. When you have completed your application testing, you can request to upgrade your application to Production status. After the upgrade request is approved, the application can call APIs through production endpoints.

Procedure

To request to upgrade a Development application to Production status, complete the following steps:

1. From the Developer Portal home page, click Apps.

2. Select the application that you want to upgrade.
3. Click Upgrade to production, then click Request upgrade to confirm the upgrade request.

Results

An upgrade approval request is submitted. Your application continues to have Development status while the approval request is pending, and a Pending Upgrade label is displayed on the application details page. When the request is approved, you will receive a notification, and the application will move automatically to Production status.

If the request is rejected, the status of the application remains unchanged, and the Pending Upgrade label is removed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. 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.
4. To reset your client ID, click Reset.
   Warning: API calls made by the application with the existing client ID will fail.
5. If the client secret is compromised or you forget it, click Reset to generate a new value for the client secret.
6. 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.
7. To add an additional client ID and client secret to the application, complete the following steps:
   a. Click Add alongside Client Credentials.
      The "Request additional client credentials" window opens.
   b. Enter an optional description, and click Submit.
   c. Select the Show Client ID or Show Client Secret check box to display the client ID or client secret for the new credentials.
   d. To add a description to a set of client credentials, or to change the current description, click Update alongside the required credentials.
   e. 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. An application can have at most 20 client ID/client secret pairs.
   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.
8. To view more details about an individual API, click the API name.
   The operations are listed.
9. To unsubscribe from a Plan, click Unsubscribe alongside the Plan name.

Related tasks

- Setting up notifications for an application

Related information

- IBM API Connect overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.
4. Make any required changes, then click Submit.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click the application for which you want to change the image, and click Update.
3. Browse for the target image, then click Submit.
   You can remove the image by clicking Remove.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 Verify for the client secret.
4. 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Setting up notifications for an application

You can enable notifications for applications so that you are alerted when the usage of an API is nearing its rate limit. If you notice that an application is reaching its rate limit, you can take action such as changing the Plan to avoid impacting the users of your application.

About this task

You can control the frequency and method of receiving notifications that inform you when the usage of an API is approaching and exceeding the rate limit set on the Plan you selected.
You can view notifications from the built-in activity feed and by email with the option to turn off email notifications.

Note: If you have one or both of the following conditions, then you cannot receive notifications:

- The API developer disabled the activity log policy for an operation.
- The name of the rate limit in the Plan that you are subscribing to is not one of the names that is valid for the notifications to function; for more information, see Add a rate limit to your Plan.

Procedure

To set up notifications for an application, complete the following steps:

1. Click Apps.
   The Apps page opens.
2. Select the application that you want to work with.
3. Click the Notifications tab.
   The notification settings page is displayed.
4. Select the Enable notifications for this application check box.
5. By default you always receive notifications when the rate limit is exceeded. If you want to be notified before the rate limit is exceeded, select one of the following options:
   - 50% of the Rate limit is reached
   - 75% of the Rate limit is reached
   - 90% of the Rate limit is reached
6. Select how frequently you want to receive a notification from one of the following options.
   - Every minute
   - Every hour
   - Every day

Note: You only receive one notification per operation per rate limit threshold within your chosen time interval. For example, if you choose Every hour and the 75% rate limit is reached twice, you are notified about it only once.

7. After you enable notifications, you always receive them in the built-in activity feed. If you also want to receive notifications by email, select the Email check box.

Results

The notification settings for your application are defined.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 associated with the Developer Portal is using Portal Delegated User Registry, portal analytics is disabled. For information on configuring a Catalog to use Portal Delegated User Registry, see Selecting the Portal Delegated User Registry.
V5.0.7+ Note: 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 developer 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.
3. Click Analytics icon 📊 .
   From the Analytics page, you can interact with the following analytic graphs:

   Success rate

   You can view a graphical summary of the success rate for all API calls used by your application by using the interactive Success rate graph. This graph describes the data in terms of API calls over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:
   - Success - describes a summary of the number of successful calls to APIs used by your application.
   - Failure - describes a summary of the number of failed calls to APIs used by your application.
   - Total - describes the total number of calls made to APIs used by your application.

   Data usage (bytes)

You can view a graphical summary of the data usage for all APIs used by your application with the interactive graph. This graph describes the data in terms of bytes of data over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:

- Maximum - each datapoint describes the maximum data usage in a set time period.
- Average - each datapoint describes the average data usage in a set time period.
- Minimum - each datapoint describes the minimum data usage in a set time period.

Latency (ms)

You can view a graphical summary of the latency of all APIs used by your application with the interactive graph. This graph describes the data in terms of milliseconds of latency over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:

- Maximum - each datapoint describes the maximum period of latency for API calls made in a set time period.
- Average - each datapoint describes the average period of latency for API calls made in a set time period.
- Minimum - each datapoint describes the minimum period of latency for API calls made in a set time period.

Individual data points

You can, on all graphs within the analytics page, hover over individual data points to view their exact value.

Analytics across date and time ranges

You can zoom in on analytics data for a particular date and time range by clicking and dragging the cursor across sections of any graphs that represent dates and times. To remove the resulting filter, you can either click the browser Back button, or use the Time Picker icon to select the previous, or a different time range. For more information, see Drilling down into data in your time-based visualizations.

V5.0.7+ You can use the legend on the charts to apply filters or to change the colors on the charts. For more information, see Applying a filter by using the legend in a visualization. V5.0.6 and earlier You can use the legend on the charts to apply filters to the charts, as described in Applying a filter by using the legend in a visualization

V5.0.7+ You can also click the Time Picker icon 🕘 to open the Time Picker and set a time filter: V5.0.6 and earlier You can also click the Time Picker icon 📅 to open the Time Picker and 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 above 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. Click Go.

When you open the Time Picker, Auto-refresh is shown to the left of the Time Picker icon. To specify a frequency at which the data should automatically be refreshed in your charts, click Auto-refresh and then select a predefined refresh interval. The auto-refresh interval is displayed to the left of the Time Picker icon, together with a Pause icon that you can use to pause auto-refresh if required. V5.0.7+ Click the auto-refresh value, which is displayed next to the Time Picker icon 🕘, to confirm your settings and close the time selection panels.

V5.0.7+ If the Time Picker panel is open, you can close it by clicking within the box where the Time Picker icon 🕘 is located. 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.

V5.0.6 and earlier You can close the Time Picker by clicking the caret (^) at the bottom of the panel. To switch off the auto-refresh capability, you must open the Time Picker again, click the auto-refresh value to the left of the Time Picker icon, and click Off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 developer organization can view analytics information relating to the APIs that 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 page, you can interact with the following analytic graphs:

   Success rate

   You can view a graphical summary of the success rate for all API calls used by your application by using the interactive Success rate graph. This graph describes the data in terms of API calls over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:

   - Success - describes a summary of the number of successful calls to APIs used by your application.
   - Failure - describes a summary of the number of failed calls to APIs used by your application.
   - Total - describes the total number of calls made to APIs used by your application.

   Data usage (bytes)

   You can view a graphical summary of the data usage for all APIs used by your application with the interactive graph. This graph describes the data in terms of bytes of data over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:

   - Maximum - each datapoint describes the maximum data usage in a set time period.
   - Average - each datapoint describes the average data usage in a set time period.
   - Minimum - each datapoint describes the minimum data usage in a set time period.

   Latency (ms)

   You can view a graphical summary of the latency of all APIs used by your application with the interactive graph. This graph describes the data in terms of milliseconds of latency over time, and you can use the manouverable time scale to determine the time-frame over which you view your application data. The graph shows a linear representation of the following factors:

   - Maximum - each datapoint describes the maximum period of latency for API calls made in a set time period.
   - Average - each datapoint describes the average period of latency for API calls made in a set time period.
   - Minimum - each datapoint describes the minimum period of latency for API calls made in a set time period.

   Individual data points

   You can, on all graphs within the analytics page, hover over individual data points to view their exact value.

   Analytics across date and time ranges

   You can zoom in on analytics data for a particular date and time range by clicking and dragging the cursor across sections of any graphs that represent dates and times. To remove the resulting filter, you can either click the browser Back button, or use the Time Picker icon to select the previous, or a different time range. For more information, see [Drilling down into data in your time-based visualizations]().

   V5.0.7+ You can use the legend on the charts to apply filters or to change the colors on the charts. For more information, see [Applying a filter by using the legend in a visualization](). V5.0.6 and earlier You can use the legend on the charts to apply filters to the charts, as described in [Applying a filter by using the legend in a visualization]()

   V5.0.7+ You can also click the Time Picker icon 🕘 to open the Time Picker and set a time filter: V5.0.6 and earlier You can also click the Time Picker icon 📅 to open the Time Picker and 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 above 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. Click Go.

   When you open the Time Picker, Auto-refresh is shown to the left of the Time Picker icon. To specify a frequency at which the data should automatically be refreshed in your charts, click Auto-refresh and then select a predefined refresh interval. The auto-refresh interval is displayed to the left of the Time Picker icon, together with a Pause icon that you can use to pause auto-refresh if required. V5.0.7+ Click the auto-refresh value, which is displayed next to the Time Picker icon 🕘, to confirm your settings and close the time selection panels.

   V5.0.7+ If the Time Picker panel is open, you can close it by clicking within the box where the Time Picker icon 🕘 is located. 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.

   V5.0.6 and earlier You can close the Time Picker by clicking the caret (^) at the bottom of the panel. To switch off the auto-refresh capability, you must open the Time Picker again, click the auto-refresh value to the left of the Time Picker icon, and click Off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Developer organizations in the Developer Portal

You can add, remove, and configure Developer organizations in the Developer Portal.

To learn about Developer organizations in the Developer Portal, use the following topic links:

- **Adding a Developer organization from within the Developer Portal**
  You can create a new Developer organization from within the Developer Portal. The new Developer organization appears in the API Manager UI after it is created.
- **Editing the name of a Developer organization**
  You can edit the name of your Developer organization from within the Developer Portal.
- **Switching Developer organizations**
  You can switch from one Developer organization to another in the Developer Portal.
- **Adding users to a Developer organization**
  If the API provider organization administrator has granted you permission to collaborate, you add users to your Developer organization with the Users panel. Those users can then access the Developer Portal and use the APIs that have been made available to the Developer organization.
- **Removing a user from a Developer organization**
  You can remove a developer from a Developer organization in the Developer Portal.
- V5.0.8 + **Changing the ownership of a Developer organization**
  From API Connect Version 5.0.8.2, you can change the ownership of a Developer organization in the Developer Portal.
- V5.0.8 + **Deleting a Developer organization**
  From API Connect Version 5.0.8.2, you can delete a Developer organization in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a Developer organization from within the Developer Portal

You can create a new Developer organization from within the Developer Portal. The new Developer organization appears in the API Manager UI after it is created.

Procedure

1. Click the arrow next to `your_user_name` from the Developer Portal home page.
2. Select Create organization from the drop-down menu.
3. In the Organization name field, type the name of the Developer organization, then click Submit.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Editing the name of a Developer organization

You can edit the name of your Developer organization from within the Developer Portal.

Procedure

1. On the Developer Portal home page, click your account name, then click My organization.
2. In the Manage tab, click Edit Organization.
3. Enter the new name for your Developer organization in the Organization name field, then click Submit.
   You have changed the name of your Developer organization.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Switching Developer organizations

You can switch from one Developer organization to another in the Developer Portal.

Procedure

1. If you are a member of more than one Developer organization, click the menu of Developer organizations in the upper-right corner of the Developer Portal home page.
2. Select the Developer organization that you want to switch to, and ensure that any applications you register are registered to the correct Catalog.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding users to a Developer organization

If the API provider organization administrator has granted you permission to collaborate, you add users to your Developer organization with the Users panel. Those users can then access the Developer Portal and use the APIs that have been made available to the Developer organization.

About this task

Depending on the permissions enabled by the provider organization administrator, collaborators can view application and application activity, create and edit applications, manage client keys and subscribe to Plans, or perform all activities.

Procedure

To invite users to join your Developer organization, complete the following steps.

Note: The Users panel is only visible if the API provider organization administrator has granted you permission to collaborate.

1. In the upper-right corner of the Developer Portal, click your user name, and select My organization.
   The summary page for your organization opens.
2. Click Add a user.
3. Enter the e-mail address of the new user in the New user's e-mail address field.
4. Under the Role heading, use the radio buttons to select a role for the new user, the roles are:
   - **App Developer**-- Can create and edit applications, manage client keys and subscribe to 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.
5. Click Invite new user.

Results

The user is added to the Developer organization, and they are sent an email with the subject line: "Welcome to *developer_organization_name*". The user must click the link that is provided to activate their account and complete the setup.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing a user from a Developer organization

You can remove a developer from a Developer organization in the Developer Portal.

Before you begin

You must be the owner of your Developer organization and have previously added the developer you wish to remove from the 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Changing the ownership of a Developer organization

From API Connect Version 5.0.8.2, you can change the ownership of a Developer organization in the Developer Portal.

Before you begin

You can change the ownership of a Developer organization only if you are the owner of that organization. The user that you transfer ownership to must already be a member of the organization. Note that to have an account on a Developer Portal, you must be a member of at least one Developer organization.

About this task

Transferring the ownership of a Developer organization to another organization member, changes your role in that organization to App Developer.

Procedure

1. Select the Developer organization that you want to transfer ownership of by using the drop-down arrow in the upper right corner of the Developer Portal home page.
2. Click your user name in the Developer Portal home page, and select My organization.
3. Click Change ownership, and select the user to take ownership.
4. Click Change to confirm the transfer.
   This action transfers the Organization Owner role to the new owner, and changes the role of the previous owner to be App Developer.

Results

You successfully transferred ownership of a Developer organization in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Deleting a Developer organization

From API Connect Version 5.0.8.2, you can delete a Developer organization in the Developer Portal.

Before you begin

You can delete a Developer organization only if you are the owner of that organization. Note that to have an account on a Developer Portal, you must be a member of at least one Developer organization. Therefore, if you are a member of only one Developer organization, to delete that organization you must either join or create another Developer organization first, or you can delete your whole Developer account. For more information, see Deleting your Developer account.

About this task

Deleting a Developer 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 Developer organizations before you delete the organization. Any users that were members of only the deleted organization, must create a new Developer organization when they next logon to the Developer Portal.
Important: When an organization has been deleted, it cannot be reactivated. You might want to consider changing the ownership of a Developer organization, rather than deleting it. For more information, see Changing the ownership of a Developer organization.

Procedure

1. If you own more than one Developer organization, select the organization that you want to delete by using the drop-down arrow in the upper right corner of the Developer Portal home page.
2. Click your user name in the Developer Portal home page, and select My organization.
3. Click Delete organization.
4. Click Confirm to delete the organization.
   This action permanently removes access to the organization, and all of its applications and subscriptions, for all members of the organization. When an organization has been deleted, it cannot be reactivated.

Results

You permanently deleted a Developer organization in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

User accounts, passwords, and support in the Developer Portal

You can change user properties such as `timezone` and `passwords` in the Developer Portal, for your own user.

Important:

- Email addresses for all Developer Portal users must be unique, including the email address that is used to create the Administrator account.
- All email addresses must be unique in the portal with a 1:1 mapping between them. Even if your user registry allows you to have more than one email address in the attributes, only the first email address works for your site.
- It is not possible to change the email address of any user.

To learn about general user features of the Developer Portal, use the following topic links:

- **Creating a new developer account**
  You can create new developer accounts in the Developer Portal.

- **Changing your password**
  You can change your password in the Developer Portal password.
- **Change user settings**
  You can edit user settings in the Developer Portal, including timezone, language, and account settings.
- V5.0.8 + **Deleting your Developer account**
  From API Connect Version 5.0.8.2, you can delete your account in the Developer Portal.
- V5.0.2 + **Adding identities in the Developer Portal**
  In addition to logging in with your administrator with the accounts credentials, you can add identities to your administrator account by using the credentials of an account that belongs to an external authentication provider.
- **Raising a support ticket**
  If you need help, you can raise a support ticket in the Developer Portal. This support ticket will be forwarded to the portal administrator.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a new developer account

You can create new developer accounts in the Developer Portal.

Procedure

1. From the Developer Portal home page, click Create an account..
2. Populate the available fields with the details of the developer account you want to create.
3. Click Create new account.
   The e-mail address you provided for the new developer account will receive an e-mail containing an activation link for the new developer account.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your password

You can change your password in the Developer Portal password.

Procedure

1. From the Developer Portal home page, click the link for your user name.
2. From the drop-down menu, select My account.
3. Click Password.
4. Enter your current password, then enter your new password in the Password and Confirm password fields.
5. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Change user settings

You can edit user settings in the Developer Portal, including timezone, language, and account settings.

To learn how to change settings, use the following topics:

- **Changing your account settings**
  You can edit your Developer Portal account settings.
- **Changing your language setting**
  You can change your language setting in the Developer Portal
- **Changing your timezone setting**
  You can change your timezone setting in the Developer Portal.
- **Changing your application notification settings**
  You can change your application notification settings in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your account settings

You can edit your Developer Portal account settings.

Procedure

1. Click your user name in the Developer Portal home page.
2. Select My account, then click Edit.
3. From this view you can change multiple features of your account.
4. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your language setting

You can change your language setting in the Developer Portal

Procedure

1. Click your username from the Developer Portal home page.
2. Click Edit.
3. You can change the language from this view.
4. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your timezone setting

You can change your timezone setting in the Developer Portal.

Procedure

1. Click your user name from the Developer Portal home page.

2. Select My account, then click Edit.
3. You can change the timezone from this view, under the LOCALE SETTINGS tab.
4. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing your application notification settings

You can change your application notification settings in the Developer Portal.

Procedure

1. From the Developer Portal home page, click Apps.
2. Select the target application.
3. Click Notifications.
4. Enable notifications, then configure what you require notification for.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Deleting your Developer account

From API Connect Version 5.0.8.2, 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 Developer organization. If you do own more than one Developer organization, you must either change the ownership of those organizations, or delete them, before you can delete your account. If you own only one Developer organization, when you delete your account the Developer organization is deleted at the same time as your account.

About this task

When you delete your account, any Developer organizations that you own must also be deleted or transferred to new owners. If there are other members in a Developer 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 Developer organizations before you delete your account. For more information, see Deleting a Developer organization.
Important: When an organization has been deleted, it cannot be reactivated. You might want to consider changing the ownership of your Developer organizations before you delete your account. For more information, see Changing the ownership of a Developer 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 Confirm to delete the account.
   This action deletes your Developer account and, if you own a Developer organization, this organization is also deleted.

Results

You successfully deleted your Developer account in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Adding identities in the Developer Portal

In addition to logging in with your administrator with the accounts credentials, you can add identities to your administrator account by using the credentials of an account that belongs to an external authentication provider.

Before you begin

You must have administrator access to complete this task.

You must have enabled Portal Delegated User Registry in the API Manager UI to complete the procedure. For more information, see Portal Delegated User Registry.

The external authentication provider that you want to add as an identity must be enabled. For more information, see Using external authentication provider credentials to access the Developer Portal.

About this task

By adding external authentication provider identities to your administrator account, you can improve the ease and speed to which you log into the Developer Portal with your administrator account.

Procedure

1. In the Developer Portal, click the administrator account name drop down list, then click the name of the administrator account.
2. Click HybridAuth.
3. In the Add more identities section, click on the external authentication provider that you want to create an additional identity for.
4. Progress through the external steps that authorize the connection between the external authentication provider and your Developer Portal site.

Results

Details of the account that belongs to your external authentication provider are displayed under the Add more identities section as it has become a new identity for your administrator account. When you want to log in to the Developer Portal site as an administrator, click the button that represents the external authentication provider, and you are logged into the Developer Portal as the administrator.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Raising a support ticket

If you need help, you can raise a support ticket in the Developer Portal. This support ticket will be forwarded to the portal administrator.

Procedure

1. From the Developer Portal home page, click Support.
2. Click Raise a support ticket.
3. Click Post new support ticket.
4. Enter a name and body into the relevant fields, then click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configure bookmarks in the Developer Portal

You can configure, add, and remove bookmarks in the Developer Portal.

To learn about bookmarks in the Developer Portal, use the following topic links:

- **Bookmarking content**
  You can create bookmarks for any piece of content in the Developer Portal. Your bookmark profile is individual to you and will not be available to other users.
- **Accessing bookmarks**
  You can access your personal bookmarks in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Bookmarking content

You can create bookmarks for any piece of content in the Developer Portal. Your bookmark profile is individual to you and will not be available to other users.

Procedure

1. From the desired content element, click Bookmark this.
   You have bookmarked the required content element.
2. You can remove bookmarks by clicking Unbookmark this.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Accessing bookmarks

You can access your personal bookmarks in the Developer Portal.

Procedure

1. From the Developer Portal home page, click the link for your user name.
2. Select My bookmarks from the drop-down menu.
   You can view your personal bookmarks from this display.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configure the Developer Portal site

Depending on the permissions that the role of a Developer Portal user possesses, you can configure features of the Developer Portal.

A user can be assigned one of four Developer Portal roles:

Authenticated user
: An authenticated user can perform all of the actions that are described in the Use the Developer Portal section.

Forum Moderator, Content Author, and Administrator
: In addition to performing all of the tasks that an Authenticated user role can, a Forum Moderator, Content Author, and Administrator role have additional permissions. For more information, see Working with roles in the Developer Portal.
Note: The Administrator role cannot configure APIs, Products, and Apps.

Important: Editing API Connect themes, modules, included modules, or Drupal core on the filesystem is not recommended. For more information, see Edit themes and modules.
By default, the Developer Portal administration operations are available only if you have administrator access.

After you have created your Developer Portal site, some first returned products appear in the "popularproduct" block on the front page of the Developer Portal. The products in the "popularproduct" block can be changed.

You can perform the following tasks to customize and configure your Developer Portal site if your role has the correct permissions:

- Change the front page banner image
- Integrating Twitter data into the Developer Portal social feed
- Customizing themes for the Developer Portal
- Assigning people to a role
- Assigning permission to a role

In addition to the aforementioned tasks, Developer Portal configuration operations are described in the following sections.

- **Concepts in the Developer Portal**
  You are recommended to understand the various concepts and terminology that is referenced throughout the Developer Portal. The exposure of the concepts is dependent on the permissions that your role possesses.
- **Edit themes and modules**
  Editing API Connect themes, modules, included modules, or Drupal core on the filesystem is not supported. Edited versions of these files are overwritten when a fix pack or iFix is installed.
- **Forum Moderator actions in the Developer Portal**
  You can create and control forums in the Developer Portal.
- **Content Author actions in the Developer Portal**
  You can control multiple content elements in the Developer Portal including the customization and restriction of specific elements.
- **Administrator actions in the Developer Portal**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Concepts in the Developer Portal

You are recommended to understand the various concepts and terminology that is referenced throughout the Developer Portal. The exposure of the concepts is dependent on the permissions that your role possesses.

Some concepts are linked to other concepts, and you are recommended to understand how the following concepts function independently, but also work together:

Nodes

A node is a piece of individual content such as a page, poll, article, forum topic, or a blog entry. Each piece of content in a Developer Portal site is a node. You can apply new features or changes to all content of a single type.

Fields

A field is something that allows you to add additional 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

Themes enable you to control the appearance of your Developer Portal site. You can modify the appearance of the theme in following ways:

1. Identify and use a theme that is provided by the Drupal community or third-party website, and modify the theme settings
2. Modify or extend the code of an existing theme
3. Create a complete theme from scratch (For doing this you need to copy files from core theme and place it under sites/all/themes and do the necessary changes on .info, template, tpl and css files.)

You can also create a sub-theme that inherits all of the settings of its parent theme, unless the settings are overridden. For more information, see Using a sub-theme.

Most themes make use of the PHPTemplate layout engine.

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.

Modules

Modules are similar to the concept of plugins, in that they extend the core functionality of the Developer Portal site. A set of modules is implemented by default with the Drupal core, and there are additional modules that can be enabled to extend the default functionality. You can find and add more modules to your Developer Portal site from the internet.

Views

Views enable you to manipulate the content that is displayed on a page, block, and other visual elements in the Developer Portal site. Views can be used in conjunction with content types to format the appearance of your site to your specification.

Panels

Panels are a module that allow a site administrator to create customized layouts for multiple uses. It is a drag and drop content manager that allows you to visually design a layout, and place content within that layout. Integration with other systems allows you to create nodes and landing pages that use this, and you override system pages such as taxonomy and the node page. Panels enable you to customize the layout of your site with detailed permissions.

Pages

Pages are used by panels and other appearance modifying features of the Developer Portal to customize the appearance of your Developer Portal. Pages can be customized to be as specific as you require, and can be configured to satisfy the context of the situation 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 will always be a local account to enable administration of the Developer Portal site.

Roles

A role is a collection of permissions that define the actions that a user can perform 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 will be 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.

Permissions

Permissions define the actions that a user can or cannot perform in a Developer Portal site. Permissions are additive. If a permission that enables a user to perform an action is not assigned, the user cannot perform the action. If a user has multiple roles, and any of them contain a specific permission, the user will be able to perform that action. There are also permissions have security implications, and you are recommended to assign those to trusted roles.

Templates

Template files define how the output of a given component can look. They are formatted as PHPTemplate files. The following levels are the examples of where you can use templates:

- html.tpl.php - the template for Developer Portal HTML pages.
- page.tpl.php - the body of Developer Portal HTML page
- node.tpl.php - the template for all of the content nodes
- comment.tpl.php - the template for all of the comment nodes
- search-result.tpl.php - the template for search results
- node—product.tpl.php - the template for nodes that use the Product content type
- node—product—teaser.tpl.php - the template for the previews for the Product content type.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Edit themes and modules

Editing API Connect themes, modules, included modules, or Drupal core on the filesystem is not supported. Edited versions of these files are overwritten when a fix pack or iFix is installed.

Important: Editing API Connect themes, modules, included modules, or Drupal core on the filesystem is not permitted or supported.
To edit a theme, the recommended procedure is to create a sub-theme. For more information on creating sub-themes, see Using a sub-theme.
You can edit the .css file in a theme to alter the appearance of your Developer Portal site. For more information, see Customizing themes for the Developer Portal.
Important: You are not permitted to include any IBM® API Connect code within any custom themes that you create.
To edit modules or functionality, the recommended procedure is to create a custom module. For more information on creating custom modules, see Creating custom modules to extend functionality.
Important: You are not permitted to include any IBM API Connect modules within any custom modules that you create.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Forum Moderator actions in the Developer Portal

You can create and control forums in the Developer Portal.

To learn how to create and control forums in the Developer Portal, 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.
- **Configuring access to forums**
  You can make preexisting forums private in the Developer Portal. You can also configure other forum access features to customize forum privacy for different user roles.
- **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.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Structure in the administrator dashboard.
2. Click Forums.
3. Click Add forum.
4. Fill in the Forum name and Description fields for the new forum.
5. From the Parent drop-down list, select the hierarchical position of the new forum.
6. Click Save. You have created a new forum.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Structure in the administrator dashboard.
2. Click Forums.
3. Click Add container.
4. Fill in the Container name and Description fields for the new forum container.
5. From the Parent drop-down list, select the hierarchical position of the new forum container.
6. Click Save. You have created a new forum container.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Configuration in the administrator dashboard.
2. Under the SYSTEM heading, click IBM API Connect.
3. Select or deselect the "Automatically create a forum per API" check box to turn the feature on or off.
4. Click Save configuration.
   You have configured the creation of new forums for each API.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring access to forums

You can make preexisting forums private in the Developer Portal. You can also configure other forum access features to customize forum privacy for different user roles.

Before you begin

You must have administrator access to complete this task.

Procedure

To select the forum for which you want to control the access, complete the following steps:

1. Log in to the Developer Portal as an administrator.
2. From the administrator dashboard, select Structure.
3. Click Forums.

4. Select Edit forum for the forum you want to edit.
   The Edit forum page displays.

To edit the access to your selected forum, complete the following steps:

5. Ensure that the ACCESS CONTROL section is expanded, then navigate to the ROLES table.
   In this table, you can see the user roles available in the Developer Portal.
6. From this table, you can use the check boxes in the row corresponding to each user role to assign the following permissions to each user:
   - VIEW THIS FORUM
   - POST IN THIS FORUM
   - EDIT POSTS
   - DELETE POSTS
7. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Click edit for the forum topic that you want to lock.
3. Click Comments settings. The view expands.
4. Click the Closed radio button.
5. Click Save.
   You have disallowed any further replies and marked the topic as locked.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Select the check boxes next to the forum topics for which you want to make the content sticky.

3. From the UPDATE OPTIONS drop-down menu, select Make selected content sticky.
4. Click Update.
   You have marked the selected forum content as sticky.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Click delete for the forum topic that you want to remove.
   You have removed the selected forum post.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Content Author actions in the Developer Portal

You can control multiple content elements in the Developer Portal including the customization and restriction of specific elements.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.
- **Creating custom pages**
  You can create custom pages within the Developer Portal.

- **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.
- V5.0.5+ **Adding custom pages to APIs and Products**
  After you create a custom page in the Developer Portal, you can add them 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.
- **Adding new static content pages**
  You can add basic pages for static content in 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 have them displayed as documentation attachments. You can attach multiple files to APIs.
- **Cloning a static document node to create a new node**
  You can clone single document pages and recreate them in the Developer Portal. This allows you to clone and modify documents for large amounts of APIs from one document template.
- **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.
- **Creating a popular Product block**
  You can display any popular Products on the front page of your Developer Portal.
- **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 the tag hierarchy**
  You can manage the tag hierarchy in the Developer Portal.
- **Posting a poll**
  You can post a poll in the Developer Portal.
- **Uploading images for use in site content**
  You can upload .jpg, .png and .gif images in the Developer Portal for use in site content.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

The following entities have the meta tag form availability on their respective edit pages by default, and can be configured:

- Node
  - API
  - Application
  - Article

    - Basic page
    - Blog entry
    - FAQ
    - Forum topic
    - Poll
    - Product
    - Support ticket
- Taxonomy term
    - Forums
    - Tags
- User

Procedure

1. Click Modules in the administrator dashboard, then enter meta tag in the Filter list field.
2. From the list of meta tag modules that are enabled and disabled by default, select the meta tags that you want to enable or disable by turning them ON or OFF.
   The meta tag form that is available on the edit page of specified entities is comprised of the meta tag modules that are enabled.
3. After the functionality of the meta tag modules is configured to your specification, click Save configuration.
4. In the administrator dashboard, click Configuration > Metatag.
5. Click the Settings tab in the Metatag section, then expand the Master controls for all entities.
6. Select the check boxes for the entities that you want the meta tag form for. Conversely, clear the check boxes for the entities that you do not want the meta tag form for.
7. After you have configured the meta tag form availability for the specified entities, click Save configuration.
   You can configure the specified meta tag modules for the specified entities in the Developer Portal.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. On the administrator dashboard, click Configuration > Search and metadata > Metatag.
2. Click Override for the Global type.
3. Expand the Advanced tags section, and select the Prevents search engines from indexing this page check box.
4. Click Save.

Results

You have disabled external search engine indexing.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating custom pages

You can create custom pages within the Developer Portal.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

1. In the administrator dashboard, click Structure > Pages.
2. Click Add custom page.
3. Define the Basic settings:
    a. Enter a name for the page in the Administrative title field.
       The page is named, and a configurable machine name is created.
    b. Provide a description for the page in the Administrative description box.
    c. Enter the URL path that is required to get to the page in the Path field.
    d. Optional: Select the Make this your site home page check box if you want the panel to be your homepage.
    e. Optional: Select the Use this page in an admin overlay check box if you want to use the page in an admin overlay.
    f. Select the variant type from the Variant type list.
    g. Optional: Select any of the Optional features check boxes that satisfy your requirements.
    h. Click Continue.
4. Select the layout of page from the Category list, and then select the check box associated with the block formation that satisfies your layout specifications.
   For example, if you want to have your page split into two columns with two blocks above and below, you can select the Columns: 2 category, and the Two column stacked check box.
5. Optional: Define the Panel settings:
    a. Provide a title for the variant in the Administrative title field.
    b. Select the Disable Drupal blocks/regions check box if you want to disable all of the regions displayed in the theme.
    c. Define the CSS classes to remove from the body element of the page in the Remove body CSS classes field.
    d. Define the CSS classes to add to the body element of the page in the Add body CSS classes field.
    e. Enter a CSS ID for the page in the CSS ID field.
    f. Enter any additional custom CSS code to embed into the page in the CSS code field.
6. Click Continue.
7. Define the Panel content:
    a. Select the Title type from the list.
    b. Specify a title for the panel in the Title field.
    c. Specify the content of the blocks by clicking the Settings icon ⚙ for the appropriate block.
8. Click Finish, then Save.
   You have created a custom page within the Developer Portal.

What to do next

You can associate this page for a specific role. For more information, see Configurable role-based front pages.
You can clone a default, created, or custom page in the Developer Portal. For more information, see Cloning pages.

- **Cloning pages**
  You can clone a default, created, or custom page in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Cloning pages

You can clone a default, created, or custom page in the Developer Portal.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

1. On the administrator dashboard, click Structure > Pages.
2. In the Operations column, click Edit for the page that you want to clone.
3. In the Edit window, click the Clone tab.
4. Enter the name of the clone page in the Administrative title field.
5. Optional: Edit the machine name by clicking Edit adjacent to the Administrative title.
   You cannot change the machine name after you create the clone page.
6. Enter the URL path that is followed to get to the page in the Path field.
7. Click Clone.
   A page with summary information about your clone is displayed.
8. Configure your clone to your requirements.
   The specifications that are displayed by default are the specifications for the page that you cloned.
9. Click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. On the administrator dashboard, click Configuration > System > Social Block Twitter OAuth.
2. Using the information from your Twitter App, enter the Consumer key, Consumer secret, Access token, and Token secret in the corresponding fields.
3. 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 Developer Portal 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.

   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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.5 +

Adding custom pages to APIs and Products

After you create a custom page in the Developer Portal, you can add them 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

The custom pages that you can add to your APIs and Products can be additional information to further use and value of the APIs and Products.

For example, you can create a custom documentation page on the implementation of an API in a specific circumstance, and add the custom documentation page to the API.

Procedure

1. On the administrator dashboard, click Content, then click Add content.
2. Click Basic page.

3. Enter a title for you custom page in the Title text field.
4. Enter your information of interest in the Body text field.
5. If you want to add the custom page to specific Products, select the Products from the Link to specific Products list. If you want to add the custom page to all Products, select the Link to all Products check box.
6. If you want to add the custom page to specific APIs, select the APIs from the Link to specific APIs list. If you want to add the custom page to all APIs, select the Link to all APIs check box.
7. Optional: Configure the additional settings of the custom page to your specification.
8. Click Save.

Results

Your custom page is created and linked to any APIs or Products that you have specified. You can edit the page after it is created, and configure the APIs and Products that is linked to by clicking Edit for the custom page.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Procedure

To add new FAQs, complete the following steps:

1. Click Content in the administrator dashboard.
2. Click Add content.
3. Click FAQ.
4. Enter the question in the Question field. Provide further details of the question in the Detailed question field.
5. Enter the answer in the Answer field.
6. Click Save.
   You have added a new FAQ and related answer.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding new static content pages

You can add basic pages for static content in the Developer Portal.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

To add new pages, complete the following steps:

1. Click Content in the administrator dashboard.
2. Click Add content.

3. Click Basic page.
4. Enter a title in the Title field. Provide the main body content in the Body field.
5. Click Save.
   You have added a new basic page.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Click Edit next to the API or Product that you want to edit.
3. Select the Edit tab.
4. Under the Image heading, click Choose File button to select an image.
5. Click Upload, then click Save.
   You have added a new picture to the selected API or Product.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Attaching a documentation file to APIs

You can attach a file to APIs in the Developer Portal, and have them displayed 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 only attach a single 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. In the Developer Portal, click the API Products tab.
2. Click the API that you want to attach a file to.
3. Click the Edit tab.
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 check box 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 have uploaded multiple files, use the drag handle next to the file name and rearrange them.

Results

You have successfully attached a documentation file to your API.

- **Configuring file name transliteration**
  By default, all file names are transliterated, where they are converted to Latin characters. You can configure the behavior so that it is no longer default.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring file name transliteration

By default, all file names are transliterated, where they are converted to Latin characters. You can configure the behavior so that it is no longer default.

Procedure

1. On the administrator dashboard, click Configuration > Media > File System > Transliteration.
2. Select the Settings tab.
3. Ensure that the check box for Transliterate file names during upload in the Transliteration section is clear, then click Save configuration.

Results

Transliterating file names in the Developer Portal is no longer default behaviour.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Cloning a static document node to create a new node

You can clone single document pages and recreate them in the Developer Portal. This allows you to clone and modify documents for large amounts of APIs from one document template.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

1. Click Content in the administrator dashboard.
   A list of all the system content is displayed.
2. Under the OPERATIONS column, click clone for the static node you want to clone.
   A new page window opens with pre-filled content from the node you selected to clone. From this view you can edit the following:
   - The page title
   - The summary
   - The body
   - The format
   - The menu options to which the new page is assigned
3. Click Save.
   You are redirected to the content view, where your new page appears as a content node.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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:

- Click Content in the administrator dashboard.
- Click Add content.
- Click Blog post.
- Fill in the Title and Main body fields.
- Click Save. You have posted a new blog

To configure a blog, complete the following steps:

- Click Content in the administrator dashboard.
- Click edit next to the blog that you want to configure.
- You can configure the selected blog from this view.
- Click Save. You have configured the selected blog.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a popular Product block

You can display any popular Products on the front page of your Developer Portal.

Before you begin

You must have administrator access to complete this task.

You must have a product that is published to the Developer Portal.

Procedure

1. To create the Popular Product block in the Developer Portal:
   a. Click Content > Add Block > Popular Product.
   b. Insert a name in the Label field to display on admin interface.
   c. Optional: Enter a title for your block in the Displayed Title field.
   d. From the Product drop-down list, select the product that you want to display in the block.
   e. Click Save.
   You have created a popular Product block.
2. To display the popular Product block in the featured API pane of the Developer Portal:
   a. Click Structure, then click Pages.
   b. Alongside the page-welcome entry, click Edit.
   c. Click Layout, select the layout for your front page from the Category drop down list.
   d. Select the radio button that corresponds to the panel distribution that you require, then click Change, then Save.
   e. Click Content, then expand Featured APIs.
   f. From the bullet list, click Content.
   g. Click the Settings icon ⚙ for the featured API pane you want to add your popular Product block to, then click Add content.
   h. Click Blocks: Popular Product and select your popular Product.
   i. Optional: If you want to override the original title with a custom title, select the Override title check box.
   j. Click Finish.
   k. Click Update, then Save to save your changes.
   Your popular Product block is displayed on the front page of your Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. On the administrator dashboard, click Content.
2. Identify the API or Product page that you want to customize the URL alias for, and click Edit.
3. In the list that is located to the lower left of the window, click URL path settings.
4. In the URL alias text field, enter the custom URL alias that you want to assign your API or Product page, then click Save.
   Note: URL alias' must all be unique.

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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. 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 Insert/Edit Embedded Media icon 🎞 in the in the WYSIWYG editor to embed multimedia in the content.
5. Click Save.
   You have embedded multimedia in the selected site content.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing the tag hierarchy

You can manage the tag hierarchy in the Developer Portal.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

To manage the tag hierarchy, complete the following steps

1. Click Structure in the administrator dashboard.
2. Click Taxonomy.
3. Click list terms for the Tags vocabulary.
4. You can delete, rearrange and add new terms from this view.
5. Click Save.
   You have managed the tag hierarchy.

- **Tagging APIs based on their phase**
  In the Developer Portal, you can tag APIs with their Phase.

Related tasks

- Providing navigation by tag hierarchy
- Editing tags for a specific item
- Restricting the ability to create new tags in the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tagging APIs based on their phase

In the Developer Portal, you can tag APIs with their Phase.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

Tagging APIs with their Phase is disabled by default. To enable the function:

1. On the administrator dashboard, click Configuration > System > IBM API Connect.
2. Select the Automatically tag APIs with their phase check box.
3. Click Save configuration.

What to do next

You can manage the tag hierarchy of your APIs. For more information, see Managing the tag hierarchy.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Uploading images for use in site content

You can upload .jpg, .png and .gif images in the Developer Portal for use in site content.

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. Click Content in the administrator dashboard.
2. Open the desired content item by clicking its title.
3. Select the EDIT tab.
4. Under the Image heading, click Browse... to select the image you want to use.
5. Click Upload.
6. Click Save.
   You have added a new picture to the selected content type.

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 specify the types of content that you can add the Developer Portal. You can create and configure content types within the Developer Portal.
- **Configuring documentation upload limitations**
  You can attach documentation to an API by uploading it in the Developer Portal. You can configure the content type to place limitations on the uploaded documentation.
- **Configuring image upload limitations**
  You can configure the limitations that are placed on images that are uploaded in the Developer Portal. There is a default profile that can be configured with different image uploading limitations and you can assign it to different user roles within the Developer Portal.
- **Configuring tag count display**
  You can configure how the count for the number of tags that an API has assigned to it is displayed in the Developer Portal tag hierarchy navigation block.
- **Configuring the appearance of ratings in the Developer Portal**
  You can configure the appearance of ratings in the Developer Portal.
- **Configuring the social sharing widget in the Developer Portal**
  You can configure the social sharing widget in the Developer Portal.
- **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.
- **Restricting the ability to create new tags in the Developer Portal**
  You can restrict the ability to create new tags in the Developer Portal.

Creating content types

Content types specify the types of content that you can add the Developer Portal. You can create and configure content types within the Developer Portal.

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 entry

- FAQ
- Forum topic
- Product
- Poll
- Support ticket

The content type that you create, and the default content types, can be edited and configured further to your specifications.

Procedure

1. Click Structure > Content types.
2. Click +Add content type.
3. Enter a name for your content type in the Name field.
4. Optional: Enter the text that you want to appear on the Add new content page, in the Description field.
5. Enter a title field label for your content type in the Title field label field.
6. Optional: Configure any of the following options that are available for your content type:
   - Submission form settings
   - Publishing options
   - Display settings
   - Comment settings
   - Content approval
   - Fasttoggle settings
   - Multilingual settings
   - Menu settings
   - Notifications settings
7. After you have configured your content type to your specifications, click Save content type.
   You have created a new content type.

What to do next

You can add fields to your new content type by clicking manage fields in the "Content types" window.

- **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 cannot delete any of the existing fields for the following IBM® API Connect content types:

- API
- Product
- Application

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. Click Structure > Content types, and identify the content type that you want to add a field to.
2. Click manage fields for the content type.
3. Enter a label for your field in the Label field.
   The label that you specify for your field is displayed in the UI.
4. 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.
5. Specify the type of data that your field can store by selecting the type from the Field type list.
   You cannot change the field type after the field is created.
6. If possible, select the form element to edit the data from the Widget list.
   Field types have multiple display options that define how a field is displayed to users. You can change the widget and display options after the field is created.
7. Click Save.
   You have added a field to your content type.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring documentation upload limitations

You can attach documentation to an API by uploading it in the Developer Portal. You can configure the 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. In the administrator dashboard, click Structure > Content types.
2. For the API content type, click manage fields.
3. For the Documentation field, click edit.
4. Configure the type of files that are allowed to be uploaded in the Allowed file extensions field.
5. 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.
6. Optional: Configure any of the other options to satisfy your requirements.
7. Click Save settings.
   You have configured the documentation upload limitations.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring image upload limitations

You can configure the limitations that are placed on images that are uploaded in the Developer Portal. There is a default profile that can be configured with different image uploading limitations and you can assign it to different user roles within the Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

The configurable options that are available for profiles include:

- Maximum file size per upload
- Directory quota
- Total user quota
- Permitted file extensions
- Maximum image dimensions
- Maximum number of files per operation
- Directory paths
- Thumbnails

After you have configured the limitations for any profiles that you have created, you can assign them to roles on a public and private file level.
Note: To disable image upload completely, disable the IMCE module. For more information, see Disabling modules.

Procedure

1. Click Configuration, then click IMCE in the Media section.
   A Default profile is present.
2. You can either edit the Default profile with the file limitations that you require, or you can create a new profile:
   a. Click Add new profile
   b. Enter a name for your profile in the Profile name field.
   c. Configure your profile with the file limitations you require. After you have configured your profile, click Save configuration.
3. Optional: If you have already created other profiles, and want to import their settings, click on the appropriate profile that are listed next to Import settings from other profiles in Add new profile .
4. In the Role-profile assignments section, assign the relevant profile to the roles that are available in the Developer Portal.
5. Optional: If a user has multiple roles, you can assign weights to them.
6. Click Save configuration.
   You have assigned configured profiles to roles within the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring tag count display

You can configure how the count for the number of tags that an API has assigned to it is displayed in the Developer Portal tag hierarchy navigation block.

Before you begin

You must have administrator access to complete this task.

About this task

By default, when you tag an item in the Developer Portal with a tag that is part of a hierarchy, the count for that tag is saved at the deepest level that it is assigned. The tag hierarchy navigation block reflects that the entire lineage is not included in the count.

However, you can configure your Developer Portal to show the tag count across every level.

Procedure

1. In the administrator dashboard, click Structure > Content types.
2. Click manage fields for the API content type.
3. For the Tags field, click Hierarchial Select.
4. Select the Save term lineage radio button in the Save lineage section, then click Continue.
   You have configured the tag count display to save the count across all levels of the hierarchy, as opposed to the deepest level that the tag is assigned to.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 for the Teaser view found in the upper right of the Manage Display window:

1. Click Structure in the administrator dashboard.
2. Click Content types.
3. Click Manage display next to the required content type.
4. From the Format drop-down menu, select the style of the Rating field.
5. Optional: If you have selected the rating format to be displayed `As Stars`, the Settings icon ⚙ is visible . Click this icon to refine the display configuration further, then click Update.
6. Click Save. You have configured the appearance of the ratings feature for the required content type.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the social sharing widget in the Developer Portal

You can configure the social sharing widget in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To configure the social sharing widget in the Developer Portal, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the System heading, click AddToAny.
3. Click Placement to expand the Placement view.
4. Select the check boxes to display the social sharing widget for the required content types.
5. Click Save configuration. You have configured the social sharing widget feature for the required content types.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click edit under the comment to be edited.
2. Edit the content of the comment as required.
3. Click Save.
   You have edited the selected comment.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Click edit next to the content element you want to edit.
3. From this view, you can edit the sample content of the selected element as required.
4. Click Save.
   You have edited the sample content for the selected element.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Procedure

To edit tags for a specific item, complete the following steps:

1. Click Content in the administrator dashboard.
2. Click edit next to the content element you want to edit.
3. From the Tags drop-down menu, select <create new item>.
   The Tags window displays.
4. Enter a name for the new tag in the new field. Click Create.
5. Click Add, then click Save.
   You have created a new tag for an individual item.

Related tasks

- Providing navigation by tag hierarchy
- Restricting the ability to create new tags in the Developer Portal
- Managing the tag hierarchy

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restricting the ability to create new tags in the Developer Portal

You can restrict the ability to create new tags in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To restrict the ability to create new tags 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. The content type must have the Tags field enabled.
4. Click Hierarchical Select in the Tags row.
5. Under the Editability Settings heading, deselect the Allow creation of new terms check box.
6. Click Continue, followed by Save. You have restricted the ability to create new tags for the required content type.

Related tasks

- Providing navigation by tag hierarchy
- Managing the tag hierarchy
- Restricting the ability to create new tags in the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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:

- **Removing a site comment**
  You can remove site comments in the Developer Portal.
- **Removing blocks**
  You can remove blocks in the Developer Portal.
- **Turning blogs off in the Developer Portal**
  You can turn blogs off in the Developer Portal, this will prevent users posting and viewing blog entries.
- **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 forums off in the Developer Portal**
  You can turn forums off for your Developer Portal, this will prevent users from viewing, and contributing to, forums.
- **Turning ratings for specific content types on or off in the Developer Portal**
  You can turn the ratings for specific content types on or off in the Developer Portal.
- **Turning the ability to tag specific content types on or 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing a site comment

You can remove site comments in the Developer Portal.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

To remove a site comment, complete the following steps

1. Click delete under the comment to be deleted.
   The confirmation screen displays.
2. Click Delete.
   You have deleted the selected comment.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Removing blocks

You can remove blocks in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

1. Click Structure in the administrator dashboard.
2. Click Pages.
3. Find the welcome page row and select Edit from the drop-down list under the OPERATIONS column.
4. Click Content in the Summary pane.
5. Click the Settings icon for the block you want to remove.
6. Select Remove from the drop-down list.
7. Click Update and save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Turning blogs off in the Developer Portal

You can turn blogs off in the Developer Portal, this will prevent users posting and viewing blog entries.

Before you begin

You must have administrator or content author access to complete this task.

Procedure

To turn blogs off, complete the following steps:

1. Click Modules in the administrator dashboard.
2. Under the NAME heading, find Blog.
3. Use the ON/OFF slide control to turn blogs off.
4. Click Save configuration.
   You have turned blogs off.

What to do next

You can configure blogs using the Configuring blogs topic.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Note:
This procedure is for new content only, for example, newly published products.

For existing content, when only one product needs to be changed, navigate from Content, find the product, click Edit, then set the comments to be closed for that product, then click save. Or, if you have many products that need to be changed, you can delete the existing ones from Content on the portal and then trigger a background sync in order for them to be re-created with the new comment settings.

Procedure

To turn comments for 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 display next to the required content type.
4. Select the COMMENT DISPLAY tab.
5. From the Format drop-down menu for the Comment row, select <hidden> to turn off or Default to turn on.
6. Click Save. You have turned the comments feature for the required content type on or off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Content in the administrator dashboard.
2. Click edit next to the item you want to edit.
3. Click Comment settings in the side pane.
4. Select the Closed radio button to turn comments off for the selected item.
5. Click Save.
   You have turned off comments for the individual item.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Turning forums off in the Developer Portal

You can turn forums off for your Developer Portal, this will prevent users from viewing, and contributing to, forums.

Before you begin

You must have administrator access to complete this task.

Procedure

To turn off forums, complete the following steps:

1. Click Modules in the administrator dashboard.
2. Under the NAME heading, find Forum notifications and Forum.
3. Use the ON/OFF slider control to switch these elements off.
   Note: Forum Notifications must be turned off before you can turn Forums off, ensure that you turn Forum notifications off before attempting to turn Forums off.
4. Click Save configuration.
   You have turned forums off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Turning ratings for specific content types on or off in the Developer Portal

You can turn the ratings 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 ratings for 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 display next to the required content type.
4. From the Format drop-down menu for the Rating row, select <hidden> to turn off or As stars to turn on.
5. Click Save. You have turned the rating feature for the required content type on or off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Turning the ability to tag specific content types on or 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 display next to the required content type.
4. From the Format drop-down menu for the Tags field, select <hidden> to turn off, or any other option in the drop-down menu to turn on.
5. Click Save. You have turned the ability to tag specific content types on or off.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Administrator actions in the Developer Portal

- **Controlling the appearance of the Developer Portal**
  You can control and configure the appearance of the Developer Portal.
- **Extending functionality in the Developer Portal**
  You can extend the Developer Portal by creating and installing custom modules. There are recommendations that you can follow to help you create and implement modules consistently.
- **General configuration of the Developer Portal**
  You can control multiple aspects of the Developer Portal as an administrator.
- **Managing Developer Portal users**
  You can manage and customize users in the Developer Portal, by altering permissions and assigning roles to specific users.
- **Managing Developer Portal security**
  You can manage multiple security elements in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Controlling the appearance of the Developer Portal

You can control and configure the appearance of the Developer Portal.

To learn how to control the appearance of the Developer Portal, use the following topic links:

- **Controlling general appearance elements in the Developer Portal**
  You can control general appearance aspects of the Developer Portal.
- **Controlling the layout of the Developer Portal**
  You can control the layout 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.
- **Using 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 any required customizations.
- **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.
- **Installing additional themes**
  You can install additional themes in the Developer Portal.
- **Uninstalling themes**
  You can uninstall a theme from the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 appearance elements in the Developer Portal**
  You can change how the Developer Portal appears by changing appearance elements.
- **Toggling the display of appearance elements in the Developer Portal**
  You can turn appearance elements on or off in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing appearance elements in the Developer Portal

You can change how the Developer Portal appears by changing appearance elements.

To learn how to change appearance elements in the Developer Portal, use the following topic links:

- **Adding context to sections of the Developer Portal**
  You can add context to sections of the Developer Portal to manage contextual conditions and reactions.
- **Changing the shortcut icon**
  You can change the shortcut icon in the Developer Portal.
- **Changing the site email address**
  You can change the site email address in the Developer Portal.
- **Changing the site colors**
  You can change the site colors in the Developer Portal.
- **Changing the site logo**
  You can change the site logo in the Developer Portal.
- **Changing the site name**
  You can change the site name in the Developer Portal.
- **Changing the site slogan**
  You can change the site slogan in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding context to sections of the Developer Portal

You can add context to sections of the Developer Portal to manage contextual conditions and reactions.

Before you begin

You must have administrator access to complete this task.

About this task

When you add and configure a context, you can choose the conditions that trigger the context to activate and produce a specified reaction. Contexts can be grouped together with tags.

Procedure

1. In the administrator dashboard, click Modules.
2. In the Filter list field, enter Context.
3. Set the Context and Context UI modules to the ON position, then click Save configuration.
   Note: To improve performance, turn the Context UI module to the OFF position after you have completed the configuration of your contexts.
4. In administrator dashboard, click Structure > Context.
5. Click Add in the Context window.
6. Enter a unique ID for your context in the Name field.
7. Optional: If you want to create a group for your contexts, or already have a group created for your contexts, enter the name of the group in the Tag field.
8. Optional: Enter a description of the context definition in the Description field.
9. From the Conditions list, select a condition that will trigger the activation of the context.
   After a condition is selected, a description of the condition appears with the configurable options.
   Note: You can add multiple conditions for a context.
10. Configure the selected condition.
11. From the Reactions list, select a reaction that is produced after the context is activated.
    After a reaction is selected, a description of the reaction appears with the configurable options.
    Note: You can add multiple reactions for a context.
12. Configure the selected reaction.
13. Optional: Select the Require all conditions check box if you want the context to active only after all of the conditions are met.
    Note: If you do not select the Require all conditions check box, the context is activated after any specified condition within the context is met.
14. After you have specified your conditions and reactions, click Save.
    You have specified and configured the conditions and reactions of a context.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the shortcut icon

You can change the shortcut icon in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To change the shortcut icon, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM API Connect theme.
3. Click Shortcut Icon Settings to expand the view.
4. Deselect the Use the default shortcut icon check box.
5. Provide a path to an image on the server by entering it in the Path to custom icon field. Alternatively, you can browse for, and upload, an image under the Upload icon image subheading.
6. Click Save configuration.
   You have changed the shortcut icon for the site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Configuration in the administrator dashboard.
2. Under the System heading, click Site information..
3. Enter the new email address in the E-mail address field.
4. Click Save configuration.
   You have changed the email address for the site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the site colors

You can change the site colors in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To change the site colors, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM API Connect theme.
3. Under the COLOR SCHEME heading, select a preset color scheme from the Color set drop-down list. Alternatively, you can choose individual colors by choosing from the palette next to each appearance element.
4. Click Save configuration.
   You have chosen a color scheme for the site.

Note: You can modify all site colors at once using these alternative steps, as an advanced option:

5. Click Appearance in the administrator dashboard.
6. Click Settings for the IBM API theme.
7. Under the COLOR SCHEME heading, click in a color field for which you wish to set a color.
8. Use the color wheel to select a color for the content element in which you have clicked.
   The padlock symbols indicate that colors linked to the color you have chosen on the color wheel will be applied to all other elements with editable color in the Developer Portal.

You can change the color of elements individually without the color link feature:

9. Select the field of the content element for which you want to change the individual color, and type the hex value, in lowercase, manually. The linking feature will only be active when colors are selected from the color wheel.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the site logo

You can change the site logo in the 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. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM API Connect theme.
3. Click Logo Image Settings to expand the view.
4. Deselect the Use the default logo check box.
5. Provide a path to an image on the server by entering it in the Path to custom logo field. Alternatively, you can browse for, and upload, an image under the Upload logo image subheading.
6. Click Save configuration.
   You have chosen a logo for the site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the site name

You can change the site name in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To change the site name, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the System heading, click Site information..
3. Enter the new site name in the Site name field.
4. Click Save configuration.
   You have changed the name for the site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the site slogan

You can change the site slogan in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To change the site slogan, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the System heading, click Site information..
3. Enter a new slogan in the Slogan field.
4. Click Save configuration.
   You have changed the slogan for the site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Toggling the display of appearance elements in the Developer Portal

You can turn appearance elements on or off in the Developer Portal.

To learn how to toggle appearance elements in the Developer Portal, use the following topic links:

- **Toggling the display of the site logo**
  You can turn the display of the logo on or off in the Developer Portal.
- **Toggling the display of the site name**
  You can turn the display of the site name on or off in the Developer Portal.
- **Toggling the display of the site slogan**
  You can turn the display of the site slogan on or off in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Toggling the display of the site logo

You can turn the display of the logo on or off in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To toggle the appearance of the site logo, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM® API Connect theme.
3. Click TOGGLE DISPLAY to expand the view.
4. Select and deselect the Logo check box to toggle the appearance of the site logo.
5. Click Save configuration.
   You have toggled the appearance of the site logo.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Toggling the display of the site name

You can turn the display of the site name on or off in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To toggle the appearance of the site name, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM® API Connect theme.
3. Click TOGGLE DISPLAY to expand the view.
4. Select and deselect the Site name check box to toggle the appearance of the site name.
5. Click Save configuration.
   You have toggled the appearance of the site name.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Toggling the display of the site slogan

You can turn the display of the site slogan on or off in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To toggle the appearance of the site slogan, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM® API Connect theme.
3. Click TOGGLE DISPLAY to expand the view.
4. Select and deselect the Site slogan check box to toggle the appearance of the site slogan.
5. Click Save configuration.
   You have toggled the appearance of the site slogan.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Controlling the layout of the Developer Portal

You can control the layout of the Developer Portal.

To learn how to control the layout of the Developer Portal, use the following topic links:

- **Configuring the front page**
  You can configure the Developer Portal front page.
- **Adding a menu**
  You can add a new 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 image**
  You can change the banner image that is displayed on the Home page when a user first logs in to the Developer Portal.
- **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 layout of the Developer Portal for different client devices**
  You can change the layout of the Developer Portal for different devices that clients may use to access the Developer Portal.
- V5.0.6 + **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.
- **Providing navigation by tag hierarchy**
  You can provide the ability for users in the Developer Portal to navigate by using tag hierarchy.
- **Using PHP in a custom block**
  You can use PHP to customize your Developer Portal blocks. Using PHP to customize your blocks is an advanced option, and widens the customization capability of the Developer Portal.
- V5.0.8 + **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 popular Products block.

Procedure

To configure the front page, complete the following steps:

1. Click Structure, then click Pages.
2. Alongside the page-welcome entry, click Edit.
3. Use the options provided to configure the front page.
4. Click Update and save to save your changes.

What to do next

You can create a custom page and specify role-based front pages. For more information, see Creating custom pages and Configurable role-based front pages.

- **Configurable role-based front pages**
  In the Developer Portal, you can configure the front page according to the role of the user. Unauthenticated users can see a different front page to authenticated users and users with different roles can see different front pages according to their role.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configurable role-based front pages

In the Developer Portal, you can configure the front page according to the role of the user. Unauthenticated users can see a different front page to authenticated users and users with different roles can see different front pages according to their role.

Before you begin

You must have administrator access to complete this task.

About this task

If you want to have different front pages for different users, you can create the page and then apply the custom URL alias that is provided for that page, to the appropriate role's front page settings. For more information on creating the page, see Creating custom pages.

Procedure

1. Log in to the Developer Portal as an administrator.
2. Click Configuration in the administrator dashboard.
3. Under the FRONT PAGE heading, click Settings.
4. Under the ROLES heading, configure the front pages for the following roles by expanding the role you want to work with:
   - FORUM MODERATOR
   - CONTENT AUTHOR
   - ADMINISTRATOR
   - AUTHENTICATED USER
   - ANONYMOUS USER

   For each of these roles, you can use the radio buttons to select one of the following front page modes:
   Note: when a configurable mode is selected, the necessary inputs to configure the mode displays under the list of modes.
   - Skip - the front page will not be customized for this role.
   - Themed - your default theme will apply to the front page for this role.
   - Full - you can fully customize the front page for this user, including the layout of the page.
   - Redirect - users in this role will be redirected to a specified page.
   - Alias - users in this role see the path that they typed, displayed as a front page.
   -
5. Click Save Settings.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Adding a menu

You can add a new 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. Click Structure, then click Menus.
2. Click Add menu, provide a title, and an optional description, and click Save.
3. To add items to your menu, click Add link.
4. Click Save configuration to save your changes.

What to do next

A new block for your menu is created automatically. You can use block configuration options to control which pages your menu is displayed on, and where it is positioned. For more information, see Changing the blocks displayed on Developer Portal pages.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

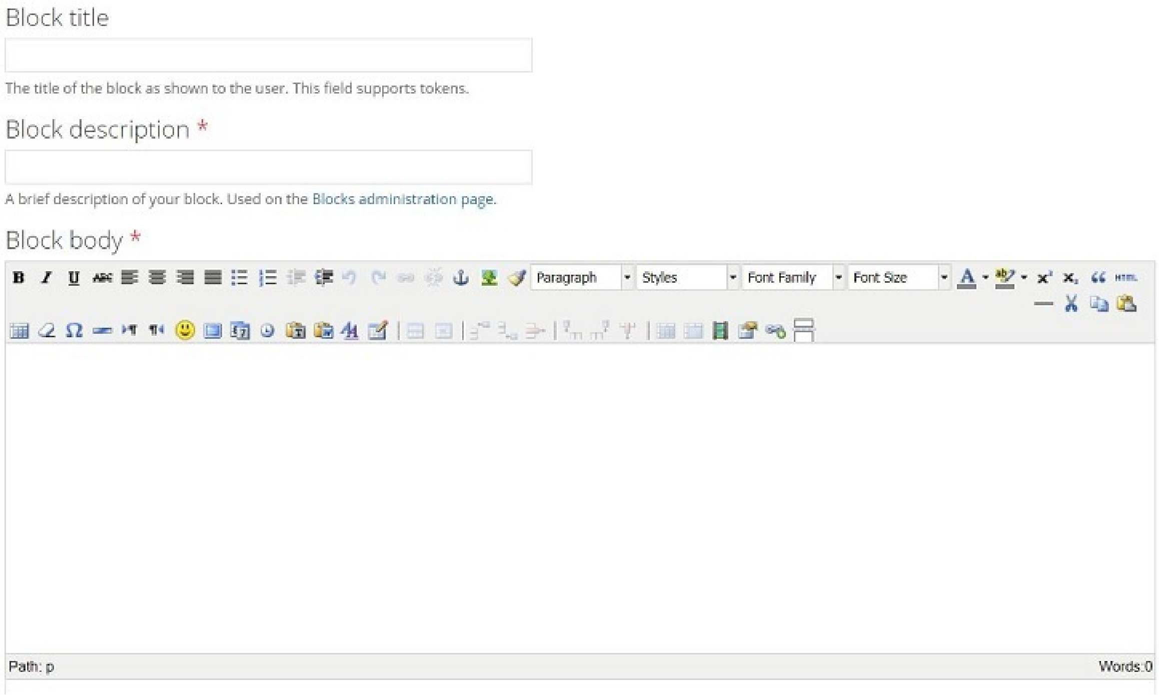
About this task

Blocks are boxes of content that are rendered into an area, or region, of a page. The content can be custom HTML 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.

Procedure

To add or modify blocks, complete the following steps as required.

1. Click Structure > Blocks.
   All of the blocks that are in the system are displayed by theme. From the Blocks administration 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.
2. To add a block, complete the following steps:
   a. Click Add block.
      The Blocks create page is displayed.
   b. Complete the Block title and Block description fields.
   c. Enter the content that you require for your block into the Block body.
      You can write your content in Full HTML, Filtered HTML, or Plain text. You can use the edit icons at the top of the Block body field to configure your content. Click the Edit HTML Source icon HTML to enter HTML content.


Block title

The title of the block as shown to the user. This field supports tokens.

Block description *

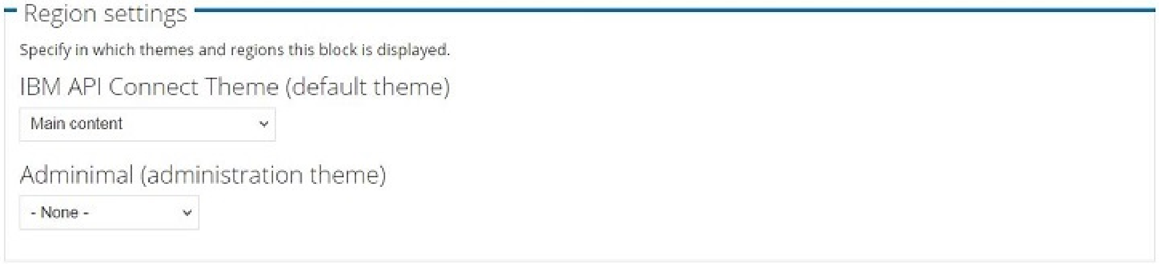
A brief description of your block. Used on the Blocks administration page.

Block body *

Path: p

Words:0

d. In the Region settings section, specify in which themes and regions the block is displayed.


Region settings

Specify in which themes and regions this block is displayed.

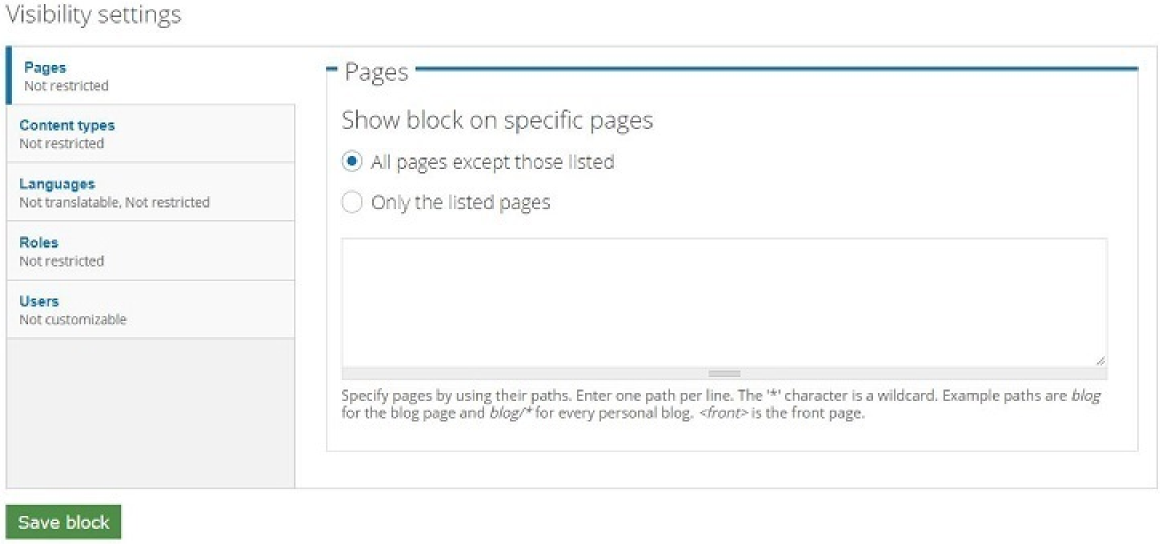
IBM API Connect Theme (default theme)

Main content

Adminimal (administration theme)

- None -

e. In the Visibility settings section, specify how you want the block to be displayed.
You can specify which pages the block is displayed on, limit the block to show only on pages that display specific content types, configure the language settings, restrict which roles the block is visible to, and allow individual users to customize the visibility of the block.


Visibility settings

Pages
Not restricted

Content types
Not restricted

Languages
Not translatable, Not restricted

Roles
Not restricted

Users
Not customizable

Pages

Show block on specific pages

● All pages except those listed

○ Only the listed pages

Specify pages by using their paths. Enter one path per line. The '*' character is a wildcard. Example paths are *blog* for the blog page and *blog/** for every personal blog. *<front>* is the front page.

f. Click Save block.

3. To modify an existing block, from the Blocks administration page complete the following steps:
   a. Click configure alongside the required block to change the block title, the region settings, and the visibility settings.
   b. To change the region in which a block is positioned, select the required Region from the drop-down list next to the required block.
   c. To change the vertical sort-order of a block within a region, drag the block to the required position.
4. Click Save blocks to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the front page banner image

You can change the banner image that is displayed on the Home page when a user first logs in to the Developer Portal.

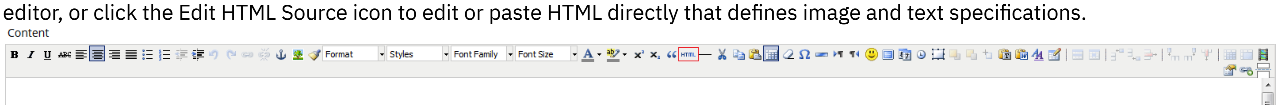
Before you begin

You must have administrator or content author access to complete this task.

Procedure

To change the front page banner image, complete the following steps:

1. Click Content, then select the BLOCKS tab.
2. Click edit in the Welcome Banner block.
3. Under the Content heading, to change the Content text and image for the Welcome banner, you can either enter text into the content editor, or click the Edit HTML Source icon to edit or paste HTML directly that defines image and text specifications.



4. Under the Image heading, browse for the required image, and click Update.
5. Click Save block to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the items in the main menu

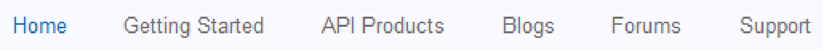
You can change the items in the main menu that is displayed on all pages in the Developer Portal.

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. Click Structure, then click Menus.
2. Alongside the Main menu entry, click list links.
3. Use the options provided to change the menu items.
4. Click Save configuration to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Changing the layout of the Developer Portal for different client devices

You can change the layout of the Developer Portal for different devices that clients may use to access the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To change the layout of the Developer Portal for different client devices, complete the following steps:

1. Click Appearance in the administrator dashboard.
2. Click Settings for the IBM APIM theme.
3. Configure the column layouts according to device.
4. Click Save configuration. You have changed the site layout for different client devices.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

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.

By default, all taxonomies must be created in the Developer Portal. For more information, see Managing the tag hierarchy.

After you have created 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.
Note: Any categories in a YAML file that do not exist in the Developer Portal taxonomy are automatically ignored.
However, if you have created categories for your APIs and Products in the API Designer or API Manager UI, you can have your APIs and Products displayed in those categories in the Developer Portal. For information on how to create categories for your APIs and Products in the API Designer or API Manager UI, see Organizing your APIs and Products into categories. For information on how to have your APIs and Products displayed in those categories in the Developer Portal, see Enabling dynamic category creation.

You can add additional Developer Portal taxonomies to the categories that are defined in the API Designer or API Manager UI. If you remove a taxonomy from a category, it is added back when the background sync next runs.

- V5.0.6 + **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 Designer or API Manager UI.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

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 Designer 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 in the API Designer 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 to tagging in the Developer Portal. For more information on tagging, see Providing navigation by tag hierarchy.
The categories that are defined in the API Designer 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 force the Developer Portal to update and display the change that you made in step 1:
   a. On the administrator dashboard, click Configuration > System > Cron.
   b. From the Operations drop-down list for the IBM API Connect module, select Run.
   The following message appears when your operation is complete:

```
ibm_apim_cron: Launched manually
```

Results

You have enabled the Developer Portal to display the categories that you have defined for your APIs and Products. You can see the categories that you have 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Procedure

To provide navigation by tag hierarchy, complete the following steps:

1. Click Structure, then click Blocks.
   All of the blocks in the system are displayed.
2. The Tags block provides navigation by tag hierarchy. You can complete the following configuration tasks for the Tags block:
   a. To change the region in the Tags block is positioned, select the region in the drop down list alongside the Tags entry.
   b. To change the block title, the region settings, and the visibility settings, click configure alongside the Tags entry.
3. Click Save blocks to save your changes.

Related tasks

- Editing tags for a specific item
- Restricting the ability to create new tags in the Developer Portal
- Managing the tag hierarchy

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using PHP in a custom block

You can use PHP to customize your Developer Portal blocks. Using PHP to customize your blocks is an advanced option, and widens the customization capability of the Developer Portal.

Before you begin

You must have administrator access to complete this task.

You must also have prior PHP experience.

About this task

CAUTION:
Allowing the use of PHP can affect the security of your Developer Portal site.

Procedure

1. Allow embedded PHP code to be evaluated in the Developer Portal:
   a. Click Modules in the administrator dashboard.
   b. Switch the PHP filter to ON.
   c. Click Save configuration.
2. Configure the Developer Portal to use PHP code:
   a. Click Configuration in the administrator dashboard.
   b. Under the CONTENT AUTHORING heading, click Text formats.
   c. Click configure under the OPERATIONS heading for the PHP code row.
   d. Under the Roles heading, select the administrator check box. If you enable PHP code for a role, anyone in that role can execute PHP code. Do not enable this option for roles in which PHP code is not used to edit and configure blocks.
   e. Click Save configuration.
3. Remove the WYSIWYG editor capabilities for PHP code:
   Note: The WYSIWIG editor is not compatible with PHP code, and must be switched off for PHP code.
   a. Select Configuration in the administrator dashboard.
   b. Under the CONTENT AUTHORING heading, click Wysiwyg profiles.
   c. Click Edit for PHP code.
   d. Expand the BASIC SETUP section.
   e. Deselect the Enabled by default check box.
   f. Click Save.
4. Create a custom block for your PHP code:
   a. Select Structure in the administrator dashboard.
   b. Click Blocks.
   c. Click Add block.
   d. Assign a name and description to your new custom block.
   e. Under the Block body heading, use the Text format drop-down list to select PHP code.
   f. Optional: You can enter the PHP code manually by clicking Disable rich-text under the Block body heading. PHP tags should be used, as in the following example:

      ```
      <?php print "myCustomBlockCode"; ?>
      ```

   g. Click Save block.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

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. Enabling the Views submodule, Views UI, provides a GUI to help create queries and provide options on how to display the data. After you define your views, the Views UI module can be disabled to improve the efficiency of your site.

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. You can also follow a tutorial about creating a custom sort order view for a list of APIs; see Tutorial: Configuring a custom sort order view for APIs in the Developer Portal.

- V5.0.8 + **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 UI module.

Related information

- Introduction to Views in Drupal 7
- Views overview

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

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 UI 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. Enable the Views UI (views_ui) module.

   a. On the administrator dashboard, click Modules.
   b. Enter `views` into the search field of the Modules pane to find the Views UI module, and set the module to ON.
   c. Click Save configuration.
2. Create a new view.
   a. On the administrator dashboard, click Structure > Add new view.
   b. Complete the View name and, optionally, the Description.
   c. Select what you want your view to Show, and optionally complete the of type, tagged with, and sorted by fields.
      For example, to create a view of APIs that is sorted by title, select Show `Content`, of type `API`, and sorted by `Title`.

   d. Select whether to create a page, 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.

   e. Complete the options that you require, 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.
   f. Click Save & exit if you have finished creating your view, or click Continue & edit > Save if you want to refine the view further.
      When saved, your new view is visible in the Structure > Views pane.
3. Optional: Refine your view further by clicking Structure > Views and selecting Edit against your newly created view.
   The details pane of your view is displayed, where 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 details pane also includes a preview area, so you can check your view before publishing.

   Click Save to make your changes permanent.

Results

You enabled the Views UI module and created a new view in the Developer Portal.

Related tasks

- Tutorial: Configuring a custom sort order view for APIs in the Developer Portal

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. In your custom theme, locate and open the .info file.
2. Place your custom JavaScript into a new sub-directory in your theme. For example, the sub-directory can be named js.
3. After you have placed your custom JavaScript into a sub-directory, add your JavaScript customizations to the script's array:

   ```
   scripts[] = js/example.js
   ```

   scripts[] is the special Drupal array of JavaScript files that is used to load as part of a theme, and *example.js* is your custom JavaScript.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Before you begin

You must have administrator access to complete this task.

You must also obtain a custom theme and a copy of the existing template. To obtain a copy of the existing template:

- If you have access to your Developer Portal node, you can obtain a copy of the template by following the path: /var/aegir/platforms/{platformname}/sites/all/modules/ibm_apim/templates.
- If you do not have access to your Developer Portal node,you can download a copy of the template, here: https://github.com/ibm-apiconnect/devportal/tree/APIC_v5/ibm_apim/templates

About this task

After you have applied a modified content types template, the changes will take precedence over the default content types template.
Note: You cannot modify any files on the file system of the Developer Portal.

Procedure

1. Apply any modifications to the default template that you have 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 the custom theme and name it templates.
3. Add your modified template to templates.
   Note: The modified template file name must be identical to the name of the template file that it is overriding.
4. Install, enable and set the custom theme with the modified template as default in the Developer Portal. For more information, see Installing additional themes

Results

You have modified and applied a template to override the default content types in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using 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 any required customizations.

For example, in Tutorial: Customizing themes for the Developer Portal, you are provided with a starting sub-theme. You then customize the styling by overriding the .css file that is inherited from the parent theme.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Modules, ensure that Views UI is set to `ON`, and click Save configuration.
   Note: Having the Views UI in the `ON` state slows the performance in the Developer Portal. Ensure that it is set to `ON` only for the setup process.
2. Create a content type for your carousel.
   a. Click Structure, then click 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 content type, then click manage fields for your newly created content type.
   e. Delete the `Body` field.
   f. Label your field in the Add a new field section.
      For example, *slide image*.
   g. Optional: Label your machine name by clicking edit next to the label that is generated automatically.
   h. Select Image as your field type from the list.
   i. Ensure that the widget type is Image, then click Save.
   j. Click Browse to assign the image that you want as your default image for your slide show if no other images are found.
   k. Click Save field settings, then click Save settings.
3. 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 2.
   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 Browse to find an image for your slide show, click Upload to upload the image, then click Save.
   d. Click + Clone content to open a clone of the image you created. Replace the title and image of the clone with what you want the next slide to contain.
   e. Repeat the cloning and editing process until you are satisfied with the content your slides, and the number of them.
4. Create a view for your carousel.
   a. Click Structure, then click Views > Add new view.
   b. Label your view.
      For example, Slide show.
   c. Clear the Create a page check box, and select the Create a block check box.
   d. Select Slick carousel in the display format section.
   e. Enter the number of slides you have for your carousel in the Items per page field, then click Continue and edit.
   f. Click Title and enter `<none>` into the text field. Click Apply (all displays).
   g. In the Format section, click Settings for Slick Carousel, and then set Skin Main to Default. Click Apply (all displays).
   h. In the Fields section, click Add > Content: slide image > Apply (all displays).
   i. Clear the check box for Create a label, then click Apply (all displays).
   j. In the Fields section, click Content: Title > Remove.
   k. In the Filter Criteria section, click Content: Published > Remove.
   l. In the Filter Criteria section, click Add and select one part of the slide image content type that you created for your carousel in step 2. Click Apply (all displays), then click Apply (all displays) again.
      Setting this filter criteria ensures that when new content is published to the Developer Portal, the slide show continues to display correctly.
   m. Click Save.
5. Enable the carousel functionality on your Developer Portal home page.
   a. Click Configuration, then click Media > Slick carousel.
   b. For Default, click edit, then click the settings tab in the list on the left.
   c. Select Autoplay, then click Save.
6. 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 Panel section.
   c. To remove the welcome banner, click the Settings icon in the Welcome Banner section for the Top pane, and click Remove.
   d. For the Top pane, click Settings > Add content > Miscellaneous > View: *name_of_your_view*. Click Finish.
   e. Click Update and save.
7. Click Modules, and set Views UI to `OFF`. Click Save configuration.

Results

Your Developer Portal front page has a slide show carousel.

What to do next

The default color for the carousel arrows are orange. You can override the default color by inserting the following code into the overrides.css file in a custom theme:

```
/* carousel */
.slick-prev::before, .slick-next::before {
    color: #ffffff;
}
```

For more information on editing custom themes, see Tutorial: Customizing themes for the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

What to do next

As blocks are positioned on a per-theme basis, you should verify that the region settings for the blocks in the new theme are correct. Click Structure > Blocks in the administrator dashboard, and reset the region settings for the blocks as required. Click Save blocks to save your updates. For a guided example of installing a new theme, see Tutorial: Installing a theme in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Uninstalling themes

You can uninstall a theme from the Developer Portal

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 uninstall it and remove it from the Developer Portal. Uninstalling 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 theme that you want to uninstall, and ensure that it is disabled.
3. Click the Uninstall tab.
4. Select the check box for the theme that you want to uninstall, and click Uninstall. You have uninstalled the theme, and it will not be present in the List tab.

Results

You have uninstalled a disabled theme in the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Extending functionality in the Developer Portal

You can extend the Developer Portal by creating and installing custom modules. There are recommendations that you can follow to help you create and implement modules consistently.

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 had a module named `custom_module`, a `menu` method within it is labeled as `custom_module_menu()`.

PHP does not have a concept of private or public functions. A function name that begins with an underscore should be regarded as private. You are recommended not to design it to be invoked outside its own module, as it is subject to change at without notice.

Important: If you are defining URL paths in a `hook_menu`, you must consider any potential future implication of any implementation. For example, you are **not** permitted to use `ibm_apim/*`, because it might cause conflict with future developments in the IBM provided code.

- **Creating custom modules to extend functionality**
  You can create custom modules in the Developer Portal. Custom modules can be created to use the custom hooks in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating custom modules to extend functionality

You can create custom modules in the Developer Portal. Custom modules can be created to use the custom hooks in the Developer Portal.

For more information on creating custom modules, see the Drupal developer guide at https://www.drupal.org/documentation/.
Important: You are **not** permitted to include any IBM® API Connect modules within any custom modules that you create.
By creating custom modules, you can exploit the following custom hooks that are available in the Developer Portal:

```
hook_ibm_apim_application_create
hook_ibm_apim_application_delete
hook_ibm_apim_application_update
```

```
hook_ibm_apim_application_clientsecret_reset
hook_ibm_apim_application_clientid_reset
hook_ibm_apim_application_unsubscribe
hook_ibm_apim_application_subscribe
hook_ibm_apim_application_creds_update
hook_ibm_apim_application_creds_create
hook_ibm_apim_application_creds_delete
```

V5.0.5 + There are four `drupal_alter` hooks that can be used to modify the client ID and client secret values:

```
hook_ibm_apim_application_new_application_clientcreds_alter
hook_ibm_apim_application_reset_application_clientid_alter
hook_ibm_apim_application_reset_application_secret_alter
hook_ibm_apim_application_add_application_alter
```

You can use the hooks to integrate with a third party system. For example, the hooks can allow an external server to generate the client IDs and client secrets:

```
module_name_ibm_apim_application_add_application_alter($appId, &$data){
   $newdata = make_rest_call_to_external_server($appId);
   $data['clientID'] = $newdata['clientid'];
   $data['clientSecret'] = $newdata['clientsecret'];
}
module_name_ibm_apim_application_add_application_alter(&$data){
   // invoke external server to get new credentials
   $newdata = make_rest_call_to_external_server();
   // set $data to the new credentials
   $data['clientID'] = $newdata['clientid'];
   $data['clientSecret'] = $newdata['clientsecret'];
}
```

where *module_name* is the name of your custom module.
V5.0.6 +

```
hook_ibm_apim_api_create
hook_ibm_apim_api_update
hook_ibm_apim_product_create
hook_ibm_apim_product_update
hook_ibm_apim_devorg_create
hook_ibm_apim_devorg_update
```

V5.0.7 +

```
hook_ibm_apim_subscription_create
hook_ibm_apim_subscription_update
hook_ibm_apim_subscription_delete
```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

General configuration of the Developer Portal

You can control multiple aspects of the Developer Portal as an administrator.

To learn how to configure specific aspects of the Developer Portal, use the following topic links:

- V5.0.1 + **Configuring workbench moderation**
  By configuring workbench moderation, you can use a dashboard to manage the review and approval process for content types in the Developer Portal. You can specify which roles can access the workbench dashboard by assigning them the appropriate permissions.
- **Applying rules in the Developer Portal**
  You can create rules in the Developer Portal which automatically trigger actions in response to situations or other actions.
- V5.0.4 + **Checking links in the Developer Portal**
  You can periodically check for any broken links in your 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 cron to run scheduled tasks**
  You can configure cron to run scheduled tasks in the Developer Portal.

- **Configuring the administrator dashboard**
  You configure the administrator dashboard in the Developer Portal.
- **Configuring the date and time**
  You can configure the date and time in the Developer Portal.
- **Configuring the site default time zone**
  You can configure the site default time zone in the Developer Portal.
- V5.0.2+ **Configuring the site error handling**
  You can configure the site error handling in the Developer Portal.
- **Configuring the support ticket system**
  You can configure the support ticket system in the Developer Portal.
- **Configuring translation of the taxonomy**
  You can configure the translation of your taxonomy in the Developer Portal, so that you can create terms in the default language, but give them different labels for different languages.
- **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.
- **Disabling modules**
  You can disable an entire module in the Developer Portal if you want to improve performance, or remove functionality.
- V5.0.3+ **Enabling code languages for code snippets**
  You can specify the languages that code snippets for APIs can be displayed in.
- V5.0.3+ **Enabling code snippets for SOAP APIs**
  By default, code snippets are only shown for REST APIs. You can enable code snippets for SOAP APIs.
- **Enabling regression testing in the Developer Portal**
  By enabling a collection of modules, you can test basic functionality in the Developer Portal.
- **Installing additional modules**
  You can install additional modules in the Developer Portal.
- **Search Autocomplete in the Developer Portal**
  The Search Autocomplete feature in the Developer Portal optimizes the search experience when looking-up content in the Developer Portal. The Search Autocomplete feature is also customizable for Developer Portal administrators.
- **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 an administrator is able to access the site, and 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Configuring workbench moderation

By configuring workbench moderation, you can use a dashboard to manage the review and approval process for content types in the Developer Portal. You can specify which roles can access the workbench dashboard by assigning them the appropriate permissions.

Before you begin

You must have administrator access to complete this task.

About this task

By using workbench moderation, you can configure the workflow of your Developer Portal site, and you can moderate content types, such as APIs and Products, with the workbench dashboard. However, you cannot moderate Developer organizations or applications with the workbench dashboard.

Procedure

1. To enable the workbench moderation module:

    a. On the administrator dashboard, click Modules.
    b. In the Search text field, enter the following modules:
        - `Workbench`
        - V5.0.5+ `Drafty`

       The Workbench, V5.0.5+ Drafty, and Workbench Moderation modules appear.
    c. Select the toggle buttons for the Workbench, Drafty, and Workbench Moderation modules so that a green tick is displayed.
    d. Click Save configuration.
       Note: You must rebuild your content access permissions after completing this step.
2. To enable revisions and moderation for specific content types:
    a. On the administrator dashboard, click Structure > Content types.
       The Content types window is displayed.
    b. Click edit for the content type that you want to enable revisions and moderation for.
    c. In the list to the lower left of the window, click Publishing options and select the check boxes for Create new revision and Enable moderation of revisions.
       Important: You must ensure that the Published check box is not selected.
    d. Optional: Select the default moderation state of the content type from the Default moderation state drop-down list.
    e. Click Save content type.
    f. On the administrator dashboard, click Reports > Status report
    g. In the Node Access Permissions section, click Rebuild permissions, then click Rebuild permissions again.
3. To configure workbench states and transitions:
    a. On the administrator dashboard, click Configuration > Workbench > Workbench Moderation.
    b. Optional: Enter a name and description for your new desired state in the corresponding text fields for New state.
       Note: You must reconfigure Views after completing this step.
    c. Optional: Select the Delete check boxes adjacent to any existing states that you want to delete.
       Note: You must rebuild your views after completing this step.
    d. Select the Transitions tab.
    e. Optional: Enter a name for your new desired transition in the Transition Name text field.
    f. Optional: Select the two transition states for your new transition state from the two New transition drop-down lists.
    g. Optional: Select the Delete check boxes adjacent to any existing states that you want to delete.
    h. Click Save after you have completed your configurations.
4. To set up the permissions for a specific content type for the Content Author role:
    a. On the administrator dashboard, click People > Permissions > Roles.
    b. Click edit permissions for the Content Author role.
    c. Ensure that the following check boxes are selected:
        - View content revisions
        - View all unpublished content
        - View the moderation messages on a node
        - Use "My drafts" workbench tab
        - Use "Needs review" workbench tab
        - *Content_type_that_you_want_to_moderate*: Edit own content
        - *Content_type_that_you_want_to_moderate*: Create new content
    d. Click Save permissions.
5. To identify the permissions that specific roles require to access the workbench dashboard:
    a. On the administrator dashboard, click Configuration > Workbench > Workbench Moderation > Check permissions
    b. Select the user role that you want to investigate from the Drupal role drop-down list.
    c. Select the type of moderation task that you want the role to be enabled to perform from the Moderation task drop-down list.
    d. Select the check boxes that correspond to the content types that you want the user to perform moderation tasks on.
    e. Click Check permissions.
       A list of recommended actions are displayed at the top of the Workbench Moderation window. By following the recommended actions, you can assign the permissions that are required for a role to access the workbench dashboard.

- V5.0.1+ **Editing, reviewing, and publishing content with the workbench**
  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 workbench before you publish it in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.1+

Editing, reviewing, and publishing content with the workbench

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 workbench 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 workbench moderation.

About this task

In addition to editing content through the workbench, you can specify whether the content needs further review, or even publish the content if you have the permission.

Procedure

1. On the administrator dashboard, click My Workbench.
   The My Workbench window is displayed.
2. If you have the necessary permissions to edit your content type:
   a. Click My drafts.
   b. Click the Title of the content type that you want to edit, then click the Edit draft tab.
   c. Edit your content to your specification, then click Save.
   You are redirected to the View draft tab.
   d. After you have finished editing your content type, ensure that Needs Review is selected from the Set moderation state field, then click Apply.
3. If you have the necessary permissions to review your content type:
   a. On the administrator dashboard, click My Workbench.
   The My Workbench window is displayed.
   b. Click Needs review.
   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 draft, apply the changes to your content type, then click Save.
   You are redirected to the View draft tab.
4. If you have the necessary permissions to publish your content type:
   a. On the administrator dashboard, click My Workbench.
   The My Workbench window is displayed.
   b. Click Needs review.
   c. In the Set moderation state column, click Change to Published for the content type that you want to publish.

Results

You have published your content type, and the content type is removed from the Needs review tab.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Applying rules in the Developer Portal

You can create rules in the Developer Portal which automatically trigger actions in response to situations or other actions.

Before you begin

You must have administrator access to complete this task.

About this task

By creating rules, you can automate and anticipate responses to situations, which can help provide a more personalized and efficient user experience.

You can define the type of situation a rule is used in. Rules are comprised of the conditions and actions. A condition is a piece of criteria that the situation must satisfy to result in the rule activating. The action is the result of the rule activating.

For example, you can create a rule for account deletion. You might define the condition to activate the rule when an account is deleted from the Developer Portal. After the rule is activated, the action can be to remove account data and send a final email to email address that was associated with the account holder to confirm the account deletion.

Procedure

1. To enable the necessary modules to apply rules, complete the following steps:
    a. On the administrator dashboard, click Modules.
    b. Ensure that the following modules are enabled:
        - Rules
        - Rules Scheduler
        - Rules translation
        - Rules UI
    c. Click Save configuration.
2. On the administrator dashboard, click Configuration > Workflow > Rules.
3. To implement the basic configuration of your rule, complete the following steps:
    a. In the Rules window, click Add new rule.
    b. Enter a name for your rule in the Name text field.
    c. Optional: In the Tags text field, enter a tag that is associated to the rule that you are configuring.
    d. From the React on event drop-down list, select the type of situation that will trigger your rule.
    e. Click Save.
4. Optional: Rules are triggered whenever an event occurs. To limit the action of a Rule to occur when certain criteria is true, you can configure the conditions that trigger your rule by completing the following steps:
    a. In the Conditions section, click Add condition to add a specific condition that trigger the results of your rule.
    b. From the drop down list, select the type of condition to add.
    The type of condition that you select results in the specific configuration window for that condition to open.
    c. Depending on the condition that you have selected, complete the mandatory fields, and any optional fields that you think might improve how accurate your condition is.
    d. Click Save.
5. To configure the action that your rule takes after it is triggered, complete the following steps:
    a. In the Actions section, click Add action.
    b. From the drop-down list, select the action that you want to add.
    The type of action that you select results in the specific configuration window for that action to open.
    c. Depending on the action that you have selected, complete the mandatory fields, and any optional fields that think might improve how effective your action is.
    d. Click Save.
6. Click Save.
    Your rule is active and can respond to the situations that you have specified, with the action that you have configured.

- **Importing and exporting Rules**
  To forgo the process of creating and configuring a Rule, you can import the contents of a Rule that already exists. In addition to importing Rules, you can export the contents of Rules that exist.
- **List of Rules events**
  You can see the list of events that you can create and configure Rules for in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.6 +

Importing and exporting Rules

To forgo the process of creating and configuring a Rule, you can import the contents of a Rule that already exists. In addition to importing Rules, you can export the contents of Rules that exist.

Before you begin

You must have administrator access to complete this task.

To import a Rule, you must have a local copy of the contents of an existing Rule from the Developer Portal.

To export a Rule, you must have access to a Rule that exists in the Developer Portal.

About this task

By importing and exporting Rules, you can speed up the creation of Rules, and maintain local copies and versions of Rules.

V5.0.6 + For more information on the creation of various types of Rules, see Tutorials for configuring Rules in the Developer Portal.

Procedure

- To import a Rule:
  1. Select and copy the contents of the Rule that is locally stored on your machine.
  2. On the administrator dashboard, click Configuration > Workflow > Rules.
  3. Click Import rule, then paste the contents of the Rule that you copied.
  4. Click Import.
     You have imported a Rule into the Developer Portal.
- To export an existing Rule:
  1. On the administrator dashboard, click Configuration > Workflow > Rules.
  2. Click export adjacent to the Rule that you want to export.
  3. Copy and paste the content that is displayed into a text editor, and save the contents to your machine.
     You have exported the contents of a Rule.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List of Rules events

You can see the list of events that you can create and configure Rules for in the Developer Portal.

Application

For more information on creating and configuring Rules to trigger upon events, see Tutorials for configuring Rules in the Developer Portal.

- After creating a new application
- After deleting an application
- After deleting application credentials
- After resetting the application client ID
- After resetting the application client Secret
- After saving new application credentials
- After subscribing to a plan
- After unsubscribing from a plan
- After updating an existing application
- After updating existing application credentials

Comment

- A comment is viewed
- After deleting a comment
- After saving a new comment
- After updating an existing comment
- Before saving a comment

Developer organization

- After creating a new developer organization

Node

- After cloning a node
- After deleting content
- After saving new content
- After updating existing content
- Before saving content
- Content is viewed

Product

- After deprecating a product
- After publishing a new product
- After removing a product
- After replacing an existing product
- After re-staging a product
- After retiring a product
- After superseding an existing product
- After updating the visibility of a product

Support Client

- After deleting a support client
- After saving a new support client
- After updating an existing support client
- Before saving a support client
- Support client is viewed

System

- Cron maintenance tasks are performed
- Drupal is initializing
- System log entry is created

Taxonomy

- After deleting a term
- After deleting a vocabulary
- After saving a new term
- After saving a new vocabulary
- After updating an existing term
- After updating an existing vocabulary
- Before saving a taxonomy term
- Before saving a vocabulary

User

- After a user account has been deleted
- After saving a new user account
- After updating an existing user account
- Before saving a user account
- User account page is viewed
- User has logged in
- User has logged out

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.4 +

Checking links in the Developer Portal

You can periodically check for any broken links in your Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

1. On the administrator dashboard, click Configuration > Content authoring > Link checker.
2. In the Scan content types for links section, select the check boxes adjacent to the content types for which you want the links checked.
3. Select the type of links that must be checked, from the What types of links should be checked? drop-down list.
4. Click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the administrator dashboard

You configure the administrator dashboard in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To configure the administrator dashboard, complete the following steps:

1. Click Dashboard in the administrator dashboard.
2. Click + Customize dashboard.
3. By dragging and dropping elements, you can configure what shows on the administrator dashboard from this view.
4. Click Done.
   You have configured the administrator dashboard.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Configuration in the administrator dashboard.
2. Under the REGIONAL AND LANGUAGE heading, click Date and time.
3. From the FORMAT drop-down menus, configure the date and time.
4. Click Save configuration.
   You have configured the date and time.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the site default time zone

You can configure the site default time zone 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.
4. Click Save configuration.
   You have configured the site error handling.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the support ticket system

You can configure the support ticket system in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To configure the support ticket system, complete the following steps:

1. Click Support ticketing system in the administrator dashboard.
2. Click Settings.
3. You can configure elements of the support ticket system from this view.
4. Click Save configuration.
   You have configured the support ticketing system.

To enable additional support ticket functionality:

5. Click Modules in the administrator dashboard.
6. In the list of module categories on the left, click Support, then enable any of the following modules that you require by setting them to ON:
   - Support charting
   - Support Mail Commands
   - Support Overview
   - Support Project Management
   - Support Reference
7. After you have enabled the modules that you require, click Save configuration.
   You have enabled additional support ticket functionality.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring translation of the taxonomy

You can configure the translation of your taxonomy in the Developer Portal, so that you can create terms in the default language, but give them different labels for different languages.

Before you begin

You must have administrator access to complete this task.

About this task

Taxonomy enables you to connect, relate, and classify your Developer Portal content by using organizational keywords (also known as categories, tags, or metadata). The i18n_taxonomy module provides multiple options to translate taxonomy vocabularies and terms. This task shows you how to configure your taxonomy to use the localize option, which means that you create terms in the default language of your site and then set translations for these terms in the required languages.

Procedure

To configure your taxonomy translation, complete the following steps.

1. Enable the i18n_taxonomy module:
   a. Click Modules in the administrator dashboard.
   b. Enter `i18n_taxonomy` into the Search field to find the module.
   c. Set the module to ON and click Save configuration.
2. Set the translation option to Localize:
   a. Click Structure > Taxonomy > Tags in the administrator dashboard.
   b. In the Multilingual options section, set the Translation mode to Localize. Terms are common for all languages, but their name and description may be localized.
   Setting the translation mode to Localize is important, as it means that you can have a single taxonomy term with different labels for different languages.

   c. Click Save.
3. Translate your terms:
   a. Click the List view for the Tags vocabulary.
   b. Find the term that you want to translate, and click Edit.
   c. Select the Translate tab to display the Translate taxonomy term window.
   d. Configure the translations for the term in the languages that you require.

Results

You configured your taxonomy translation so that you can create terms in the default language of your site, but with different labels for different languages.

Related tasks

- Configuring the taxonomy menu block

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Configuration in the administrator dashboard.

2. Under the CONTENT AUTHORING heading, click Wysiwyg profiles.
3. Click Edit for the Full HTML profile.
4. Click BUTTONS AND PLUGINS. You can toggle the editor buttons from this view using the relevant check boxes.
5. Click Save.
   You have configured the editor buttons displayed in the WYSIWYG rich text editor.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 ENABLED heading, select the check boxes for the languages you want to be available.
4. Under the DEFAULT heading, select the radio button for the language you want to set as default.
5. To add a new language, click +Add language. Use the drop-down menu to select the language you want to add.
6. Click Save configuration.
   You have configured the languages that are available.
7. You can also provide your own translations, for strings of the website that are not translated, from the Translate interface view, under Configuration in the administrator dashboard.

Results

You have configured the available languages for the Developer Portal.
V5.0.7+ Note: From IBM® API Connect Version 5.0.7.2, multilingual API and Product documentation can be created by using an `x-ibm-languages` extension directly in the OpenAPI (Swagger 2.0) definition. For more information, see Using `x-ibm-languages` to create multilingual API and Product documentation.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

Procedure

1. Click Modules.
2. Either enter the name of the module that you want to disable in the Filter list field, or scroll through the list of modules.
3. After you have identified the module that you want to disable, set the module to OFF, then click Save configuration.

You have disabled the module. To enable the module, repeat the steps, but set the module to ON instead of OFF.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Enabling regression testing in the Developer Portal

By enabling a collection of modules, you can test basic functionality in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

You must be a registered user in API Manager to complete this task.

About this task

The regression tests use API Manager as they are run locally on your Developer Portal site
Important: Do not run the Developer Portal regression tests on a production site.

Procedure

1. On the administrator dashboard, click Modules.
2. Search for, and enable, the following modules:
   - simpletest
   - site_test
   - xautoload
   - drupal_helpers
   - site_test_helpers
3. Click Save configuration.
4. On the administrator dashboard, click Configuration > Development > Testing.
   A list of tests that are available to use is displayed.

Results

You can run any of the tests that are now available to you. After you run a test, you can monitor and analyze the output.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Installing additional modules

You can install additional modules in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

Note: The following modules are unsupported in the Developer Portal:

- `entity_legal`

To install additional modules, complete the following steps:

1. Click Modules in the administrator dashboard.
2. Click Install new modules.
3. You can enter a path to the module in the Install from a URL field. Alternatively, you can upload a module under the Upload a module or theme archive to install heading.
4. Click Install.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Search Autocomplete in the Developer Portal

The Search Autocomplete feature in the Developer Portal optimizes the search experience when looking-up content in the Developer Portal. The Search Autocomplete feature is also customizable for Developer Portal administrators.

Before you begin

You must have administrator access to complete this task.

Procedure

1. Log in to the Developer Portal as an administrator.
2. Click Configuration in the administrator dashboard.
3. Under the SEARCH AND METADATA heading, click Search Autocomplete settings.
4. From this view, you can configure the Search Autcomplete feature, click Save.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 an administrator is able to access the site, and 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.

Procedure

To put the site into maintenance mode, complete the following steps:

1. Click Configuration in the administrator dashboard.
2. Under the DEVELOPMENT heading, click Maintenance mode.
3. Select the Put site into maintenance mode check box.
4. Enter a site message or, alternatively, leave the default site maintenance text.
   This message is what will be seen by those who visit the site while it is in maintenance mode.
5. Click Save configuration.
   Your site is now in maintenance mode.

To take the site out of maintenance mode, complete the following steps:

6. Enter your site URL in a web browser by using the following format: `https://`*`my_site_url`*`/user/login`.
7. Log in to your site as the admin user.
8. Click Configuration in the administrator dashboard.
9. Under the DEVELOPMENT heading, click Maintenance mode.
10. Clear the Put site into maintenance mode check box.
    Your site is now available to users and out of maintenance mode.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 apply updates from this view, it is strongly recommended that you do so regularly.

What to do next

Use the following link to learn how to update the Drupal core and modules for the Developer Portal: Maintaining the Developer Portal.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing Developer Portal users

You can manage and customize users in the Developer Portal, by altering permissions and assigning roles to specific users.

- **Working with roles in the Developer Portal**
  You can create and customize roles in the Developer Portal.
- **Adding custom fields to user records**
  You can add custom fields to user records 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 Developer organizations through an access content list. However, in the Developer Portal you can assign permissions 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 customize the properties for public APIs, as well as for APIs that are not viewable by the administrator, in the Developer Portal by creating an internal developer organization that can view and access all APIs staged to that Catalog as part of a Product.
- **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.
- **Customizing the terms and conditions statement**
  You can define and customize a terms and conditions statement that your users must accept before they can register to use your Developer Portal.
- **Customizing the terms of use statement**
  You can define and customize a terms of use statement that sets out the terms and conditions for using your Developer Portal.
- **Customizing user registration**
  You can customize the options that are displayed to users when registering to use the Developer Portal.
- **Restricting access by IP address**
  You can restrict access to a user or role by IP address in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Working with roles in the Developer Portal

You can create and customize 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 roles to users in the Developer Portal:

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 configuration 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: The email address that is used to create an Administrator account for the Developer Portal must be unique. It is not possible to create an account with the email address that is associated with the Administrator account. Any attempts to create an account with the associated email address results in the new account not functioning correctly, and returning the following error message when attempting 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 new role**
  You can assign permissions to a new role in the Developer Portal.
- **Creating a new role**
  You can create a new role within the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 an Developer Portal site created. For more information, see Installing the Developer Portal.
You must have administrator access to complete this task.

About this task

When 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 web admin user. The admin user can be used to administer the CMS capabilities of the Developer Portal. For more information, see Configuring Catalogs and testing the Developer Portal.

However, you can create additional administrative roles for users from within the Developer Portal by assigning the role to users that have access to the Developer Portal.

Procedure

To create additional administrator users for the Developer Portal:

Follow the steps in Assigning people to a role, and select the Administrator role from the Add a role to the selected users subheading.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click People in the administrator dashboard.
2. Select the List tab.
3. Select the check boxes for the target users.
4. From the drop-down list under the Update Options heading, select a role from the Add a role to the selected users subheading.
5. Click Update.

Results

You have assigned the selected users to the required role.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Assigning permissions to a new role

You can assign permissions to a new role in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

To assign permissions for a role, complete the following steps:

1. Click People in the administrator dashboard.
2. Select the PERMISSIONS tab.
3. Click Permissions.
4. Assign a permission to a role by selecting its check box in the relevant column.
5. Click Save permissions.

You have assigned permissions to the target role

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 PERMISSIONS tab.
3. Click Roles.
4. Enter a name for the role in the value box and click Add role.
   You have created a new role in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

1. In the administrator dashboard, click Configuration > People > Account settings.
2. Click the MANAGE FIELDS tab in the upper right corner of the Account settings window.
3. Enter a label for your field in the Label field.
4. Optional: If you want to edit the machine name that is automatically created based on your label, click Edit and enter the new name.
5. Specify the type of data that your field can store by selecting the type from the Field type list.
6. If possible, select the form element to edit the data from the Widget list.
7. Click Save, then 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.

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.

About this task

You can use the Developer Portal UI to block and unblock specific users.
Note: You can use the instructions here to unblock users who are blocked only due to the Developer Portal login security settings; see Login security.
To unblock users who are blocked due to the Developer Portal flood control settings, run the `reset_locked_user` command; see reset_locked_user.

Procedure

To block or unblock a specific user, complete the following steps:

1. Click People in the administrator dashboard.
2. Select the LIST tab.
3. Select the check box next to the target user.
4. From the drop-down list under Update Options, select Block the selected users or Unblock the selected users.
5. Click Update.
   You have blocked or unblocked specific users from the Developer Portal.

What to do next

For more information about login security, see Login security.

Related tasks

- Flood control

Related information

- Useful commands for use with a running node

Controlling access to Developer Portal content

In IBM® API Connect, you can restrict access to content for Developer organizations through an access content list. However, in the Developer Portal you can assign permissions 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 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.

By default, Content Authors 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.

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 assign permissions that grant users access to all content, proceed with one of the following steps:

- Assign a user to the Content Author role. For more information, see Assigning people 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 new role, and Assigning people to a role.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creating a developer account to customize API properties

You can customize the properties for public APIs, as well as for APIs that are not viewable by the administrator, in the Developer Portal by creating an internal developer organization that can view and access all APIs staged to that Catalog as part of a Product.

Before you begin

You must have administrator or content author access to complete this task.

About this task

The following API properties can be customized by any user of the new developer organization with administrator or content author permissions in an organization:

- Image uploading
- Tag assignment
- File attachment
- Custom field editing

Procedure

To create a new developer organization within the Developer Portal, complete the following steps:

1. Click the arrow next to `your_user_name` from the Developer Portal home page.
2. Select Create organization from the drop-down menu.
3. In the Organization name field, type the name of the organization, then click Submit.

To assign the required permissions to the users of your organization to whom you want to permit to edit API icons, complete the following steps:

4. Log in to the Developer Portal as an administrator.
5. Click People on the administrator dashboard.
6. Select the check box for the user, or users, to whom you want to give new permissions.
7. Under the UPDATE OPTIONS heading, select the relevant roles from the drop-down list.
   For example, the Content Author role can edit images and content within the Developer Portal.
8. Click Update.

What to do next

You must ensure that a Product that is published is visible to the internal developer 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: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

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. Click Content, then in the Show only items where section, select Basic page in the type field, and click Filter.
2. Find the Privacy policy content item in the list, and click Edit.
3. 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.
4. Click Save to save your changes.

Results

The privacy policy statement is successfully updated.

Related tasks

- Customizing the terms of use statement
- Customizing the terms and conditions statement

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Customizing the terms and conditions statement

You can define and customize a terms and conditions statement that your users must accept before they can register to use your Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

Customize the terms and conditions statement to set out the terms that users must accept before they can use your site. You can also configure a separate terms of use statement. For more information, see Customizing the terms of use statement.

Procedure

To define or customize a terms and conditions statement, complete the following steps:

1. Click Configuration, then click Legal in the People section.
   The Legal dialog box is displayed.
2. You can complete the following tasks:
   a. Enter the text of the terms and conditions statement.
   b. Choose the display style.
   c. Define additional check boxes that the user must select before completing the registration.
   d. Enter an explanation of the changes that have been made to the terms and conditions since the last version.
   e. Update the Legal Configuration for the display of the terms and conditions.
3. To preview your terms and conditions statement, click Preview.
   A preview is displayed in the display pane of the Legal dialog box.
4. Click Save to save your changes.

Results

The terms and conditions statement is successfully updated.

Related tasks

- Customizing the terms of use statement
- Customizing the privacy policy statement

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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 for using your Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

Customize the terms of use statement to set out the terms and conditions for using your site. If you want to force users of your site to accept your terms and conditions before they can use your site, configure a terms and conditions statement. For more information, see Customizing the terms and conditions statement.

Procedure

To define or customize a terms of use statement, complete the following steps:

1. Click Content, then in the Show only items where section, select Basic page in the type field, and click Filter.
2. Find the Terms of use content item in the list, and click Edit.
3. 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.
4. To preview your terms of use statement, click Preview.
5. Click Save to save your changes.

Results

The terms of use statement is successfully updated.

Related tasks

- Customizing the terms and conditions statement
- Customizing the privacy policy statement

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Customizing user registration

You can customize the options that are displayed to users when registering to use the Developer Portal.

Procedure

To customize the user registration process, complete the following steps:

1. Click Configuration in the administrator dashboard, then click IBM API Connect.
   The IBM API Connect dialog box is displayed.
2. You can complete the following tasks:
   a. To control which fields are shown during the registration, select the corresponding check boxes in the USER REGISTRATION pane.
   b. To configure the display of terms and conditions during user registration, click the link provided; for more information, see Customizing the terms and conditions statement.
   c. To configure the use of CAPTCHAs during user login or registration, click the link provided; for more information, see Configuring CAPTCHAS.
3. Click Save configuration to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Restricting access by IP address

You can restrict access to a user or role 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. 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.
    - 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.
    - 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing Developer Portal security

You can manage multiple security elements in the Developer Portal.

- V5.0.2 + **Selecting the Portal Delegated User Registry**
  Select Portal Delegated User Registry in the API Manager UI, to improve the flexibility of user registry and account management in the Developer Portal.
- **Automatic refresh after log out**
  To ensure that confidential data is not left displayed on the screen, you can force the browser to refresh the page after you are logged out of the Developer Portal.
- **Configuring CAPTCHAS**
  You can add a CAPTCHA challenge to any of your Developer Portal pages.
- **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.

- V5.0.2 + **Configuring the timeout length for the password reset link**
  You can configure the timeout length for one-time links that are sent from the Developer Portal.
- **Configuring your site password policy**
  You can configure your site password policy for logging into the Developer Portal.
- **Disabling CORS warnings**
  You can disable CORS warning 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.
- **Flood control**
  You can configure the flood control settings for your Developer Portal at an IP and user level.
- **Login security**
  You can configure the login security for your Developer Portal at an IP and user level.
- **Managing blocked IP addresses**
  You can block specific IP addresses from accessing your Developer Portal site.
- **User name and password autocomplete**
  You can define whether browsers remember your user name and password. The option to turn the autocomplete functionality off is enabled by default.
- **Using security questions**
  You can enable security questions for when a user wants to reset their Developer Portal account password. There are default security questions already configured when you allow security questions in the Developer Portal, which you can remove, as well as adding your own security questions.
- **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 two-factor authentication**
  You can enable two-factor authentication for the Developer Portal. This feature adds an extra level of security for users attempting to access the Developer Portal.
- V5.0.8 + **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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Selecting the Portal Delegated User Registry

Select Portal Delegated User Registry in the API Manager UI, to improve the flexibility of user registry and account management in the Developer Portal.

When Portal Delegated User Registry is selected for a Catalog, the user management is delegated from the management server to the Developer Portal, and new user accounts can be created in the local Developer Portal database, also known as the local user registry. However, selecting Portal Delegated User Registry also means that the following additional user registration methods can be configured in the Developer Portal:

Third-party authentication provider credentials

Enabling third-party authentication provider credentials, such as Facebook and Google, reduces the number of authentication credentials that a user of the Developer Portal needs. For more information, see Using third-party authentication provider credentials to access the Developer Portal.

LDAP user registry

Configuring LDAP means that the Developer Portal can authenticate users against an existing LDAP user registry. For more information, see Configuring the Developer Portal to use an LDAP user registry. You can also configure a writable LDAP by extending the LDAP configuration, for more information see Tutorial: Configuring writable LDAP in the Developer Portal.

OpenID Connect

Enabling OpenID Connect means that the Developer Portal can authenticate users against Google account credentials by using the OpenID Connect protocol. For more information, see Using OpenID Connect with Google.
If you want to use a different OpenID Connect client than Google, you need to create a custom module to add the new OpenID Connect provider. See the Drupal documentation for information, The OpenID Connect module.

Note that use of the Generic OpenID Connect client is not supported.

Important:

- If the Portal Delegated User Registry is selected for a Catalog, the Developer Portal REST APIs cannot be used to gain access to the content in that Catalog, and portal analytics is disabled. This restriction is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. You also cannot enable two-factor authentication for the Developer Portal.
- The Portal Delegated User Registry (PDUR) feature is not available in IBM® API Connect Version 2018, as additional user security options are available on the Management server. For a simpler migration process from Version 5 to Version 2018 (when the tooling is available), it is recommended to not use PDUR.

Select the Portal Delegated User Registry in the API Manager UI, by completing the following steps:

1. Click Dashboard in the Navigation pane, then click the Catalog for which you want to enable the use of external authentication provider credentials.
2. Click Settings > Portal.
3. Select the IBM Developer Portal radio button to enable the Developer Portal site.
4. Enter the URL of your Developer Portal site.
5. In the User Registration and Invitation section, select Portal Delegated User Registry from the User Registry drop-down list. The *catalog_name* is the name of the Catalog that you are working in and applying the registry settings to.
6. Click Save.
7. 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. When the link is active and you have accessed it, you can change the password of this administrator account.

User management is now delegated to your Developer Portal, and user registration will take place in the local Developer Portal database (local user registry). For information about the further configuration options that are available when Portal Delegated User Registry is set, see the following topics.

- V5.0.2 + **Using third-party authentication provider credentials to access the Developer Portal**
  You can use the log in credentials that are used with third-party authentication providers, to access the Developer Portal. Using third-party authentication provider credentials reduces the number of authentication credentials that a user has.
- V5.0.2 + **Configuring the Developer Portal to use an LDAP user registry**
  Your Developer Portal site can manage users and user registries. This task shows you how to configure the Developer Portal to use an LDAP user registry.
- V5.0.2 + **Approving accounts in the Developer Portal**
  When using the Portal Delegated User Registry, you can specify whether new accounts require administrator approval.
- V5.0.2 + **Modifying the Developer Portal email templates**
  You can modify the templates that are used for the emails that are sent by the Developer Portal.
- V5.0.3 + **Using OpenID Connect with Google**
  By using OpenID Connect, you can log in to the Developer Portal with Google credentials.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Using third-party authentication provider credentials to access the Developer Portal

You can use the log in credentials that are used with third-party authentication providers, to access the Developer Portal. Using third-party authentication provider credentials reduces the number of authentication credentials that a user has.

Before you begin

- You must have administrator access to complete this task.
- You must be the Owner or Administrator of a Developer organization in the API Manager.
- Portal Delegated User Registry must be selected in the API Manager UI. For more information, see Portal Delegated User Registry.
- In order to use third-party authentication provider credentials, the HybridAuth module must be enabled. However, note that you cannot use administrator approval of new accounts with Hybrid Auth.

Important:

- If the Portal Delegated User Registry is selected for a Catalog, the Developer Portal REST APIs cannot be used to gain access to the content in that Catalog, and portal analytics is disabled. This restriction is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. You also cannot enable two-factor authentication for the Developer Portal.
- The Portal Delegated User Registry (PDUR) feature is not available in IBM® API Connect Version 2018, as additional user security options are available on the Management server. For a simpler migration process from Version 5 to Version 2018 (when the tooling is available), it is recommended to not use PDUR.

About this task

You can use your third-party authentication provider credentials instead of an LDAP or local user registry. You can specify the use of authentication credentials for the following third-party authentication providers:

- Facebook
- Google
- LinkedIn
- Windows Live
- Twitter
- GitHub
- Slack

Note: You can enable multiple third-party authentication providers at one time.

Procedure

In the Developer Portal:

1. Log in as an administrator.
2. Ensure that the HybridAuth module is enabled. For information about enabling and disabling modules, see Disabling modules.
   When you select Portal Delegated User Registry in the API Manager UI, the module is enabled automatically.
3. On the administrator dashboard, click Configuration > People > HybridAuth.
4. From the list of authentication providers that are displayed, click the check box for the authentication provider that has the authentication credentials that you want to use, then click Settings.
   The Application settings tab contains text fields that must be filled with specific values for the authentication provider. Information on obtaining these values can be found in the Obtaining the Application setting values topics for each authentication provider, which are listed at the end of this topic.
5. Fill in the required fields for the authentication provider.
6. Click Save configuration.

Results

The icon for the third-party authentication provider that you have specified is displayed at the log in screen for the Developer Portal. By clicking on the icon, you are directed to the third-party authentication provider and are requested to enter your authentication credentials. After you enter and submit the authentication credentials, you are redirected back to your Developer Portal site.

- V5.0.2 + **Obtaining the Application setting values for Facebook**
  To enable the use of authentication credentials for Facebook, you must specify the Application setting values for your Facebook account.
- V5.0.2 + **Obtaining the Application setting values for Google**
  To enable the use of authentication credentials for Google, you must specify the application setting values for your Google account.
- V5.0.2 + **Obtaining the Application setting values for Twitter**
  To enable the use of authentication credentials for Twitter, you must specify the Application setting values for your Twitter account.
- V5.0.2 + **Obtaining the Application setting values for LinkedIn**
  To enable the use of authentication credentials for LinkedIn, you must specify the Application setting values for your LinkedIn account.
- V5.0.2 + **Obtaining the Application setting values for Slack**
  To enable the use of authentication credentials for Slack, you must specify the Application setting values for your Slack account.
- V5.0.2 + **Obtaining the Application setting values for Windows Live**
  To enable the use of authentication credentials for Windows Live, you must specify the Application setting values for your Windows Live account.
- V5.0.2 + **Obtaining the Application setting values for GitHub**
  To enable the use of authentication credentials for GitHub, you must specify the Application setting values for your GitHub account.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for Facebook

To enable the use of authentication credentials for Facebook, you must specify the Application setting values for your Facebook account.

Note: You must have an account with Facebook to obtain the Application setting values.
You must enter your Application ID and Application consumer secret in the corresponding fields. You can obtain these values by completing the following steps:

1. Creating a new application at https://developers.facebook.com/apps. You can gain the details by clicking Add a New App, selecting Website as your platform, entering the values that satisfy your requirements, then clicking Create App ID.
2. In the Tell us about your website section, you must set the Site URL to `https://site_url.com/`, then click Next.
3. Click Skip Quick Start, then you can find the Application ID and Application consumer secret in the Settings tab of your Facebook App.
4. In the Basic section in the Settings tab, enter your App Domain value, which must be in the format of `example_app_domain_name.com`, as opposed to `www.example_app_domain_name.com`. Click Save Changes.
5. Click Add Product, then Facebook Login. In the Valid OAuth redirects URIs field, enter the Callback URL to `https://site_url.com/hybridauth/endpoint?hauth.done=Facebook`. Click Save Changes.
6. Click App Review, then select Yes in response to Make Appname public?. Click Confirm.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for Google

To enable the use of authentication credentials for Google, you must specify the application setting values for your Google account.

Note: You must have an account with Google to obtain the Application setting values.
Obtain your Client ID and Client Secret. The following instructions provide guidance on how to obtain a Client ID and Client Secret from Google, however, the actual steps may vary as the Google application is updated.

1. Create a new application at https://console.developers.google.com, and select website as the platform that you want to use.
2. Create a new project.
3. On the developer homepage, search for and enable the following APIs:
   - Google People API
   - Google+ API
4. Add credentials to your project, and create an OAuth 2.0 client ID.
5. Configure the consent screen that will be shown to your users.
6. Select Web application for the application type, and enter a name.
7. Set Authorized JavaScript Origins to `https://site_url.com`, and Authorized Redirect URIs to `https://site_url.com/hybridauth/endpoint?hauth.done=Google`. After you have created the OAuth credentials, the Client ID and Client Secret are displayed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for Twitter

To enable the use of authentication credentials for Twitter, you must specify the Application setting values for your Twitter account.

Note: You must have an account with Twitter to obtain the Application setting values.
Enter your consumer key and private key. You can get these by completing the following steps:

1. Create a new application at https://dev.twitter.com/apps. Click Create New App.
2. Enter values for the following fields:
    - Name
    - Description
    - Website (the URL of your Developer Portal site)
3. Set Callback URL to `https://site_url.com/hybridauth/endpoint?hauth.done=Twitter`.
4. Click Yes, I agree, then click Create your user application.
5. Click Keys and Access Tokens to display the consumer key and private key.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for LinkedIn

To enable the use of authentication credentials for LinkedIn, you must specify the Application setting values for your LinkedIn account.

Note: You must have an account with LinkedIn to obtain the Application setting values.
Enter your Client ID and Client secret. You can get these by completing the following steps:

1. Create a new application at https://www.linkedin.com/secure/developer. Enter values for the Company Name and Description text fields.
2. Add an image for the Application Logo.
   Note: The image must be exactly 80x80 pixels.
3. Enter values for the following fields:
    - Website URL
    - Business Email
    - Business Phone
4. Agree to the LinkedIn API Terms of Use, then click Submit.
5. In Default Application Permissions, ensure that r_basicprofile and r_emailaddress are selected.
6. Set the OAuth Redirect URL to `https://site_url.com/hybridauth/endpoint?hauth.done=LinkedIn`, and Integration URL to https://*site_url*.com.
7. Your Client ID and Client secret are displayed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for Slack

To enable the use of authentication credentials for Slack, you must specify the Application setting values for your Slack account.

Note: You must have an account with Slack to obtain the Application setting values. The account must be authorized to use its authentication credentials to access the Developer Portal site.
Enter your Client ID and Client secret. You can get these by completing the following steps:

1. Create a new application at https://api.slack.com/apps. Click Create App.
2. Enter a name for your App in the App name field, then select the team that your account belongs to from the Team drop-down list.
3. Enter a description of your App in the Short Description field, and set the Redirect URI(s) to `https://site_url.com/hybridauth/endpoint?hauth.done=Slack`.
4. Click Create App, and locate the Client ID and Client secret in the App Credentials section.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for Windows Live

To enable the use of authentication credentials for Windows Live, you must specify the Application setting values for your Windows Live account.

Note: You must have an account with Windows Live to obtain the Application setting values.
Enter your Client ID and Client secret. You can get these by completing the following steps:

1. Create a new application at https://account.live.com/developers/applications/create. Click New Application Registration, then enter a value in the Name field.
2. Click Create application.
3. Click Add Platform, then select Web.
4. Click Allow Implicit Flow, then set the Redirect URIs to https://*site_url*.com.
5. Click Save. Your Client ID and Client secret is displayed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Obtaining the Application setting values for GitHub

To enable the use of authentication credentials for GitHub, you must specify the Application setting values for your GitHub account.

Note: You must have an account with GitHub to obtain the Application setting values.
Enter your Client ID and Client secret. You can get these by completing the following steps:

1. Create a new application at https://github.com/settings/applications/new. Click Register new OAuth application and enter information for the following fields:
   - Application name
   - Homepage URL
2. Set Authorization Callback URL to `https://site_url.com/hybridauth/endpoint?hauth.done=GitHub`
3. Click Register application. Your Client ID and Client secret are displayed.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Configuring the Developer Portal to use an LDAP user registry

Your Developer Portal site can manage users and user registries. This task shows you how to configure the Developer Portal to use an LDAP user registry.

Before you begin

- You must have administrator access to complete this task.
- Portal Delegated User Registry must be selected in the API Manager UI. For more information, see Portal Delegated User Registry.
- You must have an LDAP server and its details.

Important: Review the following considerations before you configure your Developer Portal to use an LDAP user registry:

- You can set up LDAP as an authentication provider for your Developer Portal site by configuring an LDAP user registry in API Manager. This option offers greater security as your LDAP configuration is kept separate from the Developer Portal database. For more information, see Working with user registries.

- If you need to use writable LDAP, have multiple LDAP servers, or want to use account approval, then follow the instructions in this topic to set Portal Delegated User Registry, and configure an LDAP authentication provider within your Developer Portal.
- If Portal Delegated User Registry is selected for a Catalog, the Developer Portal REST APIs cannot be used to gain access to the content in that Catalog. This restriction is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. You also cannot enable two-factor authentication for the Developer Portal
- If you are using the Portal Delegated User Registry, you can't combine multiple LDAP servers to gain user access to the Portal. You also can't use a secure LDAP while you are using the Portal Delegated User Registry.

About this task

Configuring LDAP means that the Developer Portal can authenticate users against an existing LDAP user registry. LDAP can be configured in the Developer Portal in combination with other user registration methods, such as OpenID Connect, and can also be used with new user account approval. For a tutorial about configuring LDAP and third-party authentication providers, see Tutorial: Using the Portal Delegated User Registry.

Note: This task shows you how to configure the Developer Portal to authenticate users against an existing LDAP user registry. If you want to allow new users to be added to the LDAP user registry, you need to extend the LDAP configuration; see Tutorial: Configuring writable LDAP in the Developer Portal for more information.
Note: When you use the Portal Delegated User Registry, PDUR, you can't combine multiple LDAP servers to gain user access to the Portal. Also, you also can't use a secure LDAP when you are using the PDUR.

Procedure

To enable LDAP configuration in the Developer Portal:

1. Log in to your Developer Portal site as the administrator.
2. On the administrator dashboard, click Modules.
3. Search for, and enable, the following modules:
   - LDAP Servers
   - LDAP User Module
   - LDAP Authentication

   Then, click Save configuration.
4. Click Configuration > People > LDAP Configuration.
5. To configure your LDAP registry settings:
   a. Click Settings, then select the checkbox in the Require HTTPS on Credential Pages.
   b. Click Save configuration.
6. To configure your LDAP registry server:
   a. Click Servers > Add LDAP Server Configuration.
   b. In Connection settings, enter values for the following fields:
      - Machine name for this server configuration
      - Name

   c. Select the checkbox for Enabled.
   d. Select your type of LDAP server from the LDAP Server Type drop-down list.
   e. Enter the IP address or domain name of your LDAP server in the LDAP server text field.
   f. Enter you port number in the LDAP port text field.
   g. In the Binding Method section, select Anonymous Bind for search, then Bind with User Credentials.
   h. In the LDAP User to Drupal User Relationship section, enter values for the following fields:
      - Base DNs for LDAP users, groups, and other entries
      - AuthName attribute

   i. Optional: Enter values for the following fields:
      - AccountName attribute
      - Email attribute
      - Email template
      - Thumbnail attribute
      - Persistent and Unique User ID attribute
      - Expression for user DN. Required when "Bind with Users Credentials" method selected
      - Testing Drupal Username
      - DN of testing username

   j. Click Add.
7. To configure the LDAP Authentication:
   a. Click the Authentication tab, then in the LDAP Authentication Settings section, select the checkbox for your LDAP server that is found under Authentication LDAP Server Configurations.
   Note: If you are using any third-party authentication providers, or have local Developer Portal users, the Mixed mode radio button must be selected. In this way, if a user does not exist in the LDAP registry, their account is created in the local

Developer Portal database. If you want to use LDAP only, select the Only LDAP Authentication is allowed except for user 1 radio button.

b. In the User Login Interface section, enter values for the following fields:
   - Username Description Text
   - Password Description Text
   - LDAP Account User Help URL
   - LDAP Account User Help Link Text

c. In the Email section, select the checkbox for Don't show an email field on user forms, then click Save.

Results

The Developer Portal is now configured to authenticate users with an existing LDAP user registry.

Related tasks

- Tutorial: Configuring writable LDAP in the Developer Portal

Related information

- Tutorial: Using the Portal Delegated User Registry

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Approving accounts in the Developer Portal

When using the Portal Delegated User Registry, you can specify whether new accounts require administrator approval.

Before you begin

You must have administrator access to complete this task.

You must have enabled Portal Delegated User Registry in the API Manager UI to complete the procedure. For more information, see Portal Delegated User Registry.

About this task

You can change the account settings to enable account approval for all new accounts including external authentication providers, or you can change the account settings to enable account approval for only the external authentication providers.

Note:

- You can change the account settings for approving new accounts only if you are using the Portal Delegated User Registry.
- Administrator approval of new accounts doesn't work with all of the Drupal authentication modules, for example it doesn't work with the HybridAuth module. Therefore, you cannot use administrator approval with third-party authentication provider credentials. For more information, see the Drupal 7 documentation.

Procedure

- If you want to change the account settings to enable approval for all new accounts including external authentication providers:
  1. On the administrator dashboard, click Configuration > People > Account settings.
  2. In the Registration and cancellation section, select the check box that satisfies your account approval requirements:
     - Administrators only
     - Visitors
     - Visitors, but administrator approval is required (note that you cannot use this option with the email verification function described in Step 3 below)
  3. To enable the Require e-mail verification when a visitor creates an account function, select the adjacent check box.

    4. Click Save configuration.
- If you want to change the account settings to enable approval for only external authentication providers:
    1. On the administrator dashboard, click Configuration > People > HybridAuth.
    2. In the HybridAuth window, click Account settings.
    3. In the Account settings section, select the check box that satisfies your account approval requirements:
        - Follow core: Visitors
        - Visitors
        - Visitors, but administrator approval is required (note that you cannot use this option with the email verification function described in Step 4 below)
        - Nobody, only login for existing accounts is possible
    4. In the E-mail verification section, select the check box that satisfies your e-mail verification requirements:
        - Follow core: Don't require e-mail verification
        - Require e-mail verification
        - Don't require email verification
    5. Click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.2 +

Modifying the Developer Portal email templates

You can modify the templates that are used for the emails that are sent by the Developer Portal.

Before you begin

You must have administrator access to complete this task.

You must have enabled Portal Delegated User Registry in the API Manager UI to complete the procedure. For more information, see Portal Delegated User Registry.

About this task

Token are used in the templates, and are replaced with specified values when an email is sent.

Note: You can customize the template that is used for user activation or password reset only if you are using the Portal Delegated User Registry.
You can modify the templates for the following types of emails:

- Welcome (new user created by administrator)
- Welcome (awaiting approval)
- Welcome (no approval required)
- Account activation
- Account blocked
- Account cancellation confirmation
- Account canceled
- Password recovery

Procedure

1. On the administrator dashboard, click Configuration > People > Account settings.
2. In the E-mails section, select and modify the content of any of the available email templates.
3. After you have finished modifying any templates, click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.3 +

Using OpenID Connect with Google

By using OpenID Connect, you can log in to the Developer Portal with Google credentials.

Before you begin

- You must have administrator access to complete this task.
- Portal Delegated User Registry must be selected in the API Manager UI. For more information, see Portal Delegated User Registry.
- You must also have an account with Google.

Important:

- If the Portal Delegated User Registry is selected for a Catalog, the Developer Portal REST APIs cannot be used to gain access to the content in that Catalog, and portal analytics is disabled. This restriction is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. You also cannot enable two-factor authentication for the Developer Portal.
- The Portal Delegated User Registry (PDUR) feature is not available in IBM® API Connect Version 2018, as additional user security options are available on the Management server. For a simpler migration process from Version 5 to Version 2018 (when the tooling is available), it is recommended to not use PDUR.

About this task

The OpenID Connect module can be used along with Google credentials to enable quicker and easier access to the Developer Portal.

Note: Users can configure two-factor authentication (TFA) within their social provider credentials, such as Google, to add a further level of security. With TFA configured, users would log in to the Developer Portal site with a verification code in addition to their username and password.

Procedure

1. Obtain the necessary Client ID and Client secret from Google that is needed for OpenID Connect in the Developer Portal:
   a. Log in to the Google developers site, for example https://console.developers.google.com.
   b. Create a project, and then search for and enable the following APIs:
      - Identity toolkit API
      - Google+ API

      Enabling the Google+ API means that actual user names are displayed in the Developer Portal, rather than internal OpenID Connect strings.
   c. From within the Google+ API, create the Client ID credentials for a web application. In the Configure consent screen enter the Product name that is shown to the user. Set the Authorized Redirect URIs field to `https://site_url/openid-connect/google`.
      After you have created the Client ID credentials, the Client ID and Client secret are displayed. The Client ID and Client secret are required to enable OpenID Connect in the Developer Portal.
2. Enable the OpenID Connect module in the Developer Portal:
   a. Log in to the Developer Portal as the administrator.
   b. On the administrator dashboard, click Modules.
   c. In the Modules window, search for and enable the OpenID Connect module, then click Save configuration.
3. Enable OpenID Connect with Google:
   a. On the administrator dashboard, click Configuration > Web services > OpenID Connect.
   b. In the Enabled OpenID Connect clients section in the OpenID Connect window, select the Google check box.
   c. In the Google section, enter your Google Client ID and Client Secret in the corresponding fields.
   d. Click Save configuration.
      OpenID Connect with Google is enabled, and the Google icon appears on the account login window in the Developer Portal.

What to do next

On the login window in the Developer Portal, click the Google icon to allow the necessary permissions to enable the use of OpenID Connect.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Automatic refresh after log out

To ensure that confidential data is not left displayed on the screen, you can force the browser to refresh the page after you are logged out of the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Procedure

1. Click Configuration, then Ejector Seat settings in the People section.
2. Define how often a check happens to see whether you are logged out by selecting a time in the AJAX check rate list.
3. Select the radio button that defines whether you want the check to run in the background, in the Run AJAX check in background section.
   You have defined the rate and activity of the when the browser forces a refresh after you are logged out.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring CAPTCHAS

You can add a CAPTCHA challenge to any of your Developer Portal pages.

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.

Procedure

To configure CAPTCHAS, complete the following steps:

1. Click Configuration, then click CAPTCHA.
   The CAPTCHA dialog box is displayed.
2. 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.
   e. Select the IMAGE CAPTCHA tab to configure the character, font, and image settings.
3. Click Save configuration to save your changes.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

To change the blocks that are displayed on a page, complete the following steps:

1. Click Configuration, then click Auto Logout.
   The Auto Logout dialog box is displayed.
2. Use the session timeout configuration options provided as required. For example, you can complete the following tasks:
   Note: The default session timeout is 30 minutes.
   a. Specify the length of inactivity time, in seconds, before a user is automatically logged out.
   b. Specify session timeout values for individual user roles.
   c. Provide a redirection URL at logout.
   d. Provide session expiry message texts.
3. Click Save configuration to save your changes.
   You have specified session timeout settings for a user.

To restrict the number of sessions a user has available to them:

4. In the Configuration tab, click Session limit.
5. Specify the default maximum number of sessions for a user by entering the number into the Default maximum number of active sessions field.
6. Select the radio button that enforces the action that is taken in the When the session limit is exceeded section.
7. Select the message that the user receives when they are logged out from the Logged out message severity list.
   The default message is `Error`.
8. Define the logged out message the user receives in the Logged out message field, then click Save configuration.
   You have created session limit restrictions for a user.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring the timeout length for the password reset link

You can configure the timeout length for one-time links that are sent from the Developer Portal.

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 one-time links that are sent by the Developer Portal only.
Note:

- Setting the password reset timeout length in the Developer Portal affects only the admin account user account, and Portal Delegated User Registry (PDUR) user accounts. For more information about PDUR, see Selecting the Portal Delegated User Registry.
- 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. On the administrator dashboard, click Configuration > People > Account settings. The Account settings window is displayed.
2. In the Account settings window, navigate to the Password reset section.
3. Enter your specified value for the timeout in the User Password Reset Link Timeout field.

Note: The timeout value is specified in seconds, and is 86400 seconds by default.

4. Click Save configuration.
   The timeout length for the one-time links that are sent from the Developer Portal is configured.

Related concepts

- Selecting the Portal Delegated User Registry

Related information

- Changing the expiration settings for Cloud Manager user accounts

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuring your site password policy

You can configure your site password policy for logging into the Developer Portal.

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. On the administrator dashboard, click Configuration > People > Password policies.
   The Password policies dialog box is displayed.
2. Edit the password policy options provided as required. For example, you can complete the following tasks:
   a. Choose to display the password restrictions on the password change page.
   b. Change the default password policy by selecting the List tab, then clicking edit. To save your changes, click Save.
      The following examples are password policy options:
      - Control the password complexity based on character types.
      - Require a minimum number of characters of different types.
      - Define the minimum password length.
      - Specify that the password must not contain the user name.
      - Restrict digit placement.
3. Click Save configuration to save your password policy settings.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Disabling CORS warnings

You can disable CORS warning 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. Click Configuration > System > IBM API Connect.
2. Ensure that the Display CORS warnings for unenforced APIs check box is clear.
3. Click Save Configuration.
   You have disabled CORS warnings from 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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. Click Configuration > System > IBM API Connect.
2. Ensure that the Allow live testing of APIs check box is cleared.
3. 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 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Flood control

You can configure the flood control settings for your Developer Portal at an IP and user level.

Before you begin

You must have administrator access to complete this task.

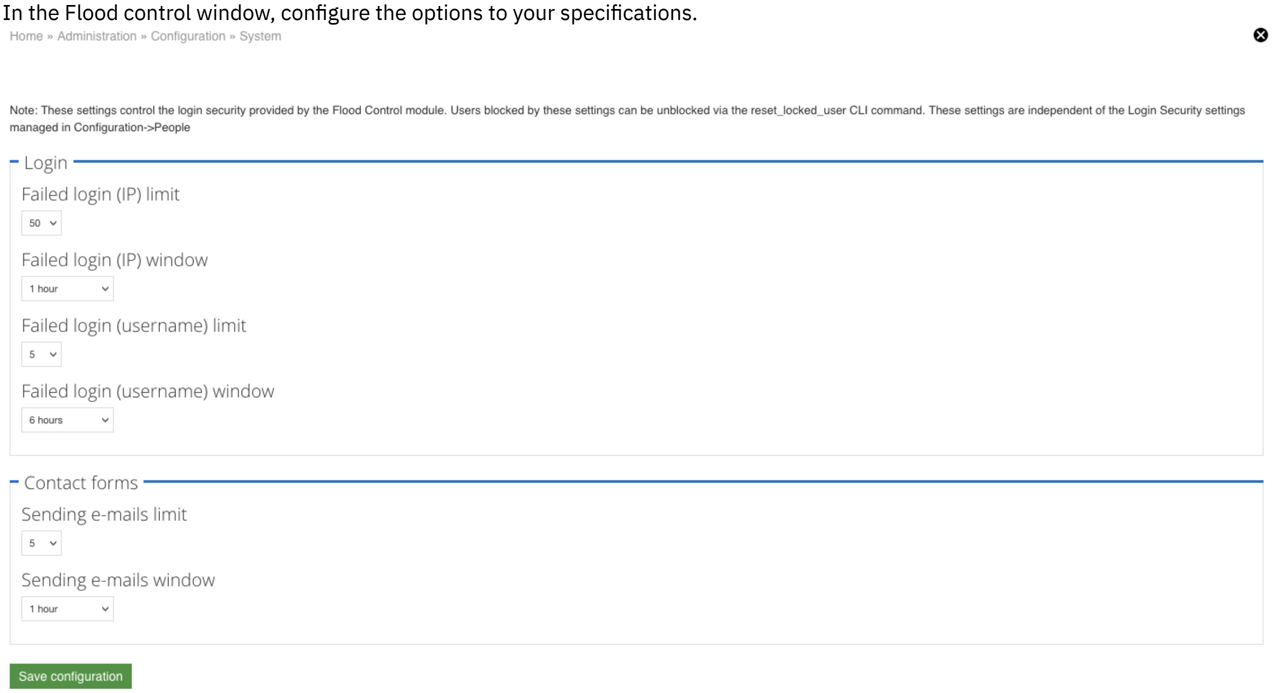
About this task

The Drupal Flood Control module provides a UI interface that can be used to configure the number of permitted login attempts, as well as the number of support emails users can send. This module is included in the Developer Portal from API Connect Version 5.0.8.4 onwards[1].

For earlier releases, the module can be installed from the Drupal organization, see https://www.drupal.org/project/flood_control. Note that the flood control feature is independent of the login security settings that can also be configured; for more information see Login security.

Procedure

1. Click Configuration in the administrator dashboard, then click System > Flood control.
2. In the Flood control window, configure the options to your specifications.



Home » Administration » Configuration » System

Note: These settings control the login security provided by the Flood Control module. Users blocked by these settings can be unblocked via the reset_locked_user CLI command. These settings are independent of the Login Security settings managed in Configuration->People

Login

Failed login (IP) limit: 50

Failed login (IP) window: 1 hour

Failed login (username) limit: 5

Failed login (username) window: 6 hours

Contact forms

Sending e-mails limit: 5

Sending e-mails window: 1 hour

3. After you have configured your flood control options, click Save configuration.
   You have successfully configured the flood control settings of your Developer Portal.

What to do next

To clear the flood control for specific Developer Portal sites and users, run the `reset_locked_user` command; see reset_locked_user.
V5.0.8 + From API Connect Version 5.0.8.5 onwards, you can clear the flood control for a specific host by running the `reset_locked_host` command; see reset_locked_host.

[1] Prior to Version 5.0.8.4, flood control settings were still enforced, but these were automatically set to 5 failed login attempts and a 6 hour lockout window.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Login security

You can configure the login security for your Developer Portal at an IP and user level.

Before you begin

You must have administrator access to complete this task.

Procedure

1. On the administrator dashboard, click Configuration > People > Login Security.
2. In the Login Security window, configure the following options to your specifications:
   - Track time
   - Maximum number of login failures before blocking a user

    - Maximum number of login failures before soft blocking a host
    - Maximum number of login failures before blocking a host
    - Maximum number of login failures before detecting an ongoing attack
3. Select the check boxes that satisfy your notification requirements in the Notifications section.
    - Disable login failure error message (this option is selected by default)
    - Notify the user about the number of remaining login attempts
    - Display last login timestamp
    - Display last access timestamp
4. Optional: Specify a user for the Select who should get an email message when a user is blocked by this module field.
5. Optional: Specify a user for the Select who should get an email message when an ongoing attack is detected field.
6. Optional: You can edit the notification texts that are displayed by configuring the options in the Edit Notification Texts tab.
7. After you have configured your login security options, click Save configuration.
   You have successfully configured the login security of your Developer Portal.

What to do next

The login security settings that are described here are independent of the flood control settings that can also be configured; for more information see Flood control.
To unblock users that are blocked by the login security settings, see Blocking and unblocking specific users.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Managing blocked IP addresses

You can block specific IP addresses from accessing your Developer Portal site.

Before you begin

You must have administrator access to complete this task.

Procedure

To manage blocked IP addresses, complete the following steps:

1. Click Configuration, then click IP address blocking.
   The IP address blocking dialog box is displayed.
2. You can complete the following tasks:
    a. To block an IP address, enter the address in the IP address field, and click Add.
       The address is added to the BLOCKED IP ADDRESSES list.
    b. To unblock an IP address by removing it from the list, click delete alongside the address that you want to unblock.

   Note: there is automatic log in 'flood' protection built into the Developer Portal. This feature means that an excess of log in attempts from the same IP address in one hour will cause that IP to be temporarily blocked.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

User name and password autocomplete

You can define whether browsers remember your user name and password. The option to turn the autocomplete functionality off is enabled by default.

Before you begin

You must have administrator access to complete this task.

Procedure

1. Click Configuration, then click No Autocomplete in the People section.
2. Select the check boxes that correspond to where you want autocomplete not to function, then click Save configuration.
   You have defined whether your user name and password is remembered by the browser.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using security questions

You can enable security questions for when a user wants to reset their Developer Portal account password. There are default security questions already configured when you allow security questions in the Developer Portal, which you can remove, as well as adding your own security questions.

Before you begin

Important: Support for using the Security questions module is deprecated and the module will not be available after the IBM® API Connect Version 5.0 release.
You must have administrator access to complete this task.

Procedure

1. Log in to the Developer Portal as an administrator.
2. Click Modules in the administrator dashboard.
3. Switch the slider for the Security questions module to ON.
4. Click Save configuration.
5. Click Configuration in the administrator dashboard.
6. Click Security questions under the PEOPLE heading.
7. Review the default security questions and settings, use the following information to configure the security questions and settings:
   From the QUESTIONS tab, you can complete the following tasks:

   - Add a new security question
     Type your new security question in to the Add a question field, then click Add.

   - Remove default security questions
     Click delete next to the security question, or questions, which you want to remove.

   From the SETTINGS tab, you can complete the following tasks:

   - Change the number of security questions required upon password reset
     Use the drop-down list under the Required number of questions heading to select the number of security questions you want the user to answer when they reset their password.

   - Allow the user to enter their own security questions when setting up their Developer Portal user account
     If you want the user to be able to set their own security questions, select the User-defined questions check box.

   - Protect additional forms using a security question
     You must use the check boxes and select at least one of the features under the PROTECTED FORMS heading to enable the security questions feature:
     - Password reset request
     - User login
     - Show "remember this computer" option on protected forms

   - Choose how longer a user will be blocked from the site after giving an incorrect security answer
     Use the drop-down list under the FLOOD CONTROL heading to select the number of hours that a user will be blocked from the site if they answer a security question wrong.

8. Click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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

You can configure the options in the Security kit to protect your website from the following types of attacks:

- Cross-site scripting
- Cross-site request forgery
- Clickjacking
- X-Content-Type-Options
- X-XSS-Protection

Note: HTTP Strict Transport Security (HSTS) is supported and enabled by default in the Developer Portal. You cannot disable HSTS.

Procedure

1. Click Configuration, then click Security kit in the System section.
2. In the Security kit module, configure your website with the security options available.
3. After you finish configuring the options, click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using two-factor authentication

You can enable two-factor authentication for the Developer Portal. This feature adds an extra level of security for users attempting to access the Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

Important: If the Portal Delegated User Registry is selected for the associated Catalog, or you are using an IBMid user registry, you cannot enable two-factor authentication for the Developer Portal.
Enable two-factor authentication (TFA) to force an additional level of security on specific types of users when accessing the Developer Portal. With TFA enabled, you log in to the site with a verification code in addition to your username and password. You can also enable the "Trusted Browsers" feature, to allow trusted browsers to bypass the TFA process for 30 days.

When TFA is enabled for the Developer Portal site, you must also configure the admin account security setting to use TFA before logging out of the account. Otherwise you will not be able to log in to the admin account again, as the site will require TFA, but the admin account won't have TFA configured.

Once TFA has been enabled for both the site and for the admin account, you can configure who needs to use TFA to access the Developer Portal. However, note again that a user must enable their TFA security setting for their account prior to their role being configured by the admin account to have to use TFA, or they will not be able to log in to the Developer Portal.

Procedure

To enable two-factor authentication for the Developer Portal site, complete the following steps:

1. On the administrator dashboard, click Configuration > People > Two-factor Authentication.
2. Select the Enable TFA check box.
   The view expands to show the configuration options for the two-factor authentication.
3. Optional: Under Login plugins, you can select the Trusted Browsers check box.
   This feature enables users to mark specific web browsers as trusted, which will cause the TFA token request from that browser to be skipped for 30 days.
4. Optional: Under Roles required to have set up TFA, select the roles that require users to set up TFA.
   Important: Users must have configured their TFA settings for their account BEFORE being assigned a role that requires TFA, otherwise login will be denied. This rule also applies to the admin account. To configure TFA settings for the admin account, you must complete the steps in the following sections before logging out of the Developer Portal.
5. Click Save configuration.
   Two-factor authentication is now enabled for the Developer Portal site.

To configure permissions for two-factor authentication, complete the following steps:

6. Click People > Permissions on the administrator dashboard.
7. In the Filter list field, type `TFA`.
   The following permissions will show under the PERMISSION heading:
   - Set up TFA for account - Sets who can turn on two-factor authentication for their account, by default this is only administrators.
   - Administer TFA - Allows the modification of two-factor authentication settings. This permission should only be granted to users with administrator roles.
8. Select and deselect the check boxes in the columns for each role to assign the previously listed permissions.
9. Click Save permissions.
   You have configured the permissions for two-factor authentication.

To configure the admin account to use two-factor authentication, complete the following steps:

10. Click the admin account name in the upper right of the UI. If you are configuring TFA settings for a a different account, click People on the administrator dashboard, and select the username of the account you want to configure.
    The account details page is displayed.
11. Select the Security tab.
12. Click Set up application and enter your current password. Click Confirm.
    The TFA setup - Application page is displayed.
13. Configure the TFA settings required.
14. Click Verify and save.
    Two-factor authentication is now enabled for the account.

Results

You have enabled and configured two-factor authentication.
V5.0.8 + You can encourage users to setup TFA on their account by applying a TFA Rules (tfa_rules) module. For more information, see Encouraging users to set up two-factor authentication on their Developer Portal account.

- V5.0.8 + **Encouraging users to set up two-factor authentication on their Developer Portal account**
  You can encourage users of your Developer Portal to set up two-factor authentication on their account by applying a TFA Rules module.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Encouraging users to set up two-factor authentication on their Developer Portal account

You can encourage users of your Developer Portal to set up two-factor authentication on their account by applying a TFA Rules module.

Before you begin

You must have administrator access to complete this task.

About this task

Important: If the Portal Delegated User Registry is selected for the associated Catalog, or you are using an IBMid user registry, you cannot enable two-factor authentication for the Developer Portal.
Two-factor authentication (TFA) forces an extra level of security on specific types of users when they access the Developer Portal. When TFA is enabled, you log in to the site with a verification code in addition to your user name and password. For information about how to enable and configure TFA, see Using two-factor authentication.

After a Developer Portal site is configured to use TFA, you can apply a TFA Rules (tfa_rules) module to encourage all authenticated users to set up and use TFA on their account. By using the following steps, the module is enabled and configured to always redirect authenticated, but non-TFA, users to the TFA security page immediately after they log in to the Developer Portal. The redirect is accompanied by a configurable message that encourages users to set up TFA for their account.
Note: It is possible to mandate specific roles to use TFA. However, a user's account must be configured for TFA before being assigned a role that requires TFA, otherwise the user becomes locked out of their account.

Procedure

To enable two-factor authentication and encourage authenticated users to set up TFA for their account, complete the following steps.

1. Enable two-factor authentication for the Developer Portal site by following the instructions in Using two-factor authentication. Ensure that when you configure the permissions for TFA, the Set up TFA for account option is enabled for all the user roles that you want to set up TFA.
2. Enable the TFA Rules module.
    a. On the administrator dashboard, click Modules.
    b. Enter `TFA` into the search field of the Modules pane to find the TFA Rules module, and set the module to ON.
    c. Click Save configuration.
3. Configure and import a User redirect to TFA setup on login rule.
   The TFA Rules module includes the User redirect to TFA setup on login rule, but this rule requires some editing to make it compatible with the Developer Portal. You can edit the rule directly in the Developer Portal, or create a new rule, but it is recommended that you use the following rule template to import and overwrite the original User redirect to TFA setup on login rule. For more information about rules, see Applying rules in the Developer Portal.

    a. On the administrator dashboard, click Configuration > Workflow > Rules.
    b. Click Import rule.
    c. Paste the following rule configuration into the Import field:

```
{ "rules_tfa_user_login_redirect_setup" : {
    "LABEL" : "User redirect to TFA setup on login",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "tfa_rules", "rules", "rules_i18n" ],
    "ON" : { "user_login" : [] },
    "IF" : [
      { "NOT tfa_rules_user_has_tfa" : { "account" : [ "account" ] } },
      { "NOT user_is_blocked" : { "account" : [ "account" ] } },
      { "NOT text_matches" : { "text" : [ "site:current-page:path" ], "match" : 
"ibm_apim\/activate" } }
    ],
    "DO" : [
      { "drupal_message" : {
          "message" : "Two-factor authentication (TFA) is strongly recommended; please 
configure your account security.",
          "type" : "warning",
          "repeat" : "0"
        }
      },
      { "redirect" : { "url" : "\/user\/[account:uid]\/account\/security\/tfa", "force" : "0" 
} }
    ]
  }
}
```

Where **message** can be updated to display a message of your choice.
This configuration checks that the user doesn't already have TFA setup, that the user isn't blocked, and that the user has activated their account, before redirecting them to the TFA setup page.

d. Select the Overwrite check box.
e. Click Import.

Results

You enabled a rule configuration that encourages authenticated users to use two-factor authentication when they log in to your Developer Portal site.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

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 Honeyspot 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. Click Configuration in the administrator dashboard.
2. In the Content Authoring section, click Honeypot configuration.
3. 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.

4. To have details of all blocked form submissions written to the log file, select Log blocked form submissions.
5. 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.

6. 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.
7. When done, click Save configuration.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Developer Portal best practices for administrators

You are recommended to follow the best practices that are listed in the next sections, to ensure you get the best experience from the Developer Portal appliance.

General advice

You must treat the Developer Portal as an appliance, and execute only the Linux operations that are described in the documentation or are advised elsewhere by IBM.

To configure the server, you must use the commands that are provided in the documentation, such as `set_smtp`, and `set_ntp`. Do not edit the Linux configuration files directly.

For a list of all the available commands, see Useful Developer Portal CLI commands.

Configuration advice

You should take a full backup before doing anything you're unsure of, to supplement the regular backups that you already take. For information about backups, see Backing up and restoring the Developer Portal.

You must always use the same Fix Pack level on all of the Developer Portal servers and the API Manager servers. The Fix Pack level is the last number on the version string, for example, the number 4 in version 5.0.8.4.

The Operating System timezone for the Developer Portal should be set to UTC, and the same NTP server should be used by all members of the API Connect deployment.

Actions to avoid

You must not perform the following actions unless specifically advised to do so by IBM.

Site administration actions to avoid:

- Patching Drupal modules that are already included with the product.
- Installing custom modules from the command line. If you're installing custom modules or themes, you should install them by using the Developer Portal UI to ensure they are saved into the right directory, and they have the correct permissions. For more information, see Installing additional modules.
- Deleting items from the Developer Portal core configuration. For example, don't delete any fields that are defined by the Developer Portal; fields can be hidden from specific view modes if required, but don't delete the field itself.
- Updating Drupal modules in the Developer Portal UI. If a module update is needed, it's provided in a Developer Portal iFix. An iFix can be requested by raising a service request, see IBM Support.

Server administration actions to avoid:

- Editing any of the Nginx configuration.
- Installing and/or running any other software products on the Developer Portal server.
- Modifying any user permissions on the server.
- Making any changes to the SSH configuration.
- Making any changes to the Developer Portal database, or accessing the database directly, unless under the instruction of IBM Support.

Related information

For general IBM® API Connect best practices, see IBM API Connect best practices.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Migration support for your Developer Portal

From IBM® API Connect Version 5.0.8.6 onwards, migration support is provided for exporting certain Developer Portal users and content.

About

When migration support is available in IBM API Connect Version 2018, you can use the tooling to export any Portal Delegated User Registry (PDUR) user information, as well as specific custom Developer Portal content. For more information, see the following topics.

- V5.0.8 + **How to export Portal Delegated User Registry user information**
  From IBM API Connect Version 5.0.8.7 onwards, if your Developer Portal uses Portal Delegated User Registry, you can export the user information that is stored in the Developer Portal local database.
- V5.0.8 + **How to export custom Developer Portal content**
  From IBM API Connect Version 5.0.8.6 onwards, you can export custom Developer Portal fields, image files, and attachments.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

How to export Portal Delegated User Registry user information

From IBM® API Connect Version 5.0.8.7 onwards, if your Developer Portal uses Portal Delegated User Registry, you can export the user information that is stored in the Developer Portal local database.

Before you begin

You must have administrator access to complete this task.

Note: The ability to migrate Portal Delegated User Information is available only from IBM API Connect Version 5.0.8.7 onwards. If your deployment is at an earlier version, you must upgrade to Version 5.0.8.7 or later before migrating. See Upgrading your API Connect cloud for information about upgrading in Version 5.

About this task

When Portal Delegated User Registry (PDUR) is selected for a Catalog, the user management is delegated from the management server to the Developer Portal. In this instance, if you want to migrate users from an IBM API Connect Version 5 Developer Portal, to an IBM API Connect Version 2018 Developer Portal, you must export the user information that is stored in the Developer Portal local database. This information can then be used by the migration tooling to migrate the PDUR user registry.

The following two methods can be used for exporting the PDUR user information:

1. You can use the Developer Portal UI to export the PDUR user information on a per site basis into a .json user-mapping file. If you have more than one site, before the user information can be used by the migration tooling you must manually combine the .json files into one compressed pdurexport.tgz file.
2. You can use a Developer Portal CLI command to export the PDUR user information for all of the sites on a Developer Portal server into a pdurexport.tgz file. This compressed file will contain all of the .json user-mapping files for that server.

Each .json file contains a list of all of the PDUR user information that's stored in the Developer Portal local database. The user information for all of the user registration methods that are used in the specific Developer Portal are included in the same file. For example, local user registry, LDAP user registry, third-party authentication, OpenID Connect, and any custom user registration method.
Note: User-mapping files must not be edited, or the migration may not complete successfully.

Procedure

- The following instructions show you how to export the PDUR user information on a per site basis by using the Developer Portal UI.
  1. In the Developer Portal UI, click People > Export.
     The Export configuration window is displayed.
  2. Complete the Filename to save as field (you can accept the default file name if preferred).
  3. Optional: If you have a large number of user accounts, you can click Advanced Settings and update the export settings that are suitable for your server configuration.
  4. Click Download File and download the .json file to your preferred location.
  5. Optional: If you have more than one site, you must export the PDUR user information for each site, and then compress all of the .json files into one pdurexport.tgz file.
- The following instructions show you how to export the Portal Delegated User Registry user information for all of the sites on a Developer Portal server by using the CLI.
  1. Log in to the Developer Portal CLI.
  2. Run the following command:

```
pdur_user_export
```

The status of the export is displayed, for example:

```
admin@apimxxx123:~$ pdur_user_export
portal123.company.com/api-provider/production: Exporting pdur users
portal123.company.com/api-provider/preprod: Exporting pdur users
portal123.company.com/api-provider/uat: Exporting pdur users
portal123.company.com/api-provider/development: Skipping as not a pdur enabled site
portal123.company.com/api-provider/sandbox: Skipping as not a pdur enabled site
~/pdurexport ~
5c20b903e4b0485cbbd81775.5c20bf43e4b0485cbbd81795.json
5c20b903e4b0485cbbd81775.5c2cddbae4b0485cbbd818a0.json
5c20b903e4b0485cbbd81775.5c335211e4b0485cbbd8196e.json
~
Output can be found in /home/admin/pdurexport.tgz
```

Note that if a site doesn't contain any PDUR data, that site is skipped. The naming convention used for the individual user-mapping files is *provider_org_id*.*catalog_id*.json. So the above example output shows that there was one provider organization on the server, and that organization had five Developer Portal sites, three of which contained PDUR user information.
The PDUR user information for all of the sites on the server is exported into a pdurexport.tgz file.

What to do next

You can use the tooling in V2018 to migrate your PDUR user information. For more information about migration, see Migrating a Version 5 deployment to Version 2018 and Migrating your Developer Portal from Version 5 to Version 2018.

Related tasks

- How to export custom Developer Portal content

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

How to export custom Developer Portal content

From IBM® API Connect Version 5.0.8.6 onwards, you can export custom Developer Portal fields, image files, and attachments.

Before you begin

You must have administrator access to complete this task.

About this task

If you've created custom content in your Developer Portal, and you want to migrate this content from an IBM API Connect Version 5 Developer Portal, to an IBM API Connect Version 2018 Developer Portal, then you must export the custom content that's stored in the Developer Portal local database. The custom content that can be exported includes the following items:

- 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, this content can then be migrated into the IBM API Connect Version 2018 Developer Portal. Custom content is exported into a .zip file. This compressed file will contain a copy of all of the custom content files that are stored in the Developer Portal local database.
Note: The custom content file must not be edited, or the migration may not complete successfully.

Procedure

1. In the Developer Portal, click Content > Export custom content.
   The Export custom content configuration window is displayed.
2. Complete the Filename to save as field (you can accept the default file name if preferred).
3. Click Download File and download the .zip file to your preferred location.

What to do next

When migration support is available in the Version 2018 Developer Portal, you can use the tooling to migrate the custom content. For more information about migration, see Migrating a Version 5 deployment to Version 2018 and Migrating your Developer Portal from Version 5 to Version 2018.

Related tasks

- How to export Portal Delegated User Registry user information

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Troubleshooting the Developer Portal

This page describes how to troubleshoot common problems that can occur when using the Developer Portal. The steps outlined apply to IBM® API Connect version 5.0.8.4 and later; some of the troubleshooting commands might be missing from earlier releases. It is recommended that you regularly upgrade to the latest fix pack or interim fix.

Refer also to Developer Portal best practices for administrators.

You can use the following links to navigate to the sections on this page:

Basic Checks

Many problems can be caused by incorrect configuration in the basic deployment settings; here are some basic checks that you should do:

- On each Portal node, run the command **status**, and confirm that the following properties are correct:
  - `APIC Hostname`: must match the API Connect Management cluster address, as configured in the Cloud Manager user interface, and if the Developer Portal API's port has been changed in the Cloud Manager, then the same port must be specified here. This value is configured with the **set_apim_host** command.
    For information on configuring the Management cluster in the Cloud Manager, see Configuring the Management service. For information on setting the Developer Portal API's port, see Specifying the cloud settings.
  - `APIC IP`: the IP address of the host specified by `APIC Hostname`.
  - `Devportal Hostname`: the host name of this Portal server, it must be unique, and is set with the **set_hostname** or **set_apim_host** command.
  - `Devportal IP`: the IP address of the host specified by `Devportal Hostname`.
  - `APIC Certificate Status`: indicates whether the certificate that the Portal has for communicating with the Management server is valid. The commands **download_apim_cert** and **set_apim_cert** can be used to set this.

  If you identify any problems with these properties, rerun the following Portal configuration commands as necessary:
  - **set_hostname**
  - **set_apim_host**
  - **download_apim_cert** and **set_apim_cert**
- Load balancers between the Management server cluster and the Portal cluster must not modify the traffic, and if the Management cluster address port was changed in the Cloud Manager, then the load balancer must use the same port to listen for traffic from the Portal nodes.

Sites are unavailable

If users are unable to access some or all Portal sites, then make the following checks:

1. Can the site be accessed locally? You can confirm local access by running the **check_site** command (run **list_sites** first to get a list of sites); if the site returns status code 200, then there is likely to be a network issue between the user's browser and the Portal server. The log file /var/log/nginx/access.log logs all site access attempts, and if nothing is logged here when the user attempts to access the site then this is also evidence of a network problem between the browser and the Portal server. Check your network routing and load balancing configuration.
2. Does the site URL that the user enters in their browser match the site URL returned from the **list_sites** command? These URLs must match.

API Product updates are not appearing in the Portal

If a publish, delete, deprecate, or retire operation on a Product is not reflected in the Portal within a few seconds, then make the following checks:

1. Does the update appear within 15 minutes? If it does, then try running the command **resubscribe_webhooks**. If that doesn't fix the problem then collect the logs and open a support request.
2. If the update still doesn't appear after 15 minutes, check that the configuration for the communication between the Portal and the Management server is correct - see Basic Checks.
3. Confirm that the site is configured as expected for the Catalog in the Management server API Manager user interface. It should show the same site URL as that listed from the **list_sites** command on the Portal server.

User registration

When a new user is registered, the Management server sends an activation link and welcome email to the new user. When the user accesses this link, further validation is done between the Portal, the Management server, and any third-party authentication site that might have been configured. Network routing and configuration issues between any of these components can cause user registration problems.

Make the following checks:

1. Make the checks detailed in Basic Checks.
2. When attempting a registration step, **tail** the log files /var/log/cmc.out on the Management server, and /var/log/syslog on the Portal. The cause of the failure might be clear from the messages in these logs; the absence of any log messages related to registration suggests a network issue between the Portal and the Management server.

If the Admin user registration link has been lost, or the email was never received, the command **site_login_link** can be used to regenerate and display the link.

User login

The same checks apply here as with User registration. Additionally, users might have been blocked with login security or flood control, in which case see: Blocking and unblocking specific users and reset_locked_user. Note that resetting passwords does not unblock users.

Clustering and database problems

Common problems are as follows:

Timezone and time synchronisation

The operating system timezone (not the PHP timezone) of all cluster members **must** be UTC, and they must all use the same NTP server so that their times are synchronized to sub-second accuracy.

SSH Settings

All cluster members should be able to make SSH connections to each other without requiring a password. The SSH configuration is handled by the clustering scripts. Users should not attempt to modify SSH settings themselves, and if any modifications have been made these should be reverted.

Problems with database clustering

These problems are indicated by the output of the **status** command showing the cluster members database status as anything other than `Active (Primary)`, unless a transitory operation is running.
For a transitory operation, the **status** command might show various statuses other than `Active (Primary)` during clustering operations and upgrades; these statuses are usually `Starting`, `Stopping`, `SST-Running-X%` and sometimes, for a few minutes, `Preparing`. These states are not a cause for concern unless the operation completed more than 15 minutes earlier, or it returned an error message.

Make the following checks:

1. Check that all members of the cluster are listed in the output of the **status** command when run on every node. If there is a mismatch between nodes, this can be corrected by explicitly setting the cluster members on each node with the **set_cluster_members <member_1_IP> <member_2_IP> <member_3_IP> ...** command.
2. On one of the nodes, run the command **bootstrap_cluster -bf**; this will try to restart the database on all the nodes, forcing it to stop and restart.
3. You can use the **stop_db** and **start_db** commands to attempt to restart the database on an individual node.
4. Check the log file /var/log/syslog for MySQL errors.

Do not attempt to modify the MySQL configuration; if problems persist after checking the above then gather logs with the **generate_logs** command and open a support ticket.

Problems with file synchronization

When a new cluster member is added, the **status** command will report that file synchronization is taking place. On a simple deployment with just a few sites this might take about 10 minutes, on larger deployments with many sites and higher network transfer times between nodes this process can take hours.
It is usual for the reported number of files being synchronized to change (increasing and decreasing) during the process, but if it appears to be taking too long for the size of deployment, or regularly restarting or cycling, then gather the logs with the **generate_logs** command and open a support ticket.

Site creation

If, when attempting to create a site in the Management server API Manager user interface, an error is returned or nothing appears to happen, and no site creation email is received, then make the following checks:

1. Refer to Basic Checks to confirm that the Management server and Portal are able to communicate with each other.
2. Run the **list_sites** command on all Portal nodes and see if the new site is listed. If the site is not listed, or it is reported to be in an error state, or is stuck in the `INSTALLING` state for more than 30 minutes, then gather the logs with the **generate_logs** and open a support request.

Backup and restore

Common problems with these operations are due to unexpected files and modules in the site directories; see Developer Portal best practices for administrators.

The log files /var/log/devportal/command_line.log and /var/log/devportal/site_action.log might provide clues as to where the operation is failing. Otherwise gather the logs with the **generate_logs** command and open a support request.

Upgrade

After an upgrade failure, the portal could be in one of several possible states; run the **status** command to confirm which, taking note of the version numbers from the output, for example:

- `System version: 7.x-5.0.8.4-iFix-20180828-2355`: this refers to the version of the portal executables and libraries.
- `Distribution version: 7.x-5.0.8.4-iFix-20180828-2206`: this refers to the version of the portal site template.

Pay attention to the datestamps in these version numbers; they should usually have the same date, although occasionally the Distribution version might be a day earlier, but not the same time. In a successful upgrade, both should show the same datestamp as that which is shown in the file name of the fix pack that was just installed.
If the System version has updated successfully but the Distribution version hasn't, or if neither have updated, then try re-running the fix pack using the **-f** (force) option.

If both System version and Distribution versions are shown to have been updated, and the error reported was due to sites failing to upgrade, then you can attempt site upgrade again with the following command **upgrade_devportal -p <platform> -s <site>**, where **<platform>** is taken from the output of the **list_platforms** command, for example:

```
list_platforms
platform_devportal_7_x_5_0_7_2_20170628_2109 => devportal-7.x-5.0.7.2-20170628-2109 : Template Exists
platform_devportal_7_x_5_0_8_2_20180121_2206 => devportal-7.x-5.0.8.2-20180121-2206 : Template Exists
platform_devportal_7_x_5_0_8_3_20180508_2206 => devportal-7.x-5.0.8.3-20180508-2206 : Template Exists 
(default)
```

and <site> is taken from the output of the list_sites command, for example:

```
list_sites
5b97c411e4b014572ebe29e1.5b97c411e4b014572ebe29ed => apimdev0030.hursley.ibm.com/iainsorg/sb (INSTALLED)
5b97c411e4b014572ebe29e1.5b97c453e4b014572ebe2a01 => apimdev0030.hursley.ibm.com/iainsorg/new-catalog-1 
(INSTALLED)
```

The following example shows a sample **upgrade_devportal** command:

```
upgrade_devportal -p devportal-7.x-5.0.8.3-20180508-2206 -s 
5b97c411e4b014572ebe29e1.5b97c411e4b014572ebe29ed
```

Emails

The Developer Portal sends emails for forum subscriptions, site creations, and site contact forms, by using the SMTP server that was configured with the **set_smtp** command. Use this command to configure SMTP and to send a test email. Check the file /var/log/mail.log for any errors in sending emails.

User registration and password reset emails are not sent by the Developer Portal, they are sent by the Management server and are logged in /var/log/cmc.out on the Management server.

Performance

Make the following checks:

- Ensure that the minimum hardware requirements have been met; see Deploying the Developer Portal OVA file.
- Confirm that there is sufficient disk space by running the command **df -h**.
- If no major operations are in progress, such as new site creations, upgrades, or clustering, and you have at least 3 GB of spare disk space, run the command **hardware_performance_test**, this tests the CPU and disk I/O speeds; note that the results will vary with each run and you should take an average over several runs. The Developer Portal is particularly sensitive to disk I/O speed, anything below 50 MB per second is likely to cause serious performance problems.

Removing a problem node from a cluster

If you have a node that you suspect is broken, and that is causing the cluster to be inoperable, then you can complete the following steps to shut down and remove the broken node:

1. Run the command **sudo halt -p**
2. Remove the node from the cluster by running the following command on the other nodes to explicitly specify all the nodes that should be in the cluster, excluding the one being removed:

   ```
   set_cluster_members <member_1_IP> <member_2_IP> <member_3_IP> ...
   ```

3. If the databases are down on the other cluster members, they should restart themselves. If they do not restart, run the following command to start any of the databases that are down:

   ```
   bootstrap_cluster -b
   ```

4. If the databases are up but not functioning or clustering together, run the following command to stop all of the databases and restart them:

   ```
   bootstrap_cluster -bf
   ```

Analytics

Common reasons for analytics problems in the Developer Portal are as follows:

- Portal Delegated User Registry is being used. Analytics are not supported if this user registry is selected.
- The API Manager port that is specified in Cloud Manager user interface Settings > TLS Profiles page is not 443 or does not match the custom port specified in the `APIC Hostname` output of the **status** command. This port must match the Developer Portal APIs port, also specified in Cloud Manager user interface Settings > TLS Profiles page, and must be the port specified when configuring the Developer Portal with the **set_apim_host** command; see Basic Checks.

Specific problems

When I attempt to deploy the Developer Portal OVA template I receive the following error: `The following manifest file entry (line 1) is invalid`

You must deploy the Developer Portal OVA template by using a version of the VMware vSphere Client that supports the SHA-256 Cryptographic Hash Algorithm.

I see the following message in the Developer Portal user interface: `The system is currently experiencing problems. Please try again later.`

Check the log files /var/log/devportal/background_sync.log and /var/log/syslog for errors. You can force re-synchronization to determine whether the problem was temporary, by completing the following steps:

1. Log in to your Developer Portal as the administrator.
2. On the administrator dashboard, click Configuration > System > Cron.
3. From the Operations list for the job that is titled Background sync, select Run.

If you see the message `Allowed memory size of number_of_bytes exhausted` in the log file /var/log/syslog, run the command **php_max_memory 1024** to increase the maximum memory to 1024 MB, then run the background synchronization task again.

In the log files, I see warnings that contain the string `using password: NO`

You can ignore these warnings; the following example shows such a message:

```
Sep 27 17:52:46 myservername mysqld: 2017-09-27 17:52:46 5085 [Warning] Access denied for user
'root'@'localhost' (using password: NO)
```

I see the error `Access Denied` when updating a cluster

If, when adding a new Developer Portal node to a cluster, you see the following error message:

```
Access denied for user 'myuser@myhost' (using password: password)
```

complete the following steps:

1. Log in to the Developer Portal CLI as the root user by entering the command: **sudo -i**
2. Check the file /etc/mysql/debian.cnf in the new and existing node or nodes to see if the passwords match. If they do not match, copy the password from the existing node into the new node, and attempt the update process again.

Restoring a standalone Portal server by using a backup file

To recover a standalone Portal server onto a new OVA deployment by using a backup file, complete the following steps:

1. Ensure that the Portal backup file is available on a separate FTP/SFTP server.
   Note: Do not confuse a Portal backup file with a Portal site backup file. By default, a Portal backup file has the following filename format: `apim-portal-<hostname>-<datestamp>.tgz`.
2. Ensure that you have the Developer Portal OVA file that matches the Portal server that you are restoring (it must have the same fix pack and build number, for example 5.0.8.3-APIConnect-Portal-Ubuntu16-20180508-1349.ova).
3. Shut down and delete your corrupted Portal server.
4. Deploy the Developer Portal OVA file in place of the deleted server, and set the same IP address and host name as the deleted server had. For detailed instructions, see Deploying the Developer Portal OVA file.
5. Configure the Developer Portal to connect to the Management server. For detailed steps, see Installing the Developer Portal.
6. Copy the backup file to the newly deployed Portal server by using FTP, SFTP, or SCP.
7. Then, restore the Portal by using the `restore_devportal -scn` command. For example:

   ```
   restore_devportal -scn backup_file_absolute_path
   ```

where ***backup_file_absolute_path*** is the location of your backup file. For more information, see restore_devportal.

8. Run the **`status`** command. Check that the server is marked as **`SUCCESS`**.

Restoring a cluster of Portal servers by using a backup file

To recover a cluster of Portal servers onto a new OVA deployment by using a backup file, complete the following steps:

1. Ensure that you have a backup file of one of the Portal servers. If the Portal Cluster Address (as configured in the Cloud Manager) was set to one of the Portal servers (instead of a load balancer), then use the backup from that server.
2. Shutdown and delete all of the Portal cluster servers.
3. Deploy and configure a single Portal server with the same hostname and IP address as that of the backup, and connect it to the Management server (by using **`set_apim_host`** and **`download_apim_cert`**/**`set_apim_cert`**). For detailed steps, see Installing the Developer Portal.
4. Run the **`status`** command to confirm that the Portal server was setup with no errors.
5. Run the following command:

   ```
   sudo mkdir /etc/mysql/certs ; sudo chmod 750 /etc/mysql/certs ; sudo chgrp mysql /etc/mysql/certs
   ```

6. Copy the backup file to the newly deployed Portal server by using FTP, SFTP, or SCP.
7. Then, restore the Portal on the new server by using the **`restore_devportal -scpn`** command. For example:

   ```
   restore_devportal -scpn backup_file_absolute_path
   ```

   where ***backup_file_absolute_path*** is the location of your backup file. For more information, see restore_devportal.
8. Run the **`status`** and **`list_sites`** commands, and confirm that the Portal and its sites have been restored and that there are no errors.
9. Run the following command to create a cluster:

   ```
   set_cluster_members -c
   ```

10. Deploy the remaining Portal cluster servers, and set them to the same IP address and hostname that they had before (by using the **`set_hostname`** command). Then run the following command:

    ```
    set_cluster_members hostname/IP_of_existing_cluster_member
    ```

    where *hostname/IP_of_existing_cluster_member* is the hostname or IP address of the server where steps 3-9 were run.
11. You can monitor the progress of the clustering by running the **`status`** command.

Your cluster of Portal servers are now restored onto a new OVA deployment.

Restoring a Portal site by re-creating

As an alternative to restoring a Portal site from a backup file, a site can be re-created from the Management server. The Portal users, and their applications and subscriptions, are all stored on the Management server, and so this data is saved. However, the following points must be noted:

- It is not possible to re-create Portal Delegated User Registry (PDUR) sites; this type of site can be restored only from a backup file.
- Any customizations that were made to the site, for example modifications to forms, uploaded images, custom themes and modules, are lost when the site is re-created.

To re-create a Portal site by using the Management server, complete the following steps:

1. In the API Manager UI, select the Catalog > Settings > Portal page that corresponds to the Portal site that you want to re-create.
2. Set the Select Portal field to None, and click Save.
3. After a few minutes the site is deleted from the Portal; run the **`list_sites`** command on the Portal CLI to confirm that the site is removed.
4. Return to the same Catalog > Settings > Portal page in the API Manager UI, set the Select Portal field back to IBM Developer Portal, and set the URL field to what it was set to previously.

After a few minutes the Portal site is re-created, and an admin user invitation email is sent.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x 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.

For API consumers Using the Developer Portal V5.0.8 + Subscribing to a Plan with pricing	Getting started configuring the Developer Portal Customizing themes Installing a theme Displaying recent blog posts
User authentication and security Using the Portal Delegated User Registry Configuring writable LDAP Authenticating users by using an external server	Advanced customization Setting up custom workbench moderation for blogs Adding and exporting custom fields Configuring Rules V5.0.8 + Configuring a custom sort order for APIs and Products Synchronizing application credentials with an external server

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Using the Developer Portal

Follow the lessons in this tutorial to learn how to use the Developer Portal. You can then monitor how your APIs are being used by using the administrator dashboard in the API Manager UI.

Before you begin

You must have a Developer Portal installed, and created an invoke REST API. For more information, see Installing the Developer Portal and Creating an invoke REST API.

Download the BankA logo.

About this task

In this tutorial you are going to complete the following lessons:

- Enabling the Developer Portal through the API Manager.
- Creating a developer account in the Developer Portal
- Inside 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.

Enabling the Developer Portal through the API Manager

By proceeding with the following steps, you will create an administrator account:

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Click Dashboard, then click Sandbox.
3. In Sandbox, click Settings > Portal.
4. Select IBM Developer Portal.
5. Enter your Developer Portal URL, which will have been configured in the DNS settings during Developer Portal installation, then click Save.
   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. When the link is active and you have accessed it, you can change the password of this administrator account.

Creating a developer account in the Developer Portal

By proceeding with the following steps, you will 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. Click Create an account. Your Developer Portal site is activated and you receive a confirmation email as a result. Click the account activation link in the confirmation email to activate your account.
3. To use the Developer Portal, click Login and sign in with the user credentials you specified.

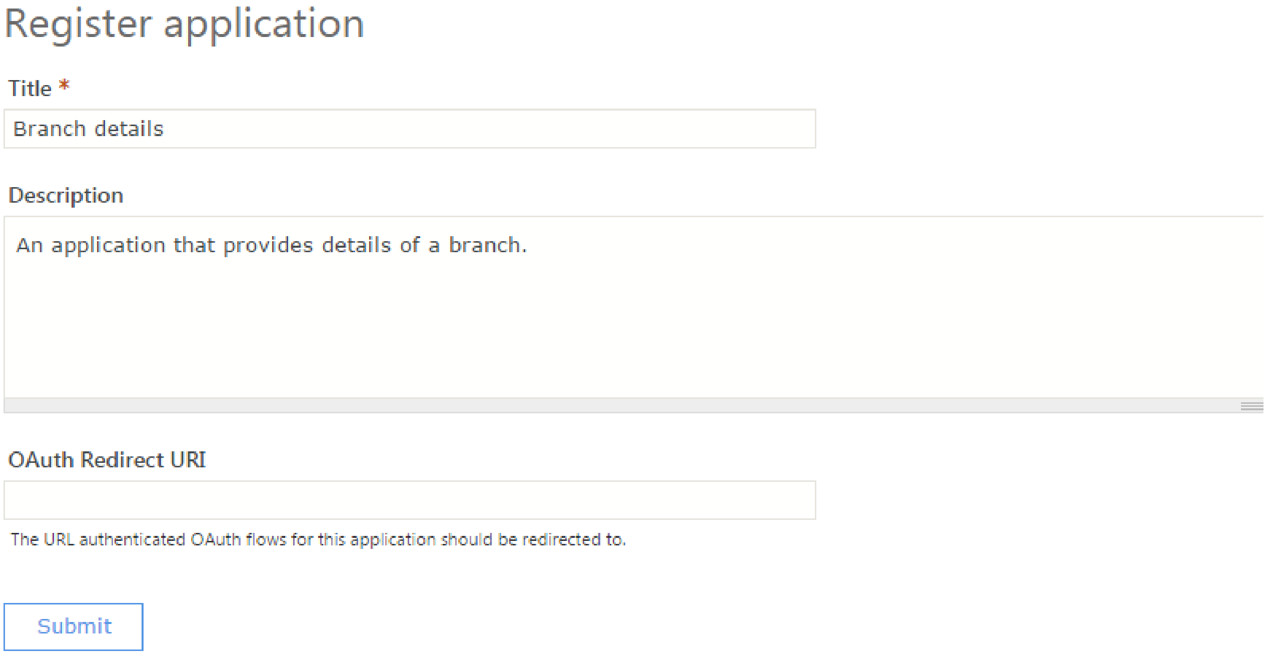
Inside the Developer Portal

By proceeding with the following steps, you will create and register a new App:

1. To view the Products available in the Developer Portal, click API Products.
2. To view the details of the Branches API that you created in the tutorial, Creating an invoke REST API, click Banking Services, then click Branches.
3. Review the GET /details operation that is available for use under the Paths heading.
4. To download and review the OpenAPI (Swagger 2.0) definition of the Branches API, click the Download icon ⬇ .
5. In the Developer Portal, click Apps.
   Note: You cannot register new applications if you are logged in as the administrator.
6. V5.0.5 + Click Create new App.
   V5.0.4 and earlier Click Register new Application.
   The "Register application" window opens.
7. Enter the following values for the application that is being registered.

Table 1. Values for registering the application

Field Name	Value
Title	`Branch details`
Description	`An application that provides details of a branch`



Register application

Title *

Branch details

Description

An application that provides details of a branch.

OAuth Redirect URI

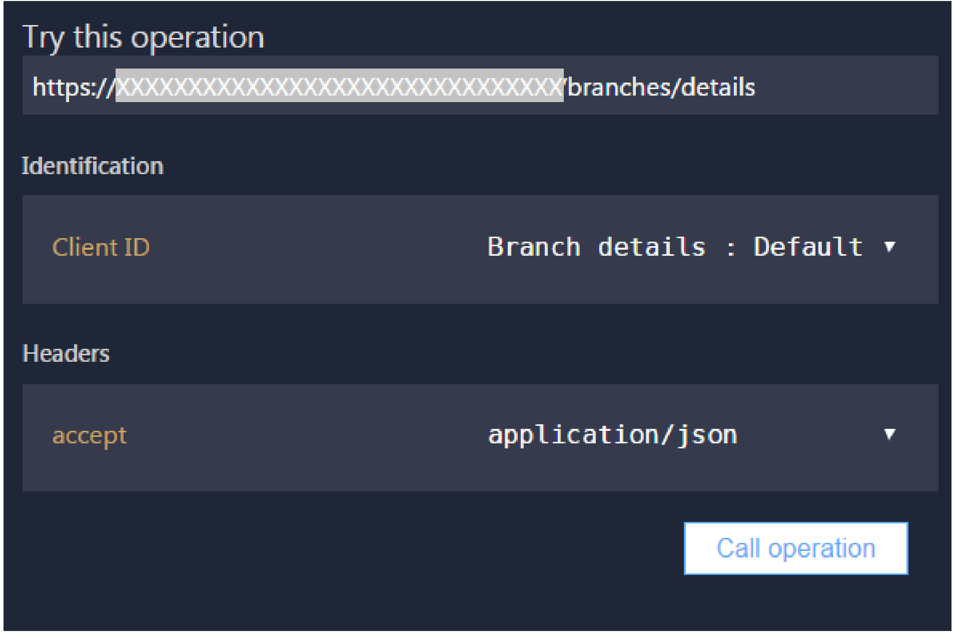
The URL authenticated OAuth flows for this application should be redirected to.

8. Click Submit.
9. To add an image for the application, click Update in the Branch details App. The Upload application image window displays.
10. Click Choose file and select the BankA logo, banka_logo.png, that you downloaded earlier.
11. Click Submit.
12. To select the Plan that you want to use with this application, click API Products.
13. Click the Banking Services Product. The details of the Plan named Basic are displayed.
14. Click Subscribe.
15. Under the Application heading, select the Branch details radio button.
16. Click Subscribe.

You have created and registered a Branch details App and subscribed it to a Plan.

Testing the Branches API in the Developer Portal

1. Click API Products in the Developer Portal dashboard.
2. Click the Banking Services Product, then select the Branches API from the list on the left of the window.
3. Scroll down the right pane of the display to the Try this operation section, then click Call operation.



Try this operation

https://XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/branches/details

Identification

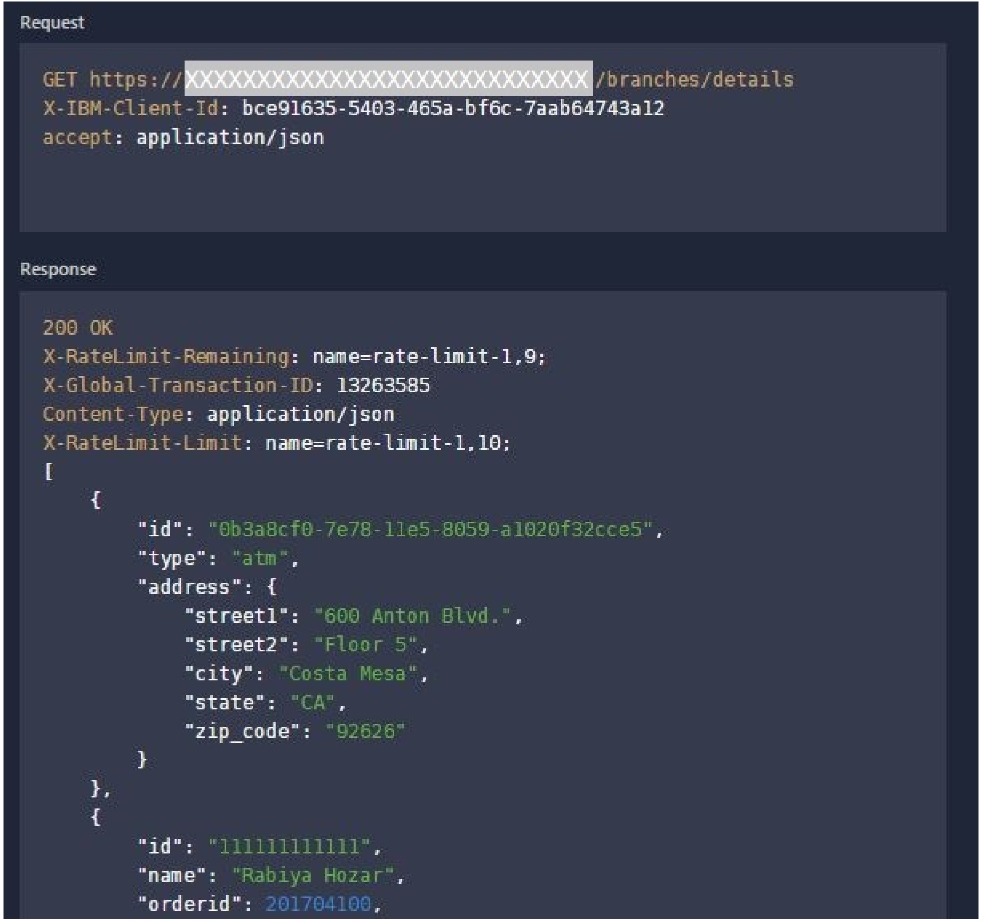
Client ID: Branch details : Default ▼

Headers

accept: application/json ▼

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.

4. A returned response of `200 OK` and the message body are displayed, indicating that the REST API operation call was successful.



Request

```
GET https://XXXXXXXXXXXXXXXXXXXXXXXXXXXXX /branches/details
X-IBM-Client-Id: bce91635-5403-465a-bf6c-7aab64743a12
accept: application/json
```

Response

```
200 OK
X-RateLimit-Remaining: name=rate-limit-1,9;
X-Global-Transaction-ID: 13263585
Content-Type: application/json
X-RateLimit-Limit: name=rate-limit-1,10;
[
    {
        "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": "111111111111",
        "name": "Rabiya Hozar",
        "orderid": 201704100,
```

What you did in this tutorial

In this tutorial, you completed the following activities:

- Enabled a Developer Portal site and created an administrator account.
- 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.

For more tutorials, see the API Connect website on IBM developerWorks® https://developer.ibm.com/apiconnect/.

What to do next

- Securing an API with a client ID and client secret
- Securing APIs by using an LDAP user registry
- DataPower Gateway only Securing an API by using OAuth 2.0

Related information

- IBM API Connect overview
- API Manager
- Publishing a Product
- Creating an invoke REST API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Tutorial: Subscribing to a Plan with pricing

When you have access to a Developer Portal, you can set up your billing information and subscribe to the Plan with automatic credit card billing.

Before you begin

This tutorial requires you to add credit card information for your Developer Portal account, so you must have organization owner permission complete all of the steps. If you do not have owner access, the billing information is not displayed.

About this task

To subscribe to a Product that contains a Plan with billing, complete the following steps:

Procedure

1. Log in to the Developer Portal that contains the Product that you want to subscribe to.
2. Set up your billing information.
    a. Select your username in the title bar, and select My organization.
    b. Select the Billing tab.
    c. Select Update to add your credit card information.
    Tip: Stripe provides a list of some test credit card numbers that you can access at: https://stripe.com/docs/testing#cards.
    d. Complete the fields with your Billing Information, and select Payment Info.
    e. Enter your credit card information in the required fields, and select Update Billing Details.
3. Select the API Products tab.
4. Select the Product that you want to subscribe to.
   A list of plans for the Product is displayed.
5. Select Subscribe for the Plan that you want to subscribe to.
6. Select your registered application that you would like to associate with the Plan, and select Subscribe.

If there are no applications that are available, register a new application. You must have a registered application to subscribe to the Plan.
Note: If there is an application that is listed, but not available for selection, it might already be subscribed to a Plan for that Product.

7. If you are running a test, and also have access to the Stripe account that was set up for the administration of the payments, open your Stripe account to verify that your the plan appears in the dashboard when you select Subscriptions.
If you do not have administrator access to the Developer Portal that contains the Plan that you are subscribing to, then you cannot access the Stripe account to verify it.

Results

You are subscribed to the Plan that you requested. Your credit card will be billed according to the plan that you subscribed to, following any free trial days that were included in the Plan. Monthly subscriptions are billed the same time each month.

Related information

- Tutorial: Defining a subscription Plan with pricing for your API

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Customizing themes for the Developer Portal

You can configure an overrides.css file in a theme to customize the appearance of your Developer Portal home page.

Before you begin

You must have administrator access to complete this task.

Download the pre-supplied sample theme: banka_connect_theme_custom_css.zip.

Important: You are not permitted to include any IBM® API Connect code within any custom themes that you create.

Customizing the overrides.css file

1. Download the overrides.css file from the pre-supplied sample theme from \banka_connect_theme\css, and open it in your chosen editor.
2. Customize the overrides.css file by entering hex codes, and pixel values when applicable, in the correct positions for the following elements:

   Body of the home page

   ```
   body.page-home #columns.no-menu-bar {
     background-color: #fff;
   ```

   Header of the home page

   ```
   #page > header {
     background-color: #a0a0a0;
   ```

   Menu bar and border

   ```
   #menu-bar {
     background-color: #a0a0a0;
     border-bottom: 10px solid #ff9900;
   ```

   Border of the footer

   ```
   #page > footer, #page > footer .block-title {
     border-top: 10px solid #ff9900;
     background-color: #dee0e2;
   ```

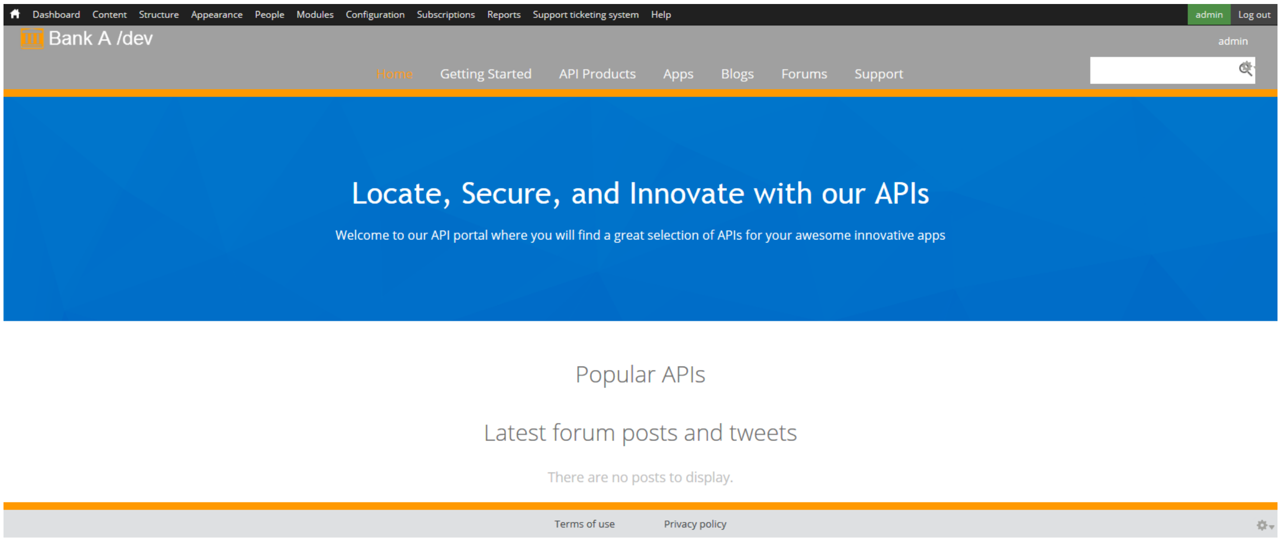
   Footer

```
#page > footer a {
  color: #454A4C;
```

3. After you have entered the hex codes to complete the elements, save the overrides.css file.

Implementing the customized overrides.css file

1. After you have finished customizing the overrides.css file, add it back to the compressed file in \banka_connect_theme\css.
2. Install the compressed file as a theme in the Developer Portal. For more information, see Installing additional themes.
3. Return to the Developer Portal home page by clicking the Home icon from the administrator dashboard. You can now see your custom theme.



Locate, Secure, and Innovate with our APIs

Welcome to our API portal where you will find a great selection of APIs for your awesome innovative apps

Popular APIs

Latest forum posts and tweets

There are no posts to display.

What you did in this tutorial

In this tutorial, you have completed the following tasks:

- Customized the elements in an overrides.css file with hex codes and pixel values.
- Installed the theme with the customized overrides.css file in the Developer Portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Installing a theme in the Developer Portal

You can install a custom theme to display on the Developer Portal, and then configure elements of that theme such as color and Welcome Banner image to match the new theme.

Before you begin

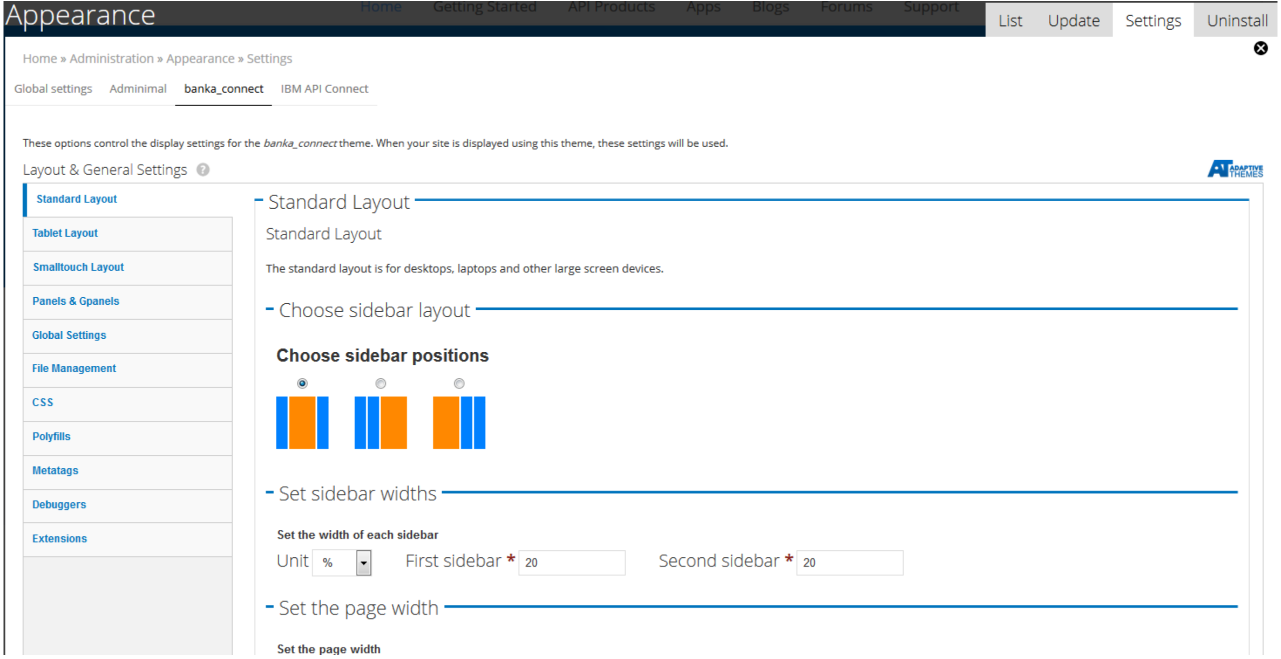
You must have administrator access to complete this task.

Download the pre-supplied sample theme, banka_connect_theme.zip.

Installing a theme in the Developer Portal

1. Log in to the Developer Portal as an administrator.
2. Select Appearance on the administrator dashboard.
3. Click Install new theme.
4. Click Browse... and select the banka_connect_theme.zip file that you downloaded previously, then click Install.
5. Click Enable newly added themes.

6. Under the DISABLED THEMES heading, you will find the theme you previously uploaded, click Enable and set default. Your theme is displayed under the ENABLED THEMES heading.
7. Click Settings for the new default theme. The Layout & General Settings page displays.



These options control the display settings for the *banka_connect* theme. When your site is displayed using this theme, these settings will be used.

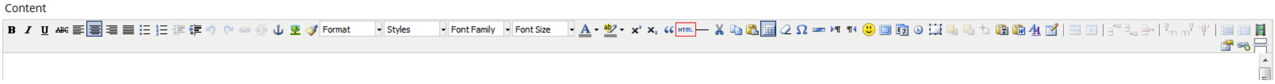
The standard layout is for desktops, laptops and other large screen devices.

Set the width of each sidebar

Unit % First sidebar * 20 Second sidebar * 20

Set the page width

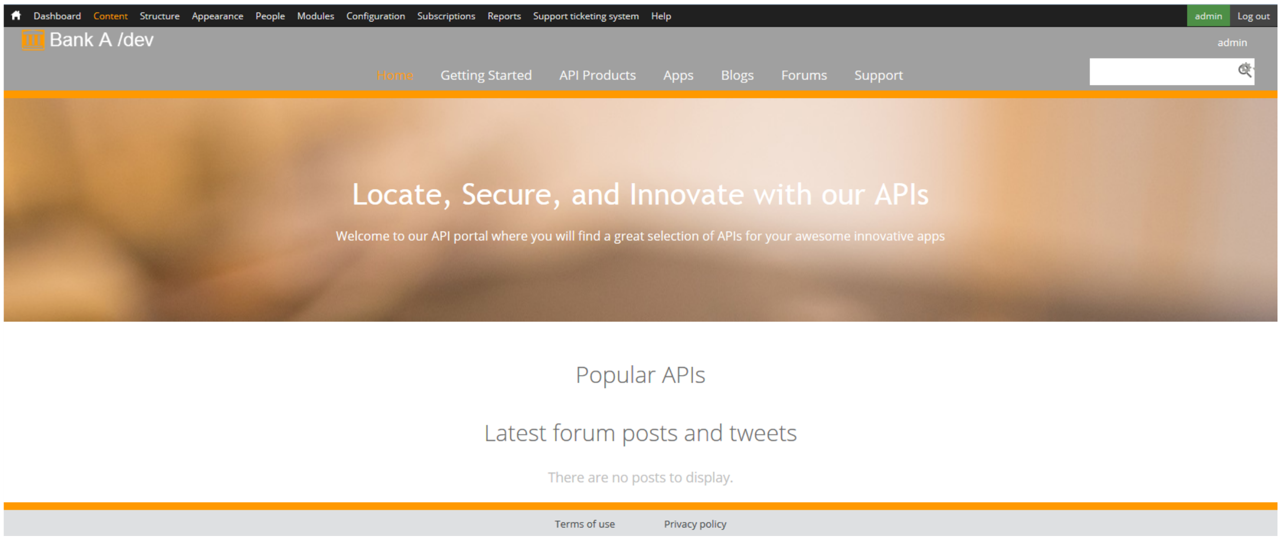
8. Expand Logo image settings, and clear the Use the default logo check box.
9. Click Browse..., and select the logo that you want to upload and display for your Developer Portal
10. Click Save configuration.
11. Return to the Developer Portal home page by clicking the Home icon on the administrator dashboard. You can now see your most recent customizations.
12. To reset the region settings for some of the blocks, click Structure on the administrator dashboard, then click Blocks.
13. Select Sidebar first from the region drop-down list for the Support block. The Support block must be the only block that is specified in the Sidebar first region.
14. Select None from the region drop-down list for the Navigation and User login blocks.
15. Click Save blocks.
16. To change the central banner from default, click Content on the administrator dashboard.
17. Select the Blocks tab on the upper right of the page.
18. Under the Title heading, find Welcome Banner and click edit.
19. Under the Image heading, click Remove to remove the default Welcome Banner image.
20. Choose the banka_connect_hero.png image from the pre-supplied sample theme.
21. After you have selected the image, click Upload.
22. Under the Content heading, to change the Content text to fit the new theme and banner, click the Edit HTML Source icon in the content editor.



23. Modify the text color, alignment and content to fit the new theme by entering the corresponding HTML into the HTML Source Editor, for example:

```
<h1 ><span >Locate, Secure, and Innovate with our APIs</span></h1><p ><span >Welcome to our API
portal where you will find a great selection of APIs for your awesome innovative apps</span></p><p
>&nbsp;</p>
```

24. Click Save.
25. Return to the Developer Portal home page by clicking the Home icon from the administrator dashboard. You can now see your customizations.



Locate, Secure, and Innovate with our APIs

Welcome to our API portal where you will find a great selection of APIs for your awesome innovative apps

Popular APIs

Latest forum posts and tweets

There are no posts to display.

What you did in this tutorial

In this tutorial, you have completed the following tasks:

- Installing a custom theme to the Developer Portal.
- Enabling that theme in the Developer Portal.
- Reconfiguring the blocks and Welcome Banner images for the Developer Portal home page.
- Editing the text of the Developer Portal site to match the new theme.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Displaying recent blog posts in the Developer Portal

You can display information for your five most recent blog posts on the front page of the Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

You can display the following information for your recent blog posts on the front page of the Developer Portal:

- Summary of the blog
- Title
- Author
- Date when it was published

By displaying your five most recent blog posts, you can maintain the communication of important or useful information easily with users of the Developer Portal.

Procedure

1. To enable the module that is required to display the blog posts:
   a. On the administrator dashboard, click Modules.
   b. Locate and enable the views_ui module, then click Save configuration.
2. To create the view and block where your blog posts are displayed:
   a. On the administrator dashboard, click Structure > Add new view.

   b. Enter a name for your view in the View name text field.
   c. For the Show, of type, and sorted by drop down lists, selecting the following options:
      - Content
      - Blog entry
      - Newest first
   d. Ensure that the Create a page check-box is not selected.
   e. Select the Create a block check-box.
   f. Enter a title for your block in the Block title text field.
   g. In the Display format section, ensure that the type of unformatted list that is selected is teasers.
   h. Ensure that without links and without comments are selected for the resulting two drop-down lists.
   i. Click Save & exit.
3. To display the view and block that you have created:
   a. On the administrator dashboard, click Structure > Pages, then click Edit for the welcome page.
   b. In the welcome window, click Content in the list on the left.
   c. For the section of the front page that you would like to display your blog posts on, click the edit icon ⚙, then click Add content.
   d. Click Miscellaneous, then select the block that you have created from the list.
   e. Optional: If you want to add a title that is different to the one that is specified in the block, select the Override title check box and enter a title in the adjacent text field.
   f. Click Finish, then click Update and save.
      Note: To improve performance in the Developer Portal, you can disable the Views UI module after you have completed the tutorial.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Using the Portal Delegated User Registry

You can use the Portal Delegated User Registry to increase the number of options that are available to you for account and security management from within the Developer Portal.

About this tutorial

This tutorial takes you through setting up the following three user registry options for the Developer Portal:

Portal Delegated User Registry

Setting the Portal Delegated User Registry in the API Manager UI means that the user management is delegated to the Developer Portal. Therefore new user accounts are created in the local Developer Portal database, also known as the local user registry. For more information, see Selecting the Portal Delegated User Registry.

Third-party authentication provider credentials

Enabling third-party authentication provider credentials, such as Facebook and Google, reduces the number of authentication credentials that a user of the Developer Portal needs. For more information, see Using third-party authentication provider credentials to access the Developer Portal.

LDAP user registry

Configuring LDAP means that the Developer Portal can authenticate users against an existing LDAP user registry. For more information, see Configuring the Developer Portal to use an LDAP user registry.

Each of these three options can be used separately, or in any combination. However, setting up the Portal Delegated User Registry is a prerequisite for both the third-party authentication option and the LDAP user registry option.
Note: This tutorial shows you how to configure the Developer Portal to authenticate users against an existing LDAP user registry. If you want to allow new users to be added to the LDAP user registry, see Tutorial: Configuring writable LDAP in the Developer Portal.
The following example takes you through setting up all three user registry options, giving new users the ability to access the Developer Portal by using LDAP credentials, third-party credentials, or by creating new accounts in the local user registry. In addition, the example shows you how to enable administrator approval for all new accounts, and how to modify the approval email template.

Before you begin

The following prerequisites are required before starting the tutorial:

- You must be an owner of a Provider Organization in the API Manager UI. For more information, see Creating a provider organization account.

- You must have a Catalog without a Developer Portal site. If you do have a Catalog that you want to work with that has a Developer Portal site, the only account that the Developer Portal site can have is the administrator account.

You will also need the following prerequisites if you want to complete the third-party authentication and LDAP sections:

- You must have an account with a third-party authentication provider. For more information on the third-party authentication providers that you can log in to the Developer Portal with, and how to obtain the necessary Application settings to enable their use, see Using third-party authentication provider credentials to access the Developer Portal.

- You must have an LDAP server and its details.

Enabling the Portal Delegated User Registry

1. In API Manager, if you have not previously pinned the UI navigation pane then click the Navigate to icon ≡. The API Manager UI navigation pane opens. To pin the UI Navigation pane, click the Pin menu icon 📌.
2. Click Dashboard in the Navigation pane, then click the Catalog for which you want to enable the use of third-party authentication provider credentials.
3. Click Settings > Portal.
4. Select the IBM Developer Portal radio button to enable the Developer Portal site.
5. Enter the URL of your Developer Portal site.
6. In the User Registration and Invitation section, select Portal Delegated User Registry from the User Registry drop-down list. The *catalog_name* is the name of the Catalog that you are working in and applying the registry settings to.
7. V5.0.3 and earlier Ensure that Self-service onboarding is set to the on position.
   Important: Self-service onboarding must be on to complete the tutorial.
8. V5.0.4 + Ensure that Developers can invite collaborators and assign the following roles is set to the on position.
   Important: Developers can invite collaborators and assign the following roles must be on to complete the tutorial.
9. Click Save.
10. 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. When the link is active and you have accessed it, you can change the password of this administrator account.

Enabling the use of third-party authentication provider credentials to access the Developer Portal

In the Developer Portal, log in as the administrator to complete the following steps:

1. Ensure that the HybridAuth module is enabled. For more information, see Disabling modules, and enable the module if necessary. When you enable Portal Delegated User Registry in the API Manager UI, the module is enabled automatically.
2. On the administrator dashboard, click Configuration > People > HybridAuth.
3. From the list of authentication providers that are displayed, click the check box for the authentication provider that has the authentication credentials that you want to use, then click Settings. The Application settings tab contains text fields that must be filled with specific values. Information on obtaining the specific values for each authentication provider can be found on their Application settings tabs.
4. Fill in the required fields for the authentication provider.
5. Click Save configuration.

Configuring your LDAP user registry in the Developer Portal

1. Enable LDAP configuration in the Developer Portal by clicking Modules on the administrator dashboard. Search for, and enable, the following modules:
   - LDAP Servers
   - LDAP User Module
   - LDAP Authentication

   Then, click Save configuration.
2. Click Configuration > People > LDAP Configuration
3. To configure your LDAP registry settings:
   a. Click Settings, then select the check box in the Require HTTPS on Credential Pages.
   b. Click Save configuration.
4. To configure your LDAP registry server:
   a. Click Servers > Add LDAP Server Configuration.
   b. In the Connection settings settings, enter values for the following fields:

        - For Machine name for this server configuration, enter `MyLDAPServer`
        - For Name, enter LDAP Server 1
    c. Select the check box for Enabled.
    d. Select your type of LDAP server from the LDAP Server Type drop-down list.
    e. Enter the IP address or domain name of your LDAP server in the LDAP server text field.
    f. Enter you port number in the LDAP port text field.
    g. In the Binding Method section, select Anonymous Bind for search, then Bind with User Credentials.
    h. In the LDAP User to Drupal User Relationship section, enter values for the following fields:
        - Base DNs for LDAP users, groups, and other entries
        - AuthName attribute
    i. Click Add.
5. To configure the LDAP Authentication:
    a. Click the Authentication tab, then in the LDAP Authentication Settings section, select the check box for your LDAP server that is found under Authentication LDAP Server Configurations.
    Note: The Mixed mode radio button must be selected as it enables you to use third-party authentication provider credentials in addition to your LDAP credentials.
    b. In the User Login Interface section, enter values for the following fields:
        - For the Username Description Text, enter `LDAP Username`
        - For the Password Description Text, enter `LDAP Password`
    c. In the Email section, select the check box for Don't show an email field on user forms, then click Save.

Change the account settings to enable approval for all new accounts including third-party authentication providers

1. On the administrator dashboard, click Configuration > People > Account settings.
2. In the Registration and cancellation section, select the Visitors, but administrator approval is required check box.
3. To enable the Require e-mail verification when a visitor creates an account function, select the adjacent check box.
4. Click Save configuration.

Modifying the Developer Portal email templates

1. On the administrator dashboard, click Configuration > People > Account settings.
2. In the E-mails section, modify the content of the Welcome (awaiting approval) with the following text into the corresponding fields:

    Subject

    Welcome *user_name*. Your *site_name* account is pending approval

    Body

    Thank you very much for signing up to *site_name*. Your account is currently pending approval, and you will receive e-mail confirmation upon its approval.

3. Click Save configuration.

Results

You have enabled the Developer Portal to authenticate users against a local user registry, a third-party authentication provider, or an LDAP user registry. With this scenario, if a user does not exist in an external user registry, either LDAP or the third-party provider, their account is created in the local Developer Portal database.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Configuring writable LDAP in the Developer Portal

You can configure writable LDAP in the Developer Portal if you already have an existing LDAP and want to include more users. In this tutorial, you create a new writable OpenLDAP server on Ubuntu 16.04, but you can use your own LDAP server if you prefer.

Before you begin

You must have administrator access to complete this task.

Portal Delegated User Registry must be selected in the API Manager UI. For more information, see Portal Delegated User Registry.

Note: The use of writable LDAP requires the storage of credentials to an LDAP server that has write access in the Developer Portal database.

About this task

By configuring writable LDAP, your organization can control its user management in the Developer Portal in a similar way to any of its existing LDAP servers, and enable users to perform self service sign-up. This tutorial is an extension of Tutorial: Using the Portal Delegated User Registry, which shows you how to configure the Developer Portal to authenticate users against an existing LDAP user registry. Configuring the Developer Portal for writable LDAP is more complex, and some of the configuration depends on your specific LDAP user registry. Step 1 of this tutorial involves setting up a test LDAP server on a separate Ubuntu Linux system. If you already have an LDAP server you can skip Step 1, but you will need to obtain any information that is relevant for the tutorial from your LDAP administrator.

Important: The steps might vary if you use other LDAP providers on other platforms.

Procedure

1. To set up a test LDAP server on an Ubuntu 16.04 system:
   a. Run the following command:

      ```
      sudo apt install slapd ldap-utils
      ```

      Note: Keep a note of the password that you enter for this step, as it is needed later in the tutorial.
   b. Run the following command:

      ```
      sudo dpkg-reconfigure slapd
      ```

      When prompted for further information, set the `dns domain name` to

      ```
      my.company.com
      ```

      and the `orgname` to

      ```
      myorg
      ```

      Accept the default entries for everything else.
   c. Create a file called portal.ldif, which contains the following information, editing the userPassword to be that set previously, and setting mail to an accessible email address:

      ```
      dn: ou=People,dc=my,dc=company,dc=com
      objectClass: organizationalUnit
      ou: People

      dn: ou=Groups,dc=my,dc=company,dc=com
      objectClass: organizationalUnit
      ou: Groups

      dn: cn=miners,ou=Groups,dc=my,dc=company,dc=com
      objectClass: posixGroup
      cn: miners
      gidNumber: 5000

      dn: uid=ld3,ou=People,dc=my,dc=company,dc=com
      objectClass: inetOrgPerson
      uid: ld3
      sn: d3
      givenName: ld3
      cn: ld3 d3
      displayName: ld3 d3
      userPassword: 7iron-hide
      mail: testuser@ibm.com
      ```

   d. Run the following command (you will need to enter the password from Step 1.a):

      ```
      ldapadd -x -D cn=admin,dc=my,dc=company,dc=com -W -f portal.ldif
      ```

      If the LDAP server set up has been successful, you will see the following output:

      ```
      Enter LDAP Password:
      adding new entry "ou=People,dc=my,dc=company,dc=com"
      adding new entry "ou=Groups,dc=my,dc=company,dc=com"
      ```

```
adding new entry "cn=miners,ou=Groups,dc=my,dc=company,dc=com"
adding new entry "uid=john,ou=People,dc=my,dc=company,dc=com"
```

2. To enable the necessary modules that are required to configure writable LDAP, complete the following steps in the Developer Portal:
   a. On the administrator dashboard, click Modules.
   b. Search for and enable the following modules:
      - ldap_authentication
      - ldap_servers
      - ldap_user

   c. Click Save configuration.
3. To configure your writable LDAP:
   a. On the administrator dashboard, click Configuration > People > LDAP Configuration > 2. Servers.
   b. Select the Add LDAP Server Configuration tab.
   c. In the Connection settings section, enter values for the Machine name for this server configuration and Name text fields.
   d. Select the Enabled check box.
   e. Select Open LDAP from the LDAP Server Type drop-down list.
   f. Enter the domain or IP address of your LDAP server in the LDAP Server text-field.
   g. In the Binding Method section, select Service Account Bind.
   h. Enter a value for the DN for non-anonymous search as **`cn=admin,dc=my,dc=company,dc=com`**, then set your LDAP administrator password that was used to set up the OpenLDAP server for Password for non-anonymous search.
   i. In the LDAP User to Drupal User Relationship section, enter values for the following fields:
      - Base DNs for LDAP users, groups, and other entries: **`ou=People,dc=my,dc=company,dc=com`**
      - AuthName attribute: **`uid`**
      - Email attribute: **`mail`**
      - Expression for user DN: **`uid=%username,%basedn`**

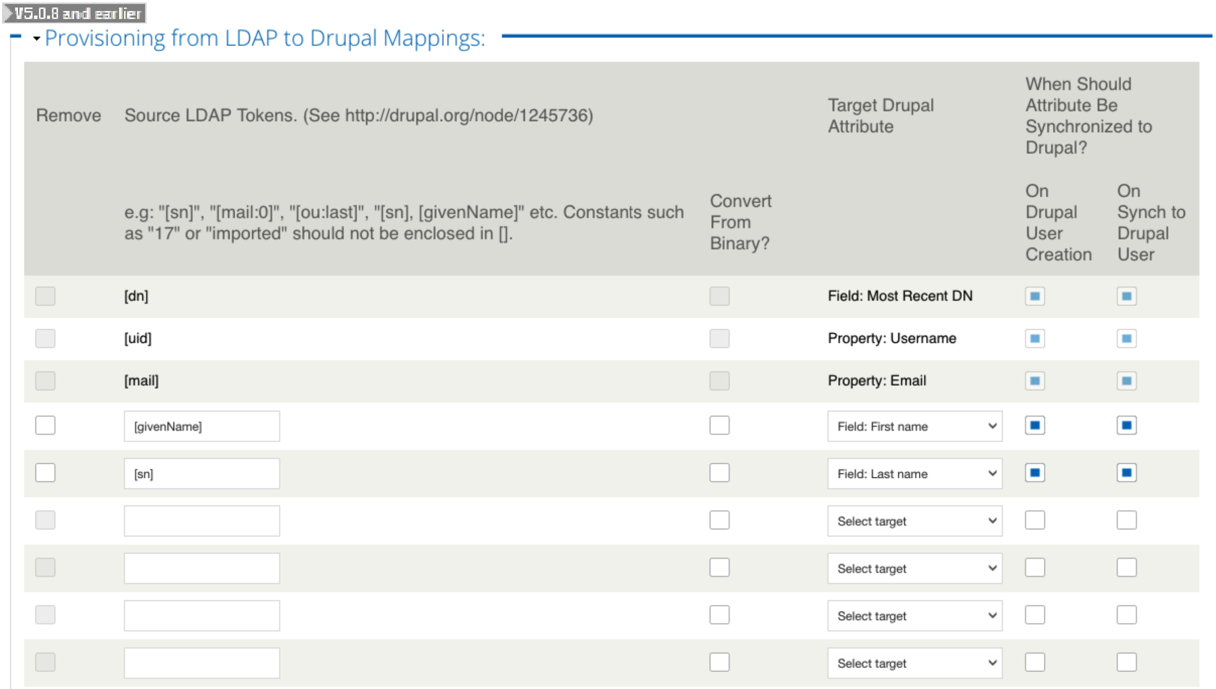
      Details for some of these fields can be found in your .ldif file, and each text-field is accompanied by help text in the Developer Portal UI.
   j. Click Add
      You are directed to the List tab for the 2. Servers tab.
   k. Under Operations, click test.
   l. Optional: To specify the purpose of the test further, populate any relevant fields in the Test LDAP Server Configuration page.
   m. Click Test.
      The output of the test is located at the top of the page.
4. To configure the User settings in your writable LDAP:
   a. On the administrator dashboard, click Configuration > People > LDAP Configuration > 3. User.
   b. In the Basic Provisioning to Drupal Account Settings section, select your LDAP server.
   c. For Drupal Account Provisioning Events, ensure that the following check boxes are selected:
      - Create or Synch to Drupal user on successful authentication with LDAP credentials
      - Create or Synch to Drupal user anytime a Drupal user account is created or updated

   d. For Existing Drupal User Account Conflict, ensure that the Associate Drupal account with the LDAP entry radio button is selected.
   e. For Application of Drupal Account settings to LDAP Authenticated Users, ensure that the Account creation settings at /admin/config/people/accounts/settings do not affect "LDAP Associated" Drupal accounts radio button is selected.
   f. For Action to perform on drupal account that no longer have a corresponding LDAP entry, ensure that the Perform no action, but email list of orphaned accounts radio button is selected.
   g. Click Save.
5. To provision your writable LDAP to Drupal mappings:
   a. In the Provisioning from LDAP to Drupal Mappings section, enter the following information into the table according to which version of API Connect you are using:
      V5.0.8 and earlier

Table 1. Provisioning from LDAP to Drupal Mappings for Version 5.0.8.3 and earlier

Source LDAP Tokens	Target Drupal Attribute
`[dn]`	`Field: Most Recent DN`
`[uid]`	`Property: Username`
`[mail]`	`Property: Email`
`[givenName]`	`Field: First name`
`[sn]`	`Field: Last name`

V5.0.8 and earlier



Provisioning from LDAP to Drupal Mappings:

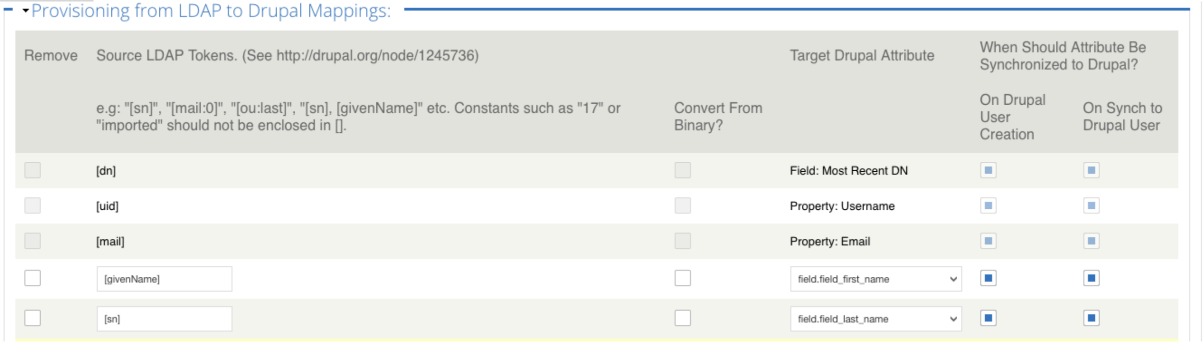
Remove	Source LDAP Tokens. (See http://drupal.org/node/1245736)		Target Drupal Attribute	When Should Attribute Be Synchronized to Drupal?	
	e.g: "[sn]", "[mail:0]", "[ou:last]", "[sn], [givenName]" etc. Constants such as "17" or "imported" should not be enclosed in [].	Convert From Binary?		On Drupal User Creation	On Synch to Drupal User
☐	[dn]	☐	Field: Most Recent DN	☑	☑
☐	[uid]	☐	Property: Username	☑	☑
☐	[mail]	☐	Property: Email	☑	☑
☐	[givenName]	☐	Field: First name	☑	☑
☐	[sn]	☐	Field: Last name	☑	☑
☐		☐	Select target	☐	☐
☐		☐	Select target	☐	☐
☐		☐	Select target	☐	☐
☐		☐	Select target	☐	☐

V5.0.8 +

Table 2. Provisioning from LDAP to Drupal Mappings for Version 5.0.8.4 onwards

Source LDAP Tokens	Target Drupal Attribute
`[dn]`	`Field: Most Recent DN`
`[uid]`	`Property: Username`
`[mail]`	`Property: Email`
`[givenName]`	`field.field_first_name`
`[sn]`	`field.field_last_name`

V5.0.8 +



Provisioning from LDAP to Drupal Mappings:

Remove	Source LDAP Tokens. (See http://drupal.org/node/1245736)		Target Drupal Attribute	When Should Attribute Be Synchronized to Drupal?	
	e.g: "[sn]", "[mail:0]", "[ou:last]", "[sn], [givenName]" etc. Constants such as "17" or "imported" should not be enclosed in [].	Convert From Binary?		On Drupal User Creation	On Synch to Drupal User
☐	[dn]	☐	Field: Most Recent DN	☑	☑
☐	[uid]	☐	Property: Username	☑	☑
☐	[mail]	☐	Property: Email	☑	☑
☐	[givenName]	☐	field.field_first_name	☑	☑
☐	[sn]	☐	field.field_last_name	☑	☑

b. Click Save.

6. To provision your Drupal to writable LDAP Mappings:

   a. In the Basic Provisioning to LDAP Settings section, select the radio button for your LDAP server.

   b. For LDAP Entry Provisioning Events, select both of the following check boxes:

      - Create or Synch to LDAP entry when a Drupal account is created or updated
      - Create or Synch to LDAP entry when a user authenticates

   c. In the Provisioning from Drupal to LDAP Mappings section, enter the following information into the table according to which version of API Connect you are using:

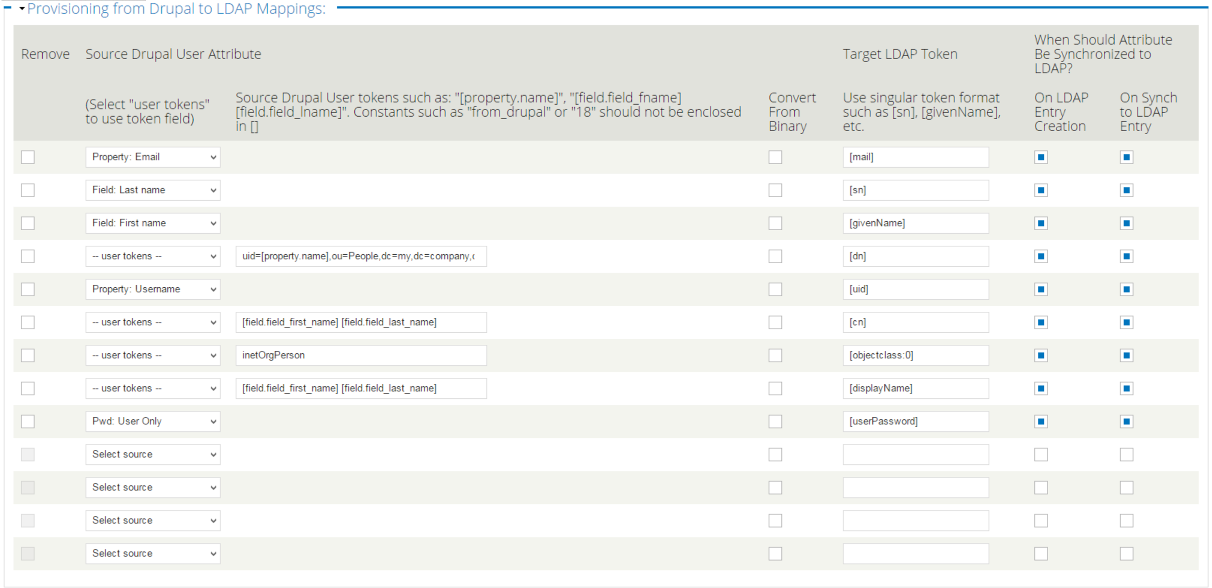
V5.0.8 and earlier

Table 3. Provisioning from Drupal to LDAP Mappings for Version 5.0.8.3 and earlier

Source Drupal User Attribute	Source Drupal User tokens	Target LDAP Token
`Property: Email`	n/a	`[mail]`
`Field: Last name`	n/a	`[sn]`
`Field: First name`	n/a	`[givenName]`
`-- user tokens --`	`uid=[property.name],ou=People,dc=my,dc=company,dc=com`	`[dn]`
`Property: Username`	n/a	`[uid]`
`-- user tokens --`	`[field.field_first_name] [field.field_last_name]`	`[cn]`
`-- user tokens --`	`inetOrgPerson`	`[objectclass:0]`
`-- user tokens --`	`[field.field_first_name] [field.field_last_name]`	`[displayName]`

Source Drupal User Attribute	Source Drupal User tokens	Target LDAP Token
`Pwd: User Only`	n/a	`[userPassword]`

V5.0.8 and earlier



▾Provisioning from Drupal to LDAP Mappings:

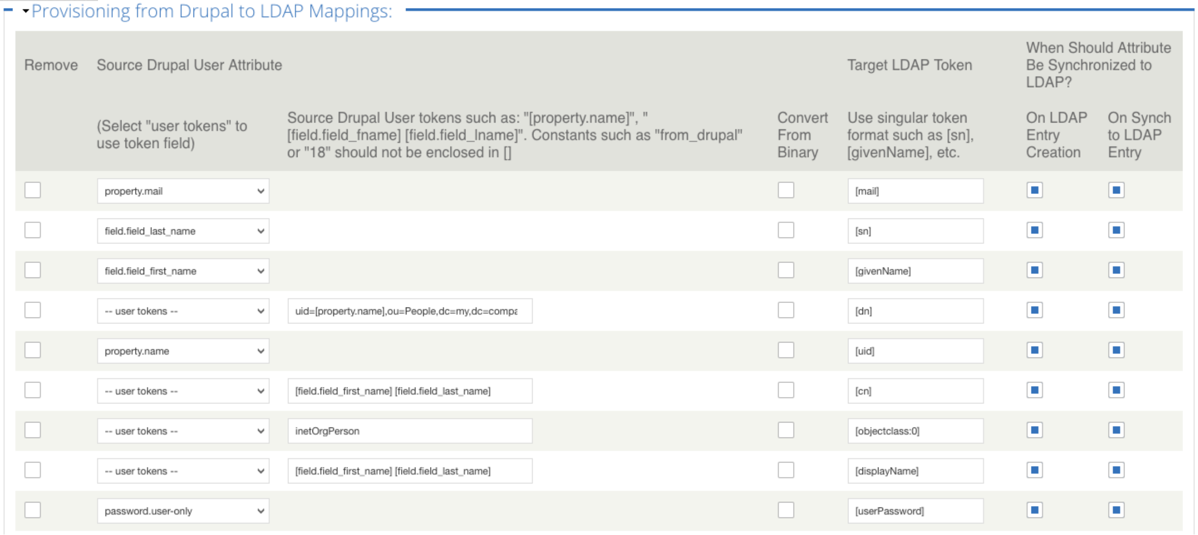
Remove	Source Drupal User Attribute			Target LDAP Token	When Should Attribute Be Synchronized to LDAP?	
	(Select "user tokens" to use token field)	Source Drupal User tokens such as: "[property.name]", "[field.field_fname] [field.field_lname]". Constants such as "from_drupal" or "18" should not be enclosed in []	Convert From Binary	Use singular token format such as [sn], [givenName], etc.	On LDAP Entry Creation	On Synch to LDAP Entry
□	Property: Email		□	[mail]	☑	☑
□	Field: Last name		□	[sn]	☑	☑
□	Field: First name		□	[givenName]	☑	☑
□	-- user tokens --	uid=[property.name],ou=People,dc=my,dc=company,c	□	[dn]	☑	☑
□	Property: Username		□	[uid]	☑	☑
□	-- user tokens --	[field.field_first_name] [field.field_last_name]	□	[cn]	☑	☑
□	-- user tokens --	inetOrgPerson	□	[objectclass:0]	☑	☑
□	-- user tokens --	[field.field_first_name] [field.field_last_name]	□	[displayName]	☑	☑
□	Pwd: User Only		□	[userPassword]	☑	☑
□	Select source		□		□	□
□	Select source		□		□	□
□	Select source		□		□	□
□	Select source		□		□	□

V5.0.8 +

Table 4. Provisioning from Drupal to LDAP Mappings for Version 5.0.8.4 onwards

Source Drupal User Attribute	Source Drupal User tokens	Target LDAP Token
`property.mail`	n/a	`[mail]`
`field.field_last_name`	n/a	`[sn]`
`field.field_first_name`	n/a	`[givenName]`
`-- user tokens --`	`uid=[property.name],ou=People,dc=my,dc=company,dc=com`	`[dn]`
`property.name`	n/a	`[uid]`
`-- user tokens --`	`[field.field_first_name] [field.field_last_name]`	`[cn]`
`-- user tokens --`	`inetOrgPerson`	`[objectclass:0]`
`-- user tokens --`	`[field.field_first_name] [field.field_last_name]`	`[displayName]`
`password.user-only`	n/a	`[userPassword]`

V5.0.8 +



▾Provisioning from Drupal to LDAP Mappings:

Remove	Source Drupal User Attribute			Target LDAP Token	When Should Attribute Be Synchronized to LDAP?	
	(Select "user tokens" to use token field)	Source Drupal User tokens such as: "[property.name]", "[field.field_fname] [field.field_lname]". Constants such as "from_drupal" or "18" should not be enclosed in []	Convert From Binary	Use singular token format such as [sn], [givenName], etc.	On LDAP Entry Creation	On Synch to LDAP Entry
□	property.mail		□	[mail]	☑	☑
□	field.field_last_name		□	[sn]	☑	☑
□	field.field_first_name		□	[givenName]	☑	☑
□	-- user tokens --	uid=[property.name],ou=People,dc=my,dc=compa	□	[dn]	☑	☑
□	property.name		□	[uid]	☑	☑
□	-- user tokens --	[field.field_first_name] [field.field_last_name]	□	[cn]	☑	☑
□	-- user tokens --	inetOrgPerson	□	[objectclass:0]	☑	☑
□	-- user tokens --	[field.field_first_name] [field.field_last_name]	□	[displayName]	☑	☑
□	password.user-only		□	[userPassword]	☑	☑

Note: If you run out of fields, click Save to add more fields to the table.

d. Click Save.

7. To configure the authentication for your writable LDAP, select the Authentication tab:
   a. In the Allowable Authentications section, select the Only LDAP Authentication is allowed except for user 1 radio-button.
   b. Select the check box for your LDAP server.
   c. Click Save.

Results

You can now login to the Developer Portal as the John user that you created when you set up the OpenLDAP server. When new user accounts are created, they are provisioned to the LDAP registry.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Authenticating users in the Developer Portal by using an external server

Use a custom module to enable an external server to authenticate users in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

Portal Delegated User Registry must be selected in the API Manager UI. For more information, see Portal Delegated User Registry. When Portal Delegated User Registry is selected, the HybridAuth module is automatically enabled in the Developer Portal.

Download the custom module sampleauth.zip and extract the files.

About this task

In this tutorial, you can enable the Developer Portal to authenticate users by using an external server. This is done by configuring a custom module and installing it in the Developer Portal.

The file that you download is a custom module that contains multiple functions that can be configured, and any further specifications that you want to define, to set up authentication with an external server.

For more information, see Creating custom modules to extend functionality.

Procedure

To configure the custom module to enable an external server to manage authentication for the Developer Portal, edit the sampleauth.module file as shown in the following steps:

1. In the `sampleauth_register_user` function, set the target URL to call when creating new users by replacing the following example target with the correct endpoint:

   ```
   $url = 'https://example.com/users/register';
   ```

2. In the `sampleauth_authenticate` function, set the target URL to call to check the authentication credentials of the user by replacing the following example target with the correct endpoint:

   ```
   $url = 'https://example.com/users/auth';
   ```

3. Optional: Edit the curl SSL settings in the `_sampleauth_json_http_request` function as you require.
4. Optional: To provide further information for others that might use your module, configure the fields in the `sampleauth_help` function to your specification.
5. After you have finished configuring the module, click Save, and then compress the module files into a .zip file.

To install the configured custom module in the Developer Portal:

6. On the administrator dashboard, click Modules
7. Click Install new modules.
8. You can enter a path to the module in the Install from a URL field. Alternatively, you can upload a module under the Upload a module or theme archive to install heading.
9. Click Install.

To enable the module:

10. On the administrator dashboard, click Modules.
11. Search for and enable the custom module.
12. Click Save configuration.

Results

You configured and installed a custom module that enables users of the Developer Portal to authenticate by using an external server.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Set up custom workbench moderation for blogs

If a blog is created, you can configure workbench moderation to ensure that the blog is reviewed before it is published in the Developer Portal.

Before you begin

You must have administrator access to complete this task.

About this task

By completing the tutorial, you can create specific transition states for blog entries to follow in the Developer Portal before they are published, and can configure permissions on the Content Author role so that they can moderate blog publishing.

Procedure

1. To enable the workbench moderation module:
    a. On the administrator dashboard, click Modules.
    b. In the Search text field, enter the following modules:
        - `Workbench`
        - V5.0.5+ `Drafty`

       The Workbench, V5.0.5+ Drafty, and Workbench Moderation modules appear.
    c. Select the toggle buttons for the Workbench, Drafty, and Workbench Moderation modules so that a green tick is displayed.
    d. Click Save configuration.
       Note: You must rebuild your content access permissions after completing this step.
2. To enable revisions and moderation for Blog entry:
    a. On the administrator dashboard, click Structure > Content types.
       The Content types window is displayed.
    b. Click edit for blogs.
    c. Click Publishing options and select the check boxes for Create new revision, then Enable moderation of revisions.
       Important: You must ensure that the Published check box is not selected.
    d. From the Default moderation state drop-down list, select Draft.
    e. Click Save content type.
    f. On the administrator dashboard, click Reports > Status report
    g. In the Node Access Permissions section, click Rebuild permissions, then click Rebuild permissions again.
3. To configure workbench states and transitions for Blog entry:
    a. On the administrator dashboard, click Configuration > Workbench > Workbench Moderation.
    b. In the New state text field, enter Final Review, and enter Final mandatory review before the blog can be published in the Description text field.
       Note: You must reconfigure Views after completing this step.
    c. Click Save.
    d. Select the Transitions tab.
    e. Select the check box in the adjacent to Publish, then click Save.
    f. In the Transition Name text field, enter Submit for Final Review.
    g. Select Needs Review from the From drop-down list, and Final review from the To drop-down list.
    h. Click Save.
    i. In the Transition Name text field, enter Reject from Final Review.
    j. Select Final Review from the From drop-down list, and Draft from the To drop-down list.
    k. Click Save.
    l. In the Transition Name text field, enter Publish.
    m. Select Final Review from the From drop-down list, and Published from the To drop-down list.
    n. Click Save.

4. To set up the permissions for Blog entry for the Content Author role:
    a. On the administrator dashboard, click People > Permissions > Roles.
    b. Click edit permissions for the Content Author role.
    c. Ensure that the following check boxes are selected:

       Node permissions
       - View content revisions
       - Revert content revisions
       - Blog entry: Edit own content
       - Blog entry: Create new content

       Workbench Moderation permissions
       - View all unpublished content
       - View the moderation messages on a node
       - View moderation history
       - Use "My drafts" workbench tab
       - Use "Needs review" workbench tab
       - Moderate all content from Needs Review to Draft
       - Moderate all content from Final Review to Draft
       - Moderate all content from Needs Review to Final Review
       - Moderate all content from Final Review to Published

    d. Click Save permissions.
5. To specify the permissions that specific roles require to access the workbench dashboard:
    a. On the administrator dashboard, click Configuration > Workbench > Workbench Moderation > Check permissions
    b. Select content author from the Drupal role drop-down list.
    c. Select Moderate content from the Moderation task drop-down list.
    d. Select the Blog entry check box.
    e. Click Check permissions.
       A list of recommended actions are displayed at the top of the Workbench Moderation window. By following the recommended actions, you can assign the permissions that are required for a role to access the workbench dashboard.

Results

You have created specific transition states to ensure that a blog does not get published unless it passes the final review, and that a Content Author can moderate the transition states of the blog.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Adding and exporting custom fields

You can add custom fields to users in the Developer Portal. In addition, the custom fields can be exported.

Before you begin

You must have administrator access to complete this task.

About this task

You can add multiple various types of data to user records, such as department names, accounting information, addresses, and additional custom information that you might want to collect from your API consumers. The custom data that is added to user records can be collected and used by exporting.

By completing the tutorial, you can add custom fields to a user, and export the user records and the field through a custom View.

Any custom fields that are added to a user will remain in the Developer Portal database.

Procedure

1. To add custom fields for a user:

    a. On the administrator dashboard, click Configuration > People > Account settings > Manage fields.
    b. Enter Department number in the Label text field for Add new field.
    c. Select Integer from the Field type drop-down list.
    d. Ensure that Text field is selected from the Widget list.
    e. Click Save, then Save field settings.
    You have added a custom field to your user record.

2. To enable the modules that allow user records and fields to be exported through a custom View:
    a. On the administrator dashboard, click Modules.
    b. Identify and enable the Views Data Export and Views UI modules.
    c. Click Save Configuration.
3. To add a View to export user records and fields:
    a. On the administrator dashboard, click Structure > Views.
    b. Click Add new view.
    c. Enter Employee information in the View name text field.
    d. In the Show section, select Users from the first drop-down list.
    e. Click Continue & edit, then click Save.
4. To configure a page that can export data:
    a. Under Displays in the Employee information (User) view, adjacent to Page, click +Add > Data export.
    b. In the Data export settings section, click No path is set.
    c. In the text field, enter employee_information, then click Apply.
    d. In the Data export settings section, click None, select the Page check box, then click Apply.
    e. Click Save.
5. To configure the format in which your data is exported, and the fields that are exported:
    a. In the Format section, click CSV file.
    b. Select the XLS file radio button, click Apply,
    c. Select the Provide as file check box, then click Apply.
    d. In the Fields section, click Add.
    e. Select User from the Filter drop-down list, then select the check boxes for the following fields from the list of available fields:
        - User: Department number
        - User: E-mail
        - User: First name
        - User: Last name
    f. Click Apply (all displays), then click Apply (all displays) for each field.
    g. Click Save.
6. To export the user data that you have specified:
    a. Enter the data export URL that you have specified.
    For example:

```
https://www.portal_host_name/employee_information
```

Results

You have exported the specified user data as an .xls file.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorials for configuring Rules in the Developer Portal

You can configure Rules to perform specific actions when they are triggered by specific events 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:

- An application is created
- A new subscription is created
- To request feedback two weeks after subscribing to a Product
- Application credentials are reset
- An application is unsubscribed from a Plan
- A Product is superseded

You can also configure Rules to send a welcome email to new Developer Organization Owners, send an email to subscribers of a Plan when a Product is deprecated, and synchronize application credentials with an external server.
Note: There are a large number of events that you can create and configure Rules for, and the tutorials cover select examples. For a complete list of Rules events, see List of Rules events.
Rules can be imported and exported to forgo the process of Rule creation and configuration, and to store local versions of the Rules. For more information, see Importing and exporting Rules.

- V5.0.6 + **Tutorial: Sending notifications when applications are created**
  You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if a member of their organization creates an application.
- V5.0.6 + **Tutorial: Sending notifications when creating new Plan subscriptions**
  You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if a member of their organization subscribes one of its applications to a new Plan.
- V5.0.6 + **Tutorial: Requesting subscription feedback after two weeks**
  You can configure Rules in the Developer Portal to request feedback from Developer Organization Owners two weeks after subscribing to a Product.
- V5.0.6 + **Tutorial: Sending notifications when application credentials reset**
  You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if an application client ID is reset.
- V5.0.6 + **Tutorial: Sending notifications when an application is unsubscribed from a Plan**
  You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if an application is unsubscribed from a Plan.
- V5.0.6 + **Tutorial: Sending notifications when a Product is superseded**
  You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners that own applications that are subscribed to a Product, that the Product is being superseded.
- V5.0.6 + **Tutorial: Sending an email to new Developer Organization Owners**
  You can configure Rules in the Developer Portal to send an email to new Developer Organization Owners.
- V5.0.6 + **Tutorial: Sending notifications when a Product is deprecated**
  You can configure Rules in the Developer Portal to send an email to Developer Organization members that are associated with an application that is subscribed to a Plan in a Product, when the Product is deprecated.
- V5.0.6 + **Tutorial: Synchronizing your application credentials with an external server**
  You can create and configure a rule to synchronize the client credentials for applications from the Developer Portal server to an external server.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when applications are created

You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if a member of their organization creates an application.

Before you begin

You must have administrator access to complete this task.

About this task

Configuring Rules about application creation is useful for Developer Organizations that have multiple members. The email that is sent to the Developer Organization Owner can contain a link to the newly created application.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter register_application.
   d. From the React on event drop-down list, select After creating a new application, then click Save.
2. To add a condition to avoid activating the rule if the Developer Organization Owner creates an application:

   a. In the Conditions section, click Add condition.
   b. From the Select the condition to add drop-down list, select Data comparison.
   c. From the Data selector drop-down list, select application-devorg-owner, then click Continue.
   d. In the Operator section, select equals from the Value drop-down list.
   e. In the Data value section, enter [site:current-user] in the Value text field.
   f. Select the Negate check box, then click Save.

3. To send emails to Developer Organization Owners as an action for your Rule:
   a. In the Actions section, click Add action.
   b. From the Select the action to add drop-down list, select Send mail.
   c. In the To section, enter [application-devorg-owner:value] in the Value text box.
   d. In the Subject section, enter An application has been created in your Developer Organization in the Value text box.
   e. In the Message section, enter the following into the Value text box:

```
<p> An application '[application-name:value]' has been created in the [application-devorg-
name:value] Developer Organization.</p>

<p>It can be accessed here:<br/>
[site:url]/node/[application-nid:value]</p>

<p>Application description:<br/>
[application-description:value]</p>
```

   Then, click Save.

Results

You have created and configured a Rule to send an email to Developer Organization Owners every time an application is created in their Developer Organization.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_new_application" : {
    "LABEL" : "new_application",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n", "application" ],
    "ON" : { "application_create" : [] },
    "IF" : [
      { "NOT data_is" : {
          "data" : [ "application-devorg-owner" ],
          "value" : "[site:current-user]"
        }
      }
    ],
    "DO" : [
      { "mail" : {
          "to" : "[application-devorg-owner:value]",
          "subject" : "A new application \u0027[application-name:value]\u0027 has been created in your 
developer organization",
          "message" : "\u003Cp\u003EA new application \u0027[application-name:value]\u0027 has been 
created in the [application-devorg-name:value] developer 
organization.\u003C\/p\u003E\r\n\r\n\u003Cp\u003EIt can be accessed 
here:\u003Cbr\/\u003E\r\n[site:url]\/node\/[application-
nid:value]\u003C\/p\u003E\r\n\r\n\u003Cp\u003EApplication description:\u003Cbr\/\u003E\r\n[application-
description:value]\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when creating new Plan subscriptions

You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if a member of their organization subscribes one of its applications to a new Plan.

Before you begin

You must have administrator access to complete this task.

About this task

Configuring Rules about application subscription is useful for Developer Organizations that have multiple members, to help Developer Organization Owners stay informed about the activity of their members.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter register_application.
   d. From the React on event drop-down list, select After subscribing to a plan, then click Save.
2. To send emails to Developer Organization Owners as an action for your Rule:
   a. In the Actions section, click Add action.
   b. From the Select the action to add drop-down list, select Send mail.
   c. In the To section, enter [application-devorg-owner:value] in the Value text box.
   d. In the Subject section, enter Application [application-name:value] has subscribed to the [plan-name:value] Plan in [product-name:value] in the Value text box.
   e. In the Message section, enter the following into the Value text box:

   ```
   <p>Application '[application-name:value]' from the [application-devorg-name:value] Developer
   Organization has subscribed to the [plan-name:value] plan in [product-name:value].</p>
   <p></p>
   <p>The application can be accessed here for more information:<br/>
   [site:url]node/[application-nid:value]</p>
   ```

   Then, click Save.

Results

You have created and configured a Rule to send an email to Developer Organization Owners every time a member of the Developer Organization subscribes an application to a new Plan.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_new_subscription" : {
    "LABEL" : "new subscription",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules_i18n", "rules", "application" ],
    "ON" : { "application_subscribe" : [] },
    "DO" : [
      { "mail" : {
          "to" : [ "application-devorg-owner" ],
          "subject" : "Application [application-name:value] has subscribed to the [plan-name:value] plan
in [product-name:value] ",
          "message" : "\u003Cp\u003EApplication \u0027[application-name:value]\u0027 from the
[application-devorg-name:value] developer organization has subscribed to the [plan-name:value] plan in
[product-name:value].\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EThe application can
be accessed here for more information:\u003Cbr\/\u003E\r\n[site:url]node\/[application-
nid:value]\u003C\/p\u003E\r\n",
          "language" : [ "" ]
        }
      }
```

```
      ]
    }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Requesting subscription feedback after two weeks

You can configure Rules in the Developer Portal to request feedback from Developer Organization Owners two weeks after subscribing to a Product.

Before you begin

You must have administrator access to complete this task.

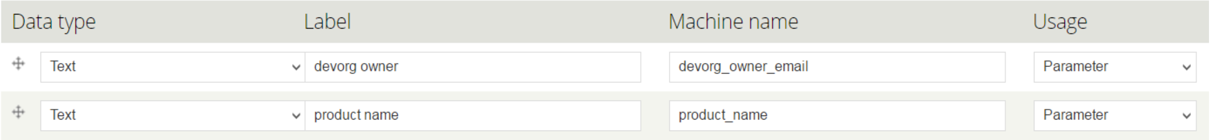
About this task

By configuring a Rule in the Developer Portal, you can schedule an email to be sent two weeks after a subscription to a Product is made. The contents of the email can be configured to contain information such as links to forums, support ticket systems, and contact details to social media sites.

Requesting feedback two weeks after subscribing to a Product can be used to solicit feedback on APIs, and inquire about whether their needs are met with the Plans that they have, and establish communication channels with API consumers.

Procedure

1. To create a reusable Rules component to send an email:
   a. On the administrator dashboard, click Conifguration > Workflow > Rules > Components > Add new component.
   b. From the Component plugin drop-down list, select Action set, then click Continue.
   c. In the Name text field, enter Email devorg for feedback.
   d. In the Variables table, enter the following values in the appropriate fields:



Data type	Label	Machine name	Usage
Text	devorg owner	devorg_owner_email	Parameter
Text	product name	product_name	Parameter

   Then, click Continue.
   e. Click Add action.
   f. From the Select the action to add drop-down list, select Send mail from the System section.
   g. In the To section, enter [devorg-owner-email:value] in the Value text box.
   h. In the Subject section, enter Please provide feedback on our APIs and Plans in the Value text box.
   i. In the Message section, enter the following into the Value text box:

```
<p>It has been 2 weeks since you subscribed to the [product-name:value] product.</p>
<p>We hope you have been enjoying using our APIs and Plans.</p>
<p>If you have any feedback on them we would love to hear from you.</p>
<p></p>
<p>Please post in our forums or raise a support ticket if you need any assistance:<br/>
[site:url]support</p>
<p></p>
<p>Thanks!</p>
```

   Then, click Save.
2. To create a Rule to trigger the component two weeks after the Product is subscribed:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter plan_feedback.
   d. From the React on event drop-down list, select After subscribing to a plan, then click Save.
   e. Click Add action
   f. From the Select the action to add drop-down list, click Schedule component evaluation.

g. In the Component section, select the component that you created in step 1 from the Value drop-down list.
h. Click Continue.
i. In the Scheduled evaluation date section, enter + 2 weeks in the Value text box.
j. In the Identifier section, enter get subscription feedback after 2 weeks in the Value text box.
k. In the devorg owner section, enter [application-devorg-owner:value] into the Value text box.
l. In the product name section, enter [product-name:value] in the Value text box, then click Save.

Results

You have created and configured a Rule that sends an email that requests feedback, two weeks after a subscription to a Product is made.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following component:

```
{ "rules_email_devorg_owner_for_feedback" : {
    "LABEL" : "email devorg owner for feedback",
    "PLUGIN" : "action set",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n" ],
    "USES VARIABLES" : {
      "devorg_owner_email" : { "label" : "devorg owner", "type" : "text" },
      "product_name" : { "label" : "product name", "type" : "text" }
    },
    "ACTION SET" : [
      { "mail" : {
          "to" : "[devorg-owner-email:value]",
          "subject" : "Please provide feedback on our APIs",
          "message" : "\u003Cp\u003EIt has been 2 weeks since you subscribed to the [product-name:value] 
product.\u003C\/p\u003E\r\n\u003Cp\u003EWe hope you have been enjoying using our 
APIs.\u003C\/p\u003E\r\n\u003Cp\u003EIf you have any feedback on them we would love to hear from 
you.\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EPlease post in our forums or raise a 
support ticket if you need any 
assistance:\u003Cbr\/\u003E\r\n[site:url]support\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003
Cp\u003EThanks!\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

Then, import the following Rule:

```
{ "rules_plan_feedback" : {
    "LABEL" : "plan_feedback",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_scheduler", "application" ],
    "ON" : { "application_subscribe" : [] },
    "DO" : [
        { "schedule" : {
            "component" :
            "rules_email_devorg_owner_for_feedback",            "date" : "+ 2 weeks",
            "identifier" : "get subscription feedback after 2 weeks",
            "param_devorg_owner_email" : "[application-devorg-owner:value]",
            "param_product_name" : "[product-name:value]"
          }
        }
      ]
    }
  }
```

into the Developer Portal and configure it.
For more information on importing Rules, see Importing and exporting Rules.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when application credentials reset

You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if an application client ID is reset.

Before you begin

You must have administrator access to complete this task.

About this task

By configuring a Rule, Developer Organization Owners can remain informed when changes occur for applications in their organization.
Note: In addition to notifying Developer Organization Owners when client IDs are reset, you can configure a rule to notify them when client secrets are reset.

Procedure

1. To create the Rule:
    a. On the administrator dashboard, click Configuration > Workflow > Rules.
    b. Click Add new rule.
    c. In the Name text field, enter clientid_reset.
    d. From the React on event drop-down list, select After resetting the application client ID, then click Save.
2. To add a condition to avoid activating the rule if the Developer Organization Owner changes the application credentials:
    a. In the Conditions section, click Add condition.
    b. From the Select the condition to add drop-down list, select Data comparison.
    c. From the Data selector drop-down list, select application-devorg-owner, then click Continue.
    d. In the Operator section, select equals from the Value drop-down list.
    e. In the Data value section, enter [site:current-user] in the Value text field.
    f. Select the Negate check box, then click Save.
3. To send emails to Developer Organization Owners as an action for your Rule:
    a. In the Actions section, click Add action.
    b. From the Select the action to add drop-down list, select Send mail.
    c. In the To section, enter [application-devorg-owner:value] in the Value text box.
    d. In the Subject section, enter Application [application-name:value] Client ID reset in the Value text box.
    e. In the Message section, enter the following into the Value text box:

    ```
    <p>This is to let you know that the Client ID for application [application-name:value]
    belonging to Developer Organization [application-devorg-name:value] has been reset.</p>
    <p></p>
    <p>The application can be accessed here:<br/>
    [site:url]node/[application-nid:value]</p>
    ```

    Then, click Save.

Results

You have created and configured a Rule that sends an email to Developer Organization Owners every time the credentials of an application are reset.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_clientid_reset" : {
    "LABEL" : "clientid_reset",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n", "application" ],
    "ON" : { "application_clientid_reset" : [] },
    "IF" : [
      { "NOT data_is" : {
          "data" : [ "application-devorg-owner" ],
          "value" : "[site:current-user]"
        }
```

```
      }
    ],
    "DO" : [
      { "mail" : {
          "to" : "[application-devorg-owner:value]",
          "subject" : "Application [application-name:value] Client ID reset",
          "message" : "\u003Cp\u003EThis is to let you know that the Client ID for application 
[application-name:value] belonging to developer organization [application-devorg-name:value] has been 
reset.\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EThe application can be accessed 
here:\u003Cbr\/\u003E\r\n[site:url]node\/[application-nid:value]\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when an application is unsubscribed from a Plan

You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners if an application is unsubscribed from a Plan.

Before you begin

You must have administrator access to complete this task.

About this task

By configuring a Rule, Developer Organization Owners can remain informed when changes occur for application subscriptions to Plans in their organization.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter app_unsubscribed.
   d. From the React on event drop-down list, select After unsubscribing from a plan, then click Save.
2. To send emails to Developer Organization Owners as an action for your Rule:
   a. In the Actions section, click Add action.
   b. From the Select the action to add drop-down list, select Send mail.
   c. In the To section, enter [application-devorg-owner:value] in the Value text box.
   d. In the Subject section, enter Application [application-name:value] has been unsubscribed to the [plan-name:value] Plan in [product-name:value] in the Value text box.
   e. In the Message section, enter the following into the Value text box:

      ```
      <p>Application '[application-name:value]' from the [application-devorg-name:value] Developer 
      Organization has been unsubscribed from the [plan-name:value] plan in [product-name:value].
      </p>
      <p></p>
      <p>The application can be accessed here for more information:<br/>
      [site:url]node/[application-nid:value]</p>
      ```

      Then, click Save.

Results

You have created and configured a Rule to send an email to Developer Organization Owners every time an application is unsubscribed from a Plan.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_app_unsubscribed" : {
    "LABEL" : "app unsubscribed",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n", "application" ],
    "ON" : { "application_unsubscribe" : [] },
    "DO" : [
      { "mail" : {
          "to" : "[application-devorg-owner:value]",
          "subject" : "Application [application-name:value] has been unsubscribed from the [plan-
name:value] plan in [product-name:value]",
          "message" : "\u003Cp\u003EApplication \u0027[application-name:value]\u0027 from the
[application-devorg-name:value] developer organization has been unsubscribed from the [plan-name:value]
plan in [product-name:value].\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EThe
application can be accessed here for more information:\u003Cbr\/\u003E\r\n[site:url]node\/[application-
nid:value]\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when a Product is superseded

You can configure Rules in the Developer Portal to send an email to notify Developer Organization Owners that own applications that are subscribed to a Product, that the Product is being superseded.

Before you begin

You must have administrator access to complete this task.

About this task

By configuring a Rule, Developer Organization Owners can remain informed when changes occur for application that they own that are subscribed to Products in an organization.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter product_superseded.
   d. From the React on event drop-down list, select After superseding an existing product, then click Save.
2. To send emails to application owners as an action for your Rule:
   a. Click Add action.
   b. From the Select the action to add drop-down list, select Email product subscription owners.
   c. In the Data selector section, click Switch to the data selector mode, and select product_nid from the drop-down list.
   d. In the Subject section, enter Product [product-name:value] has been superseded in the Value text box.
   e. In the Message section, enter the following into the Value text box:

```
<p>Product [product-name:value] has been superseded.</p>
<p></p>
<p>For more information please visit [site:url]node/[product-nid:value]</p>
```

Then, click Save.

Results

You have created and configured a Rule to send an email to Developer Organization Owners that own applications every time a Product, that their application is subscribed to, is superseded.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_product_superseded" : {
    "LABEL" : "product_superseded",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n", "product" ],
    "ON" : { "product_supersede" : [] },
    "DO" : [
      { "product_rules_action_email_subscribers_owner" : {
          "product_nid" : "[product-nid:value]",
          "subject" : "Product [product-name:value] has been superseded",
          "message" : "\u003Cp\u003EProduct [product-name:value] has been
superseded.\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EFor more information please
visit [site:url]node\/[product-nid:value]\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending an email to new Developer Organization Owners

You can configure Rules in the Developer Portal to send an email to new Developer Organization Owners.

Before you begin

You must have administrator access to complete this task.

About this task

By configuring a Rule, you can send new Developer Organization Owners a welcome email that might contain useful information such as links to forums or features that they might find useful.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter welcome_email.
   d. From the React on event drop-down list, select After creating a new developer organization, then click Save.
2. To send emails to new Developer Organization Owners as an action for your Rule:
   a. In the Actions section, click Add action.

b. From the Select the action to add drop-down list, select Send mail in the System section.
c. In the To section, enter [devorg-owner-email:value] in the Value text box.
d. In the Subject section, enter Welcome to our API Portal in the Value text box.
e. In the Message section, enter the following into the Value text box:

```
<p>Welcome to [site:name]!</p>
<p></p>
<p>We hope you find all the APIs you are looking for here: [site:url]product</p>
<p></p>
<p>You can register applications here: [site:url]application/new</p>
<p></p>
<p>Please do check out our forums and participate in the community: [site:url]forum</p>
<p></p>
<p>If you need assistance then help can be found here: [site:url]help</p>
<p></p>
<p>Regards,<br/>
The [site:name] team</p>
```

Then, click Save.

Results

You have created and configured a Rule to send an email every time a new Developer Organization Owners is created.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_welcome_email" : {
    "LABEL" : "welcome email",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_i18n", "devorg" ],
    "ON" : { "devorg_create" : [] },
    "DO" : [
      { "mail" : {
          "to" : "[devorg-owner-email:value]",
          "subject" : "Welcome to our API Portal",
          "message" : "\u003Cp\u003EWelcome to 
[site:name]!\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EWe hope you find all the 
APIs you are looking for here: 
[site:url]\/product\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EYou can register 
applications here: 
[site:url]application\/new\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EPlease do 
check out our forums and participate in the community: 
[site:url]\/forum\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EIf you need assistance 
then help can be found here: 
[site:url]\/help\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003ERegards,\u003Cbr\/\u003E
\r\nThe [site:name] team\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Sending notifications when a Product is deprecated

You can configure Rules in the Developer Portal to send an email to Developer Organization members that are associated with an application that is subscribed to a Plan in a Product, when the Product is deprecated.

Before you begin

You must have administrator access to complete this task.

About this task

By configuring a Rule, you can keep all consumers informed of changes that occur with a Product.. The email is sent to all members of a Developer Organization that own an application that has subscribed to a Plan from the Product that is deprecated.

Procedure

1. To create the Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter product_deprecated.
   d. From the React on event drop-down list, select After deprecating a product, then click Save.
2. To send emails to Developer Organization members as an action for your Rule:
   a. Click Add action.
   b. From the Select the action to add drop-down list, select Email product subscription members.
   c. In the Product NID section, use the Data Selector to select product-nid.
   d. In the Subject section, enter Product [product-name:value] has been deprecated in the Value text box.
   e. In the Message section, enter the following into the Value text box:

   ```
   <p>Product [product-name:value] has been deprecated.</p>
   <p></p>
   <p>For more information please visit [site:url]node/[product-nid:value]</p>
   ```

   Then, click Save.

Results

You have created and configured a Rule to send an email to Developer Organization members that are associated with an application that is subscribed to a Plan in a Product, that the Product is deprecated.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_product_deprecated" : {
    "LABEL" : "product_deprecated",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules_i18n", "rules", "product" ],
    "ON" : { "product_deprecate" : [] },
    "DO" : [
      { "product_rules_action_email_subscribers_members" : {
          "product_nid" : [ "product_nid" ],
          "subject" : "Product [product-name:value] has been deprecated",
          "message" : "\u003Cp\u003EProduct [product-name:value] has been 
deprecated.\u003C\/p\u003E\r\n\u003Cp\u003E\u003C\/p\u003E\r\n\u003Cp\u003EFor more information please 
visit [site:url]node\/[product-nid:value]\u003C\/p\u003E",
          "language" : [ "" ]
        }
      }
    ]
  }
 }
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Synchronizing your application credentials with an external server

You can create and configure a rule to synchronize the client credentials for applications from the Developer Portal server to an external server.

Before you begin

You must have administrator access to complete this task.

About this task

Note: The steps in the tutorial are non-specific. The following specifications must be configured in the steps depending on the specification of your server:

- Modifying target API endpoints
- Modifying payloads
- Adding authentication
- Any other necessary changes

With Rules, you can synchronize application credentials between servers and update any authorization header for the target endpoint through the Developer Portal UI.
To synchronize application credentials between servers, Rules must be created that are triggered by the following events:

- After creating a new application. For more information, see Tutorial: Sending notifications when applications are created
- After deleting an application
- After deleting application credentials
- After resetting the application client ID. For more information, see Tutorial: Sending notifications when application credentials reset
- After resetting the application client Secret
- After saving new application credentials
- After updating an existing application
- After updating existing application credentials

Procedure

1. To create an event trigger for your Rule:
   a. On the administrator dashboard, click Configuration > Workflow > Rules.
   b. Click Add new rule.
   c. In the Name text field, enter register_application.
   d. From the React on event drop-down list, select After creating a new application, then click Save.
2. To send emails to Developer Organization Owners as an action for your Rule:
   a. In the Actions section, click Add action.
   b. From the Select the action to add drop-down list, select Request HTTP Data.
   c. In the URL section, enter your target REST endpoint in the Value text field.
   d. Add any Headers that are required by your REST endpoint such as Authorization header.
   e. In the Method section, select POST from the Value drop-down list.
   f. In the Data section, enter the following into the Value text field:

```
{
'id': '[application-id:value]',
'name': '[application-name:value]',
'credentials': [application-credentials:value]
}
```

3. Repeat steps 1 and 2 for the following events:
   - After creating a new application. For more information, see Tutorial: Sending notifications when applications are created
   - After deleting an application
   - After deleting application credentials
   - After resetting the application client ID. For more information, see Tutorial: Sending notifications when application credentials reset
   - After resetting the application client Secret
   - After saving new application credentials
   - After updating an existing application

- After updating existing application credentials

Note: The target API endpoint varies depending on the event.

Results

You have created and configured a Rule to synchronize client credentials for applications from the Developer Portal server to an external server.

What to do next

If you do not want to follow the steps that are involved in creating the Rule, you can import the following Rule into the Developer Portal and configure it:

```
{ "rules_sync_newapp" : {
    "LABEL" : "sync_newapp",
    "PLUGIN" : "reaction rule",
    "OWNER" : "rules",
    "REQUIRES" : [ "rules", "rules_http_client", "application" ],
    "ON" : { "application_create" : [] },
    "DO" : [
      { "request_url" : {
          "USING" : {
            "url" : "http:\/\/myserver.com\/api\/new",
            "method" : "POST",
            "data" : "{\r\n\u0027id\u0027: \u0027[application-id:value]\u0027,\r\n\u0027name\u0027:
\u0027[application-name:value]\u0027,\r\n\u0027credentials\u0027: [application-credentials:value]\r\n}"
          },
          "PROVIDE" : { "http_response" : { "http_response" : "HTTP data" } }
        }
      }
    ]
  }
}
```

For more information on importing Rules, see Importing and exporting Rules.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

V5.0.8 +

Tutorial: Configuring a custom sort order view for APIs in the Developer Portal

You can configure the Developer Portal so that a list of APIs is displayed in a custom sort order.

Before you begin

You must have administrator access to complete this task.

About this task

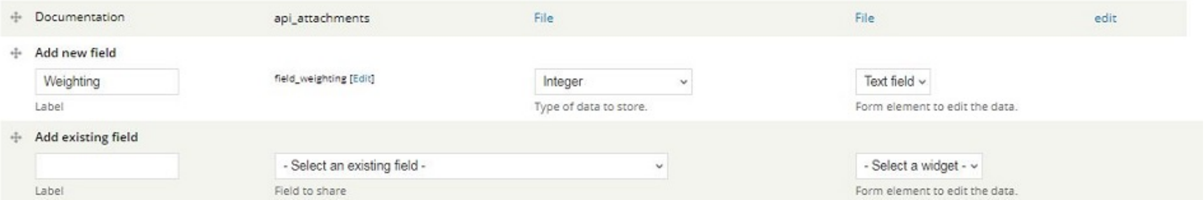
You can use the Views module to fetch content from your Developer Portal site, such as lists of APIs or Products, and present the content to users in different formats. By adding a custom field to your APIs or Products, you can then create a view that displays your APIs or Products in a custom sort order. For more information about views, see Using the Views module in the Developer Portal.

In this tutorial, you enable the Views module, create a custom integer field to use for applying weighting to APIs, and create a new view that uses the custom field to display APIs in a weighted sort order. The same set of instructions can be applied to creating a custom sort order view for other lists of items, such as Products.

Procedure

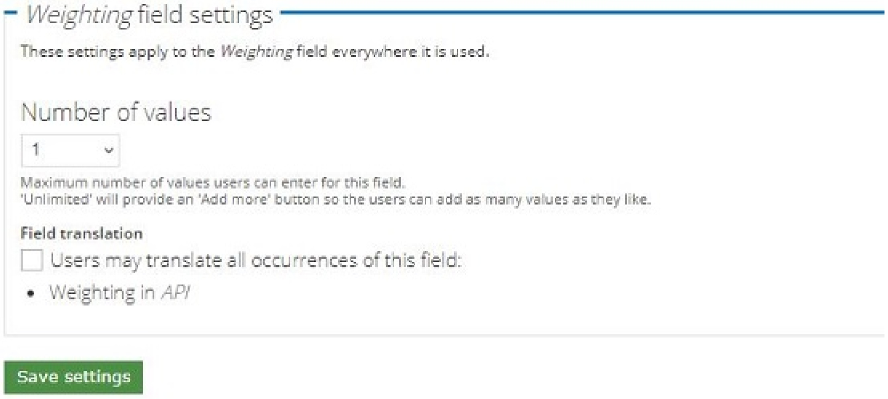
Create a custom sort order view for displaying a list of APIs by completing the following steps.

1. Log in to the Developer Portal as an administrator.
2. Enable the Views UI (views_ui) module.
    a. On the administrator dashboard, click Modules.
    b. Enter `views` into the search field of the Modules pane to find the Views UI module, and set the module to ON.
    c. Click Save configuration.
3. Add a custom integer field to the API content type.
    a. On the administrative dashboard, click Structure > Content types > API > Manage fields.
    b. Scroll down to the Add new field section, and enter `Weighting` into the Label field.



Documentation	api_attachments	File	File	edit
Add new field				
Weighting Label	field_weighting [Edit]	Integer Type of data to store.	Text field Form element to edit the data.	
Add existing field				
Label	- Select an existing field - Field to share		- Select a widget - Form element to edit the data.	

    c. Select Integer for the type of data to store, and click Save > Save field settings.
    The Edit tab for the newly created Weighting field displays.
    d. In the Weighting field settings section, ensure the Number of values is set to `1`.



Weighting field settings

These settings apply to the *Weighting* field everywhere it is used.

Number of values

1

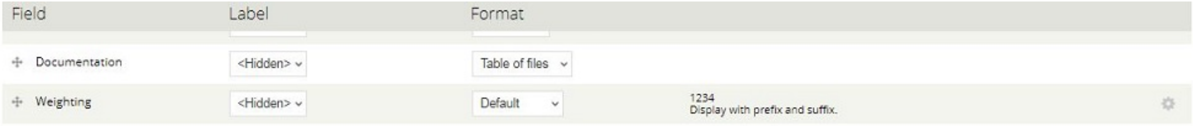
Maximum number of values users can enter for this field.
'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

Field translation

□ Users may translate all occurrences of this field:

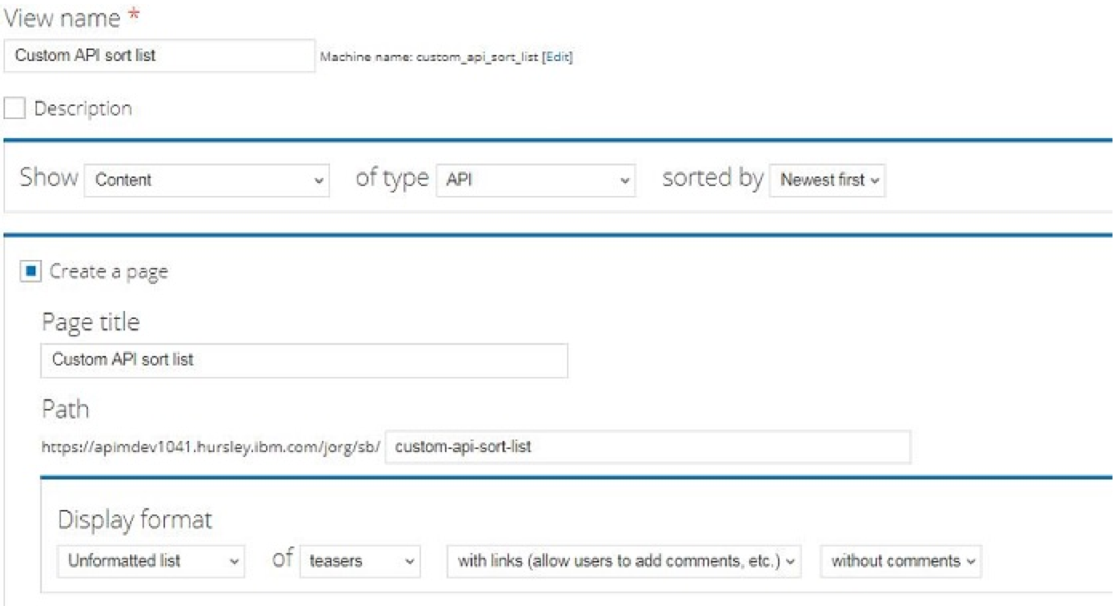
- Weighting in *API*

    e. Click Save settings.
    f. Select the Manage display tab, and scroll down to the newly created Weighting field.
    g. Select <Hidden> for the Label column, so the new field is not displayed in any view.



Field	Label	Format		
Documentation	<Hidden>	Table of files		
Weighting	<Hidden>	Default	1234 Display with prefix and suffix.	

    h. Click Save.
4. Create a new view for the custom sort list of APIs.
    a. On the administrator dashboard, click Structure > Add new view.
    b. Enter `Custom API sort list` in the View name field.
    c. For the Show, of type, and sorted by drop-down lists, select the following options:
        - Content
        - API
        - Newest first
    d. Ensure that the Create a page check-box is selected.
    e. In the Display format section, ensure that an Unformatted list of teasers is selected.
    f. Ensure that with links (allow users to add comments, etc.) and without comments are selected for the next two drop-down lists.



View name *

Custom API sort list Machine name: custom_api_sort_list [Edit]

☐ Description

Show Content of type API sorted by Newest first

☑ Create a page

Page title

Custom API sort list

Path

https://apimdev1041.hursley.ibm.com/jorg/sb/ custom-api-sort-list

Display format

Unformatted list of teasers with links (allow users to add comments, etc.) without comments

   g. Click Continue & edit.
   h. In the Sort criteria section, click Add.
      The Add sort criteria page opens. Here you will change the sort criteria to use the new Weighting field.
   i. Enter `weighting` into the Search field, and select the Content: Weighting (field_weighting) field that you created in Step 3.
   j. Click Apply (all displays), then select Sort ascending. Click Apply (all displays) again.
   k. Click the Content: Post date (desc) sort criteria, and click Remove so that only the weighting sort criteria is used.
   l. In the Page settings section, click No menu and select Normal menu entry.
   m. Provide a title and description for the menu item entry.
   n. Select Main menu for the available menu, and click Apply.
   o. Click Save.
      The new custom API sort view is saved and you can scroll down the page to see a preview of the view.

5. Add a weighting to each API.
   a. On the administrator dashboard, click Content, and in the Show only items where section select a type of API and click Filter.
      The list of available APIs on your site is displayed.
   b. For each API select edit, enter a single digit numerical value into the Weighting field, and click Save.
      In this way, you apply a custom sort order to each API, remembering that the list is sorted in an ascending order. So an API with a weighting of `1` is displayed at the top of the list.
   c. Click Finish, then click Update and save.
6. Click your new view, Custom API sort list, that is displayed in the main menu across the top of the site.
   The APIs on your site are shown in the custom sort order that you applied by using the Weighting field.

Results

You created a new custom sort order view for the APIs on your site.
Note: To improve performance in the Developer Portal, you can disable the Views UI module after you complete the tutorial.

What to do next

You can edit the view options of your new view to configure the Custom API sort list further. You can also create custom sort order views for other lists of items that you have on your site.

If you don't want to follow the steps from this tutorial to create a custom sort order view, you can import the following view into the Developer Portal and then configure it to your needs. You still need to manually create the custom field that is called Weighting, and add a sort order to each API on your site for the imported view to work.

To import a basic custom sort list view for APIs, complete the following steps:

1. Enable the Views UI (views_ui) module in the Developer Portal. See Step 2 for instructions.
2. Add a new Weighting field to the API content type. See Step 3 for instructions.
3. Add a weighting to each API on your site. See Step 5 for instructions.
4. Select and copy the contents of the following view:

```
$view = new view();
$view->name = 'custom_api_sort_list';
$view->description = '';
$view->tag = 'default';
```

```
$view->base_table = 'node';
$view->human_name = 'Custom API sort list';
$view->core = 7;
$view->api_version = '3.0';
$view->disabled = FALSE; /* Edit this to true to make a default view disabled initially */

/* Display: Master */
$handler = $view->new_display('default', 'Master', 'default');
$handler->display->display_options['title'] = 'Custom API sort list';
$handler->display->display_options['use_more_always'] = FALSE;
$handler->display->display_options['access']['type'] = 'perm';
$handler->display->display_options['cache']['type'] = 'none';
$handler->display->display_options['query']['type'] = 'views_query';
$handler->display->display_options['exposed_form']['type'] = 'basic';
$handler->display->display_options['pager']['type'] = 'full';
$handler->display->display_options['pager']['options']['items_per_page'] = '10';
$handler->display->display_options['style_plugin'] = 'default';
$handler->display->display_options['row_plugin'] = 'node';
/* Field: Content: Title */
$handler->display->display_options['fields']['title']['id'] = 'title';
$handler->display->display_options['fields']['title']['table'] = 'node';
$handler->display->display_options['fields']['title']['field'] = 'title';
$handler->display->display_options['fields']['title']['label'] = '';
$handler->display->display_options['fields']['title']['alter']['word_boundary'] = FALSE;
$handler->display->display_options['fields']['title']['alter']['ellipsis'] = FALSE;
/* Sort criterion: Content: Weighting (field_weighting) */
$handler->display->display_options['sorts']['field_weighting_value']['id'] =
'field_weighting_value';
$handler->display->display_options['sorts']['field_weighting_value']['table'] =
'field_data_field_weighting';
$handler->display->display_options['sorts']['field_weighting_value']['field'] =
'field_weighting_value';
/* Filter criterion: Content: Published */
$handler->display->display_options['filters']['status']['id'] = 'status';
$handler->display->display_options['filters']['status']['table'] = 'node';
$handler->display->display_options['filters']['status']['field'] = 'status';
$handler->display->display_options['filters']['status']['value'] = 1;
$handler->display->display_options['filters']['status']['group'] = 1;
$handler->display->display_options['filters']['status']['expose']['operator'] = FALSE;
/* Filter criterion: Content: Type */
$handler->display->display_options['filters']['type']['id'] = 'type';
$handler->display->display_options['filters']['type']['table'] = 'node';
$handler->display->display_options['filters']['type']['field'] = 'type';
$handler->display->display_options['filters']['type']['value'] = array(
  'api' => 'api',
);

/* Display: Page */
$handler = $view->new_display('page', 'Page', 'page');
$handler->display->display_options['path'] = 'custom-api-sort-list';
$handler->display->display_options['menu']['type'] = 'normal';
$handler->display->display_options['menu']['title'] = 'Custom API sort list';
$handler->display->display_options['menu']['description'] = 'List of APIs that have been sorted
into a custom order';
$handler->display->display_options['menu']['weight'] = '0';
$handler->display->display_options['menu']['name'] = 'main-menu';
$handler->display->display_options['menu']['context'] = 0;
$handler->display->display_options['menu']['context_only_inline'] = 0;
$translatables['custom_api_sort_list'] = array(
  t('Master'),
  t('Custom API sort list'),
  t('more'),
  t('Apply'),
  t('Reset'),
  t('Sort by'),
  t('Asc'),
  t('Desc'),
  t('Items per page'),
  t('- All -'),
  t('Offset'),
  t('« first'),
  t('‹ previous'),
  t('next ›'),
  t('last »'),
  t('Page'),
);
```

5. Click Structure on the administrator dashboard, then select Views. The Views page is displayed.
6. Click Import, and then paste the contents of the view that you copied into the Paste view code here box.

7. Click Import. The Custom API sort list view is now available on your site.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Tutorial: Synchronizing application credentials from the Developer Portal with an external server

You can synchronize application credentials from the Developer Portal with an external server.

Before you begin

You must have administrator access to complete this task.

Download the custom module: appcreds_sync.zip

About this task

In this tutorial, you can synchronize the application credentials from the Developer Portal with an external credentials server that you organization might already have. This is done by configuring a custom module and installing it in the Developer Portal.

The file that you have downloaded is a custom module that contains multiple methods that can be configured with your REST endpoints, and any further specifications that you want to define, to synchronize your Developer Portal application credentials with an external credential server.

For more information, see Creating custom modules to extend functionality.

Procedure

To configure the custom module to synchronize application credentials:

1. In the appcreds_sync.module file, replace `example.com` with your REST endpoint for the following method types:
   - `http://example.com/app/create`
   - `http://example.com/app/delete`
   - `http://example.com/app/creds/create`
   - `http://example.com/app/creds/update`
   - `http://example.com/app/creds/delete`
   - `http://example.com/app/clientid/reset`
   - `http://example.com/app/clientsecret/reset`
2. Optional: Replace `username` and `password` with your username and password in the `_appcreds_sync_json_http_request` function:

   ```
   ('username' . ':' . 'password')
   ```

   Note: You can also configure the SSL settings if you need to.
3. Optional: To provide further information for your module for others that might use your module, configure the fields in the `appcreds_sync_help` function to your specification.
4. After you have finished configuring the module, click Save, and zip up the module.

To install the configured custom module in the Developer Portal:

5. On the administrator dashboard, click Modules
6. Click Install new modules.
7. You can enter a path to the module in the Install from a URL field. Alternatively, you can upload a module under the Upload a module or theme archive to install heading.
8. Click Install.

To enable the module:

9. On the administrator dashboard, click Modules.
10. Search for and enable the custom module.

11. Click Save configuration.

Results

You have configured a custom module that synchronizes application credentials from the Developer Portal, installed that module in the Developer Portal, and enabled it. No further action or configuration is required.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Using the V5 Developer Portal REST APIs

V5 Developer Portal REST APIs are used to gain access to the content in a Catalog that would be exposed to an API consumer.

All of the Developer Portal API calls are either unauthenticated or made by an authenticated developer that is a member of a Developer organization.

Important: If the Portal Delegated User Registry is selected for the Catalog that you want to work with, you cannot use the Developer Portal REST APIs. This is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. For more information, see Portal Delegated User Registry.

Developer Portal operations

You can complete the following operations with the Developer Portal API:

1. Onboard developers.
2. As a developer, you can manage your own user profile and memberships.
3. As a developer organization owner, you can manage memberships.
4. Browse published applications and Plans.
5. Create applications and subscribe to Plans.
6. Gather analytics data about API, Plan, and application usage.

All Portal API calls are made within the context of a Catalog.
Note: Developer Portal REST APIs enable you to use REST APIs to perform some of the operations that are normally carried out in the Developer Portal UI. For more information about the Developer Portal UI, see Discover and use APIs through the Developer Portal.

- **Getting started**
  Use the following tasks for help with getting started with the Developer Portal REST APIs.
- **Signing up to the Developer Portal**
  You can sign users up to the Developer Portal by using REST API calls. The procedure varies according to the type of user registry that is used to authenticate log in to the Developer Portal.
- **REST APIs reference**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Getting started

Use the following tasks for help with getting started with the Developer Portal REST APIs.

- **Invoking Developer Portal REST APIs**
  To access the Developer Portal API within the context of a developer portal, you can use any of three possible methods; configure the `X-IBM-APIManagement-Context` header, directly invoke the Developer Portal REST APIs, or use the `X-Override-Host` header.
- **Viewing the OpenAPI (Swagger) documentation for the Developer Portal API**
  You can view the OpenAPI (Swagger) documentation for the Developer Portal API to interact with artifacts in the Developer Portal that are associated with the ***`APImanagerHost`***.

- **Authenticating requests**
  All requests are authenticated using HTTP Basic Authentication.
- **Accessing Developer Portal configuration information**
  You can access configuration details about your Developer Portal by making the following authenticated call.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Invoking Developer Portal REST APIs

To access the Developer Portal API within the context of a developer portal, you can use any of three possible methods; configure the `X-IBM-APIManagement-Context` header, directly invoke the Developer Portal REST APIs, or use the `X-Override-Host` header.

Before you begin

If you want to use Method 2 and directly invoke the Developer Portal REST APIs, a Domain Name Server (DNS) record must be configured first:

- Configure the base host name by setting up a DNS record. For example, *manage.dev.acme.com*.
- Point the DNS record to the IP address of the API Connect management server.

Important: If you create a base host name, you must ensure that the name that is configured is unique across your API Connect cluster.

About this task

The three possible methods that you can use to invoke the Developer Portal REST APIs are as follows:

- Method 1: Configure the `X-IBM-APIManagement-Context` header. In this instance, you must provide the Management service address for each provider organization and catalog that is required for each API call.
- Method 2: Directly invoke the Developer Portal REST APIs. If you are using a custom Developer Portal, you can create a Management service alias address (Developer Portal API host name) that can be used for each API call. For this method to work, a DNS record must be configured in advance.
- Method 3: Use the `X-Override-Host` header. If you are using a custom Developer Portal the Developer Portal REST APIs can be directly invoked without creating a DNS record; but a Management service alias address (Developer Portal API host name) is required.

Note:

- If you use Method 2 or Method 3 and you are using a custom Developer Portal, you can configure a Developer Portal API host name in the Hostname for Developer Portal API Calls field in the Catalog settings in API Manager. Configuring a host name enables API Manager to map your host name to the provider organization and Catalog of the Developer Portal API calls, instead of requiring you to look them up and include them in your calls. If you are using the IBM Developer Portal, you do **not** configure a Developer Portal API host name.
- If the Portal Delegated User Registry is selected for the Catalog that you want to work with, you cannot use the Developer Portal REST APIs. This is because the user management is delegated to the Developer Portal, and consequently the management server can no longer provide user authentication. For more information, see Using the Portal Delegated User Registry.
- To use the Developer Portal REST APIs, your Catalog must be using either the IBM Developer Portal or a custom Developer Portal. You specify the required Developer Portal type by selecting either IBM Developer Portal or Other when you configure the Developer Portal in the Catalog; for more information, see Creating and configuring Catalogs.
- You authenticate Developer Portal REST API calls by using the login credentials of a Developer Portal user account.

Procedure

- Method 1: Configure the `X-IBM-APIManagement-Context` header.
  Configure the header in the following format:

  ```
  provider_organization_name.catalog_short_name
  ```

  Where:
  - The *provider_organization_name* is the URL path segment for your provider organization
  - The *catalog_short_name* is the URL path segment for the Catalog in which your provider organization is stored.

  For example:

```
curl -k -i -H "X-IBM-APIManagement-Context: provider_organization_name.catalog_short_name" -X GET
https://management_service_address/v1/portal/products
```

Note: Developer Portal REST APIs are invoked with https://*management_service_address*/v1/portal, where *management_service_address* is the original unmapped Management service host name or IP address.

- Method 2: Directly invoke the Developer Portal REST APIs.
  1. **(Custom Developer Portal only; do not complete this step if you are using the IBM Developer Portal)** Configure a Management service alias address by completing the following steps:
     a. Log in to the API Manager UI as the owner of the Catalog for the Developer Portal in question.
     b. If you have not previously pinned the UI navigation pane, then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
     c. Select the Catalog that you want to work with in the Dashboard section of API Manager.
     d. Click the required Catalog to display its details, then click Settings.
     e. V5.0.2 and earlier Click Configuration.
     f. Click Endpoints.
     g. V5.0.7+ Click Portal.
     h. Enter a host name in the Hostname for Developer Portal API Calls field, then click Save.
  2. Call Developer Portal APIs by using the following root URL:

     ```
     https://management_service_address/v1/portal
     ```

     where *management_service_address* has one or other of the following values:
     - The host name IP address of your Management service if you are using the IBM Developer Portal.
     - The host name of your Management service alias if you are using a custom Developer Portal and have configured a Management service alias.

     For example:

     ```
     curl -k -i -X GET https://management_service_address/v1/portal/products
     ```

     Important: To directly invoke the Developer Portal REST APIs by using **`https://alias_management_service_address`**, a DNS record for the host name must be set up.
- Method 3: Use the **`X-Override-Host`** header.

  If you cannot set up a DNS record, use the **`X-Override-Host`** header with each Developer Portal API call, and use the unmapped Management service address in your calls.
  1. **(Custom Developer Portal only; do not complete this step if you are using the IBM Developer Portal)** Configure a Management service alias address by completing the following steps:
     a. Log in to the API Manager UI as the owner of the Catalog for the developer portal in question.
     b. If you have not previously pinned the UI navigation pane, then click the Navigate to icon ☰. The API Manager UI navigation pane opens. To pin the UI navigation pane, click the Pin menu icon 📌.
     c. Select the Catalog that you want to work with in the Dashboard section of API Manager.
     d. Click the required Catalog to display its details, then click Settings.
     e. V5.0.2 and earlier Click Configuration.
     f. Click Endpoints.
     g. V5.0.7+ Click Portal.
     h. Enter a host name in the Hostname for Developer Portal API Calls field, for example *manage.dev.acme.com*, then click Save.
  2. Use the following command for Developer Portal API calls, for example:

     ```
     curl -k -i -H ‘X-Override-Host: manage.dev.acme.com’ -X GET
     https://management_service_address/v1/portal/products
     ```

     where *management_service_address* is the unmapped Management service host name or IP address.

  All requests use the HTTPS scheme; there is no access by unsecured HTTP. The default certificate that is presented by the management nodes is self-signed.

Related information

- Creating and configuring Catalogs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Viewing the OpenAPI (Swagger) documentation for the Developer Portal API

You can view the OpenAPI (Swagger) documentation for the Developer Portal API to interact with artifacts in the Developer Portal that are associated with the ***APImanagerHost***.

Procedure

To view the OpenAPI (Swagger) documentation for the Developer Portal API, enter the following URL into a browser window:

```
https://APImanagerHost/v1/docs/swagger-ui/index.html
```

Where:

- *APImanagerHost* is the base host name for the management appliance. For example, if you connect to the Cloud Manager as `https://1.2.3.4/cmc`, then you would display the OpenAPI documentation by connecting to `https://1.2.3.4/v1/docs/swagger-ui/index.html`. If you are prompted for a username and password, enter the credentials of the Cloud Manager admin account, but prefix the username with `cmc/`, so the username becomes `cmc/admin`.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Authenticating requests

All requests are authenticated using HTTP Basic Authentication.

Procedure

1. For commands executed on behalf of the API provider, the credentials presented in the authorization header are of the following form:

   ```
   apimanager/username:password
   ```

2. For commands executed on behalf of the developer, the credentials presented in the authorization header are of the following form:

   ```
   username:password
   ```

3. For commands executed on behalf of the Cloud Manager administrator account, the credentials presented in the authorization header are of the following form:

   ```
   cmc/username:password
   ```

Results

Credentials are authenticated using the user registry configured for the API Manager or portal respectively.
Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Accessing Developer Portal configuration information

You can access configuration details about your Developer Portal by making the following authenticated call.

Procedure

Issue the following authenticated call.

```
GET https://mgmt-cluster-ip/v1/portal/config?originURL=the URL used to access the developer portal
```

By default, the URL used to access the Developer Portal is https://*mgmt-cluster-ip*/*provider-organization-name*|*id*/*catalog-name*|*id*.

Results

This call returns the following fields:

orgID
: The ID of the provider organization that is associated with this Developer Portal.

orgName
: The name of the provider organization that is associated with this Developer Portal.

envID
: The ID of the Catalog that is associated with this Developer Portal.

envName
: The name of the Catalog that is associated with this Developer Portal.

portalURL
: The default landing URL for this Developer Portal.

customPortalURL
: The URL for a custom portal or optionally, a customized URL for the Developer Portal provided by API Connect. This URL can be specified in the API Manager UI in the Catalogs window.

gatewayURL
: The default base URL for invoking APIs published to this Portal. If the customGatewayURL property is set, it overrides this gatewayURL value.

portalAPIHost
: The portalAPIHost must be configured to invoke the Portal API. It is the host name that is used for all Portal API calls associated with this Developer Portal. It is set in the API Manager UI in the Catalogs window.

userRegistry
: The user registry information for this Developer Portal. The type can be LDAP or APIM. A type of APIM means that the Developer Portal is configured with an API Manager local user registry.

customGatewayURL
: The Custom Gateway URL configured for this Catalog. It is the base URL for invoking APIs published to this Developer Portal. If not set, then use the value on the gatewayURL property.

invitationEnabled
: If set to false, then Developer Organization owners cannot invite new members. Specifically, the **`POST/v1/portal/orgs/:id/members`** call is disabled for all Developer Organizations in this Developer Portal. For more information, see Invite a user to join my developer organization.

selfSignUpEnabled
: If set to false, then Self-service onboarding is disabled and the **`POST/v1/portal/users/register`** call returns a `403`. Users can be added by creating new Developer Organizations from API Manager, and by inviting new members to a Developer Organization if invitationEnabled is set to true.

expired
: For the on-premises offering, this field can be ignored and is always set to false. If building a Developer Portal for the IBM® API Connect on Cloud (SaaS) offering, this property indicates if the subscription to the parent Provider Organization has expired.

Related tasks

- Invoking Developer Portal REST APIs

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Signing up to the Developer Portal

You can sign users up to the Developer Portal by using REST API calls. The procedure varies according to the type of user registry that is used to authenticate log in to the Developer Portal.

The following subtopics describe how to use REST APIs to sign users up to the Developer Portal, according to user registry type:

- **Signing up to the Developer Portal (local user registry)**
  If your Catalog is using a local user registry to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.
- **Signing up to the Developer Portal (LDAP user registry)**
  If your Catalog is using an LDAP user registry to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.
- **Signing up to the Developer Portal (authentication URL)**
  If your Catalog is using an authentication URL to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Signing up to the Developer Portal (local user registry)

If your Catalog is using a local user registry to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.

To complete the steps described in this topic, you need an environment capable of sending HTTP requests and receiving HTTP responses. Here, the **curl** command is used in a command line interface to demonstrate the REST API calls. In your environment, make the same REST API calls as are made with **curl** commands in this topic.

To sign a user up to the Developer Portal, you complete the following steps (detailed instructions are provided in the sections that follow):

1. Request to sign up. Make one or other of the following REST API calls:
   - Request user self sign up to the Developer Portal and provide basic account information. A new Developer organization is created with the user as its owner.
   - Invite the user to join an existing Developer organization.

   On successful completion of the call, an activation email is sent to the user, and the account is in a pending state until the activation is completed.
2. Obtain the user authentication information and activation URL. The activation email contains authentication information in base64 encoded form. You must base64 decode this authentication information.
3. Activate the user account. Make a REST API call to activate the account, supplying the decoded authentication information.

Requesting to sign up to the Developer Portal

A user sign up request can be initiated either by self sign up or by invitation to join a Developer organization, as follows:

- To initiate a self sign up request, make the following REST API call:

  ```
  curl -k -v -H "X-IBM-APIManagement-Context: provider_org.catalog" -H "Content-Type: 
  application/json" -X POST -d \
  '{"username":"email_address","password":"password","firstName":"first_name","lastName":"last_name",
  "organization":"dev_org_name"}' \
  https://management_service_address/v1/portal/users/register
  ```

  where:
  - *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
  - *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
  - *email_address* is the email address of the user that you want to sign up. The email address is the user ID with which the user will log in to the Developer Portal.
  - *password* is the initial login password. The password must contain characters from three of the following four categories: uppercase, lowercase, numeric, and punctuation (for example, !, $, #, %).

    - *first_name* is the user's first name.
    - *last_name* is the user's last name.
    - *dev_org_name* is the name that will be assigned to the user's Developer organization.
    - *management_service_address* is the host name or IP address of your API Connect Management service.
- To invite a user to join a Developer organization, make the following REST API call:

  ```
  curl -k -v -u "dev_org_owner_id:dev_org_owner_password" \
  -H "X-IBM-APIManagement-Context: provider_org.catalog" -H "Content-Type: application/json" \
  -X POST -d '{"name":"email_address", "roles":[array_of_roles]}' \
  https://management_service_address/v1/portal/orgs/dev_org_id/members
  ```

  where:
    - *dev_org_owner_id* is the login ID of the owner of the Developer organization that the user is being invited to join.
    - *dev_org_owner_password* is the login password of the owner of the Developer organization that the user is being invited to join.
    - *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
    - *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
    - *email_address* is the email address of the user that you want to invite.
    - (optional) *array_of_roles* is a JSON array specifying the roles that you want to assign to the user. The following roles are available:
        - "developer": Can create and edit applications, manage client keys and subscribe to Plans.
        - "viewer": Can only view applications and application activity.

      For example:

      ```
      "roles":["developer"]
      ```

      If omitted, the default value is `["developer"]`.
    - *management_service_address* is the host name or IP address of your API Connect Management service.
    - *dev_org_id* is the ID of the Developer organization.
      To obtain the ID of the Developer organization, make the following REST API call, which lists the details of the Developer organizations of which the authenticated user is a member:

      ```
      curl -k -v -u "dev_org_user_id:dev_org_password" \
      -H "X-IBM-APIManagement-Context: provider_org.catalog" \
      -H 'Content-Type: application/json' -X GET https://management_service_address/v1/portal/orgs
      ```

After the sign up request is submitted, an activation email is sent to the specified email address.

Obtaining the user authentication information and activation REST API URL

The invitation email contains an activation link URL with an activationToken parameter. The activation link has the following format:

```
https://portal_url/?q=ibm_apim/activate/x&activationToken=activation_token
```

To obtain the user authentication information and the activation REST API URL, you must decode the activation token by using a base64 decoder.

The following example shows the format of the decoded activation token:

```
{"url":"https://myhost.com/v1/portal/users/48663d86e4b0e688ee1cded4/activate",
"username":"!BASE64_SIV_ENC!_AQZ+4GefJbUFPRVVbZb4M/fGgkeyHi9zZT4I8RKA3+IRAAAAGKc0NUo0lvPq2VxBHalDyYRHKdb
QWoCs3dkCf43e9WoQ",
"authentication":
{"username":"58651bdfe4b0e687ee1cc7a1/58651d9ce5b0e687ee1cc6b0/jS2lD1fW8nQ8zW0dE3cS6eI3hU1hP5oN3jU8dK7sP
8",
"password":"CG+YUlm1cLITToGCfciSeVMdFeV7BQMxvsR0XxZbkc"},
"providerContext":{"orgID":"5819e9e6e4b0e645ee1bfdeb","environmentID":"58651bdfe4b0e687ee1cc7a1"}}
```

In the decoded token, the `authentication` field contains the authentication data in the following format:

```
"authentication":{"username":"encoded_user_name","password":"encoded_password"}
```

The `url` field contains the activation REST API URL in the following format:

```
"url":"activation_url"
```

The activation REST API URL has the following format:

```
https://management_service_address/v1/portal/users/user_id/activate
```

where:

- *management_service_address* is the host name or IP address of your API Connect Management service.
- *user_id* is a unique user ID string.

Note the *encoded_user_name*, *encoded_password*, and *activation_url* values because they are required in the REST API call that you use to activate the user account.

Activating the user account

Make the following REST API call:

```
curl -k -v -H "Content-Type: application/json" -H "X-IBM-APIManagement-Context: provider_org.catalog" \
-u 'encoded_user_name:encoded_password' -X POST -d \
'{"password":"password","firstName":"first_name","lastName":"last_name"}' activation_url
```

where:

- *provider_org* is the name of the provider organization.
- *catalog* is the name of the Catalog.
- *encoded_user_name* and *encoded_password* are the authentication information values that you noted in Obtaining the user authentication information and activation REST API URL.
- *activation_url* is the activation REST API URL that you noted in Obtaining the user authentication information and activation REST API URL.
- *password* is the login password. The password must contain characters from three of the following four categories: uppercase, lowercase, numeric, and punctuation; for example, !, $, #, % **(required only if the user was invited to join a Developer organization.)**
- *first_name* is the user's first name **(required only if the user was invited to join a Developer organization.)**
- *last_name* is the user's last name **(required only if the user was invited to join a Developer organization.)**

On successful completion of the activation REST API call, the user can log in to the Developer Portal user interface, with their email address as the user ID.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Signing up to the Developer Portal (LDAP user registry)

If your Catalog is using an LDAP user registry to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.

To complete the steps described in this topic, you need an environment capable of sending HTTP requests and receiving HTTP responses. Here, the **curl** command is used in a command line interface to demonstrate the REST API calls. In your environment, make the same REST API calls as are made with **curl** commands in this topic.

To sign a user up to the Developer Portal, make one or other of the following REST API calls:

- Sign the user up directly. A new Developer organization is created with the user as its owner.
- Invite the user to join an existing Developer organization.

On successful completion of either call, the user account is activated immediately, and an email, containing a link to the Developer Portal, is sent to the email address of the specified user.
Note: The user must already exist in the LDAP registry before you make the REST API call.
To sign the user up directly, make the following REST API call:

```
curl -k -v -H "X-IBM-APIManagement-Context: provider_org.catalog" \
-H "Content-Type: application/json" -X POST -d '{"username":"ldap_user_name","password":"password"}' \
https://management_service_address/v1/portal/users/register
```

where:

- *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
- *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
- *ldap_user_name* is the user name of the existing user that you want to sign up. The user name is the user ID with which the user will log in to the Developer Portal.
- *password* is the login password. The password must contain characters from three of the following four categories: uppercase, lowercase, numeric, and punctuation (for example, !, $, #, %).
- *management_service_address* is the host name or IP address of your API Connect Management service.

To invite a user to join a Developer organization, make the following REST API call:

```
curl -k -v -u "dev_org_owner_id:dev_org_owner_password" \
-H "X-IBM-APIManagement-Context: provider_org.catalog" -H "Content-Type: application/json" \
-X POST -d '{"name":"ldap_user_name", "roles":[array_of_roles]}' \
https://management_service_address/v1/portal/orgs/dev_org_id/members
```

where:

- *dev_org_owner_id* is the login ID of the owner of the Developer organization that the user is being invited to join.
- *dev_org_owner_password* is the login password of the owner of the Developer organization that the user is being invited to join.
- *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
- *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
- *ldap_user_name* is the user name of the existing user that you want to invite.
- (optional) *array_of_roles* is a JSON array specifying the roles that you want to assign to the user. The following roles are available:
    - "developer": Can create and edit applications, manage client keys and subscribe to Plans.
    - "viewer": Can only view applications and application activity.

  For example:

  ```
  "roles":["developer"]
  ```

  If omitted, the default value is `["developer"]`.
- *management_service_address* is the host name or IP address of your API Connect Management service.
- *dev_org_id* is the ID of the Developer organization.
  To obtain the ID of the Developer organization, make the following REST API call, which lists the details of the Developer organizations of which the authenticated user is a member:

  ```
  curl -k -v -u "dev_org_user_id:dev_org_password" \
  -H "X-IBM-APIManagement-Context: provider_org.catalog" \
  -H 'Content-Type: application/json' -X GET https://management_service_address/v1/portal/orgs
  ```

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Signing up to the Developer Portal (authentication URL)

If your Catalog is using an authentication URL to authenticate user log in to the Developer Portal, you can use Developer Portal REST API calls to sign users up.

To complete the steps described in this topic, you need an environment capable of sending HTTP requests and receiving HTTP responses. Here, the **curl** command is used in a command line interface to demonstrate the REST API calls. In your environment, make the same REST API calls as are made with **curl** commands in this topic.

To sign a user up to the Developer Portal, you complete one or other of the following steps:

- Sign the user up directly. A new Developer organization is created with the user as its owner. The user account is activated immediately.
- Sign the user up by invitation. You must complete the following steps:
    1. Invite the user to join a Developer organization by making a REST API call. On successful completion of the call, an activation email is sent to the user, and the account is in a pending state until the activation is completed.
    2. Obtain the user authentication information and activation REST API URL. The activation email contains authentication information in base64 encoded form. You must base64 decode this authentication information.
    3. Activate the user account. Make a REST API call to activate the account, supplying the decoded authentication information.

These steps are described in the following sections:

Signing the user up to the Developer Portal directly

To sign the user up directly, make the following REST API call:

```
curl -k -v -H "X-IBM-APIManagement-Context: provider_org.catalog" -H "Content-Type: application/json" -X
POST -d \
'{"username":"user_name","password":"password","firstName":"first_name","lastName":"last_name","organiza
tion":"dev_org_name"}' \
https://management_service_address/v1/portal/users/register
```

where:

- *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
- *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
- *user_name* is the user name of the user that you want to sign up. The user name is the user ID with which the user will log in to the Developer Portal.
- *password* is the initial login password. The password must contain characters from three of the following four categories: uppercase, lowercase, numeric, and punctuation (for example, !, $, #, %).
- (optional) *first_name* is the user's first name.
- (optional) *last_name* is the user's last name.
- (optional) *dev_org_name* is the name that will be assigned to the user's Developer organization. The default name is the *email_address* value.
- *management_service_address* is the host name or IP address of your API Connect Management service.

On successful completion of the call, the account is active and the user can log in to the Developer Portal.

Inviting the user to join a Developer organization

To invite a user to join a Developer organization, make the following REST API call:

```
curl -k -v -u "dev_org_owner_id:dev_org_owner_password" \
-H "X-IBM-APIManagement-Context: provider_org.catalog" -H "Content-Type: application/json" \
-X POST -d '{"name":"email_address", "roles":[array_of_roles]}' \
https://management_service_address/v1/portal/orgs/dev_org_id/members
```

where:

- *dev_org_owner_id* is the login ID of the owner of the Developer organization that the user is being invited to join.
- *dev_org_owner_password* is the login password of the owner of the Developer organization that the user is being invited to join.
- *provider_org* is the name of the provider organization of the Developer Portal to which the user is to be signed up.
- *catalog* is the name of the Catalog of the Developer Portal to which the user is to be signed up.
- *email_address* is the email address of the user that you want to invite.
- (optional) *array_of_roles* is a JSON array specifying the roles that you want to assign to the user. The following roles are available:
  - "developer": Can create and edit applications, manage client keys and subscribe to Plans.
  - "viewer": Can only view applications and application activity.

  For example:

  ```
  "roles":["developer"]
  ```

  If omitted, the default value is `["developer"]`.
- *management_service_address* is the host name or IP address of your API Connect Management service.
- *dev_org_id* is the ID of the Developer organization.
  To obtain the ID of the Developer organization, make the following REST API call, which lists the details of the Developer organizations of which the authenticated user is a member:

  ```
  curl -k -v -u "dev_org_user_id:dev_org_password" \
  -H "X-IBM-APIManagement-Context: provider_org.catalog" \
  -H 'Content-Type: application/json' -X GET https://management_service_address/v1/portal/orgs
  ```

On successful completion of the invite operation, an activation email is sent to the specified email address.

Obtaining the user authentication information and activation REST API URL

The invitation email contains an activation link URL with an activationToken parameter. The activation link has the following format:

```
https://portal_url/?q=ibm_apim/activate/x&activationToken=activation_token
```

To obtain the user authentication information and the activation REST API URL, you must decode the activation token by using a base64 decoder.

The following example shows the format of the decoded activation token:

```
{"url":"https://myhost.com/v1/portal/users/48663d86e4b0e688ee1cded4/activate",
"username":"!BASE64_SIV_ENC!_AQZ+4GefJbUFPRVVbZb4M/fGgkeyHi9zZT4I8RKA3+IRAAAAGKc0NUo0lvPq2VxBHalDyYRHKdb
QWoCs3dkCf43e9WoQ",
"authentication":
{"username":"58651bdfe4b0e687ee1cc7a1/58651d9ce5b0e687ee1cc6b0/jS2lD1fW8nQ8zW0dE3cS6eI3hU1hP5oN3jU8dK7sP
8",
"password":"CG+YUlm1cLITToGCfciSeVMdFeV7BQMxvsR0XxZbkc"},
"providerContext":{"orgID":"5819e9e6e4b0e645ee1bfdeb","environmentID":"58651bdfe4b0e687ee1cc7a1"}}
```

In the decoded token, the `authentication` field contains the authentication data in the following format:

```
"authentication":{"username":"encoded_user_name","password":"encoded_password"}
```

The `url` field contains the activation REST API URL in the following format:

```
"url":"activation_url"
```

The activation REST API URL has the following format:

```
https://management_service_address/v1/portal/users/user_id/activate
```

where:

- *management_service_address* is the host name or IP address of your API Connect Management service.
- *user_id* is a unique user ID string.

Note the *encoded_user_name*, *encoded_password*, and *activation_url* values because they are required in the REST API call that you use to activate the user account.

Activating the user account

Make the following REST API call:

```
curl -k -v -H "Content-Type: application/json" -H "X-IBM-APIManagement-Context: provider_org.catalog" \
-u 'encoded_user_name:encoded_password' -X POST -d \
'{"username":"user_id","password":"password","firstName":"first_name","lastName":"last_name"}'
activation_url
```

where:

- *provider_org* is the name of the provider organization.
- *catalog* is the name of the Catalog.
- *encoded_user_name* and *encoded_password* are the authentication information values that you noted in Obtaining the user authentication information and activation REST API URL.
- *activation_url* is the activation REST API URL that you noted in Obtaining the user authentication information and activation REST API URL.
- *user_id* is the user ID with which the user will log in to the Developer Portal.
- *password* is the login password. The password must contain characters from three of the following four categories: uppercase, lowercase, numeric, and punctuation (for example, !, $, #, %).
- (optional) *first_name* is the user's first name.
- (optional) *last_name* is the user's last name.

On successful completion of the activation REST API call, the user can log in to the Developer Portal user interface with the specified user ID.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

REST APIs reference

- **User Management APIs**
  Using APIs to manage users and developer organizations.
- **Browsing Products**
  Using APIs to browse public Products published to a portal.
- **Browsing APIs**
  Using REST APIs to browse public APIs published to a portal and APIs available to a developer organization.
- **Application Management APIs**
  Using APIs to manage applications and subscriptions.
- **Configuration APIs**
  Using APIs to obtain details about a portal configuration.
- V5.0.8 + **Analytics**
  Analyzing Catalog APIs.

User Management APIs

Using APIs to manage users and developer organizations.

Use the User Management APIs to manage individual users and user memberships to developer organizations. You can perform the following tasks.

- Invite developers to join personal developer organizations.
- List members of a developer organization and their roles.
- Remove a member from a developer organization.
- Remove yourself from a developer organization.
- Update personal profile information.
- Update personal notification settings.

- **Reset user's password**
  Reset user's password, after user has requested password reset and clicked on the password reset link in the email.
- **Request password reset**
  Request password reset when user has forgotten the password. This call is not supported if the user registry is LDAP. If the user registry is the Local User Registry, then the name property must be set to the email of the user.
- **Sign-Up to the portal**
  This call allows an unauthenticated caller to request to join a portal. Once the sign-up procedure is complete, the user will be active and will be the owner of a new developer organization.
- **Get list of users matching the search criteria**
  Get list of users matching the search criteria. The result is a list of usernames from the user registry.
- **Update my profile**
  Update my profile information, such as first name, last name, and email, if the user registry is writable.
- **Get my profile information, such as first name, last name, and email.**
  Get my profile
- **Change my password**
  Change my password (when user knows the existing password).
- **Retrieve owner of an organization**
  Returns the owner of the organization. The authenticated caller must be a member of the organization.
- **Change the owner of an organization**
  Removes the current owner of the organization and replaces with the specified user as the new owner.
- **Retrieve the list of developer organizations in which the authenticated user is a member**
  Retrieve the list of developer organizations in which the authenticated user is a member. If search filter is supplied, only return the list of developer organizations with exact matching name.
- **Retrieve the developer organization specified by its ID**
  Retrieve the developer organization specified by its ID in which the authenticated user is a member.
- **Create a new developer organization owned by the caller**
  The authenticated user must already exist and be active in the system.
- **Update the developer organization**
  Update the developer organization specified by id. Only organization owner can perform this action.
- **Return the number of notifications**
- **Return a list of notifications**
- **Retrieve the settings for an organization member.**
  Retrieve the settings for an organization member.
- **Save the settings for an organization member.**
  Save the settings for an organization member.
- **Remove a user from the developer organization.**
  Members who are not the owner may remove themselves from a developer organization. Only the owner can remove other members from the organization and he cannot remove himself.
- **List members of a developer organization**
  Returns all members of a developer organization including the owner. The authenticated caller must be a member of the organization.
- **Invite a user to join my developer organization**
  The inviter must be the owner of the developer organization.
- **List roles available for this Portal**
  One or more of these roles can be assigned to members of developer organizations in this Portal

- **Retrieve a role**
- **Activate user**
  Activate the user and developer organization memberships after a user signs up or is invited to the portal.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reset user's password

Reset user's password, after user has requested password reset and clicked on the password reset link in the email.

Note: The url/path for this API is subject to change. Do not hard-code this path in your application. Instead use the "url" field as provided in the reset password token. For more details please refer to the IBM API Connect online documentation.

Sample method invocation

```
  curl -X POST \
    -d '
        {
           "password": string
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/users/{userID}/reset-password
```

Parameters	**Path Parameters** userID : Username or ID
Response content	**MIME type: application/json** `none`
Status codes	204 : Password has been reset. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Request password reset

Request password reset when user has forgotten the password. This call is not supported if the user registry is LDAP. If the user registry is the Local User Registry, then the name property must be set to the email of the user.

Sample method invocation

```
  curl -X POST \
    -d '
        {
           "name": string
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/users/forgot-password
```

Response content	**MIME type: application/json** `none`
Status codes	200 : Requested password reset. 400 : Invalid or missing request payload properties. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Sign-Up to the portal

This call allows an unauthenticated caller to request to join a portal. Once the sign-up procedure is complete, the user will be active and will be the owner of a new developer organization.

If the user registry is LDAP, then the only required fields are username and password. If they do not specify an organization then a developer organization will be created with a default name of 'firstName lastName'.

If the user registry is a Local User Registry then all fields are required and the username must be the email address of the new user.

Sample method invocation

```
  curl -X POST \
    -d '
        {
           "additionalFields":
           {
           }
           ,
           "firstName": string,
           "lastName": string,
           "organization": string,
           "password": string,
           "username": string
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/users/register
```

Response content	**MIME type: application/json**
	`{`
	`   id: string`
	`   name: string`
	`   url: string`
	`   additionalFields:`
	`   {`
	`   }`
	
	`   email: string`
	`   firstName: string`
	`   lastLoginTime: string`
	`   lastName: string`
	`   organization:`
	`   {`
	`      id: string`
	`      name: string`
	`      url: string`
	`   }`

	`status: string`
	`}`

Status codes	201 : User has been created. 400 : Invalid or missing request payload properties. 400 : User already registered or username and/or password incorrect. 401 : Authentication failed. Valid username and password are required. 403 : Self-service onboarding is disabled for this portal. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Get list of users matching the search criteria

Get list of users matching the search criteria. The result is a list of usernames from the user registry.

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/users
```

Parameters	**Query Parameters** username : Search filter

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     username: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Retrieved list of users matching the specified search crtieria. 400 : Too many search results. Refine the search filter and try again. 401 : Authentication failed. Valid username and password are required. 403 : User must be an owner of a developer organization in the same environment to search for users. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update my profile

Update my profile information, such as first name, last name, and email, if the user registry is writable.

Sample method invocation

```
curl -X PUT \
  -d '
      {
         "additionalFields":
         {
         }
         ,
         "firstName": string,
         "lastName": string
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/me
```

Response content	MIME type: application/json
	`{ id: string name: string url: string additionalFields: { } email: string firstName: string lastLoginTime: string lastName: string status: string }`
Status codes	200 : User's profile has been updated. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Get my profile information, such as first name, last name, and email.

Get my profile

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/me
```

Parameters	Query Parameters

	expand : Resource ID

Response content	**MIME type: application/json**
	`{`
	`    id: string`
	`    name: string`
	`    url: string`
	`    additionalFields:`
	`    {`
	`    }`
	`    email: string`
	`    firstName: string`
	`    lastLoginTime: string`
	`    lastName: string`
	`    status: string`
	`}`

Status codes	200 : User's profile has been retrieved. 204 : User exists. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Change my password

Change my password (when user knows the existing password).

Sample method invocation

```
 curl -X PUT \
   -d '
       {
          "newPassword": string,
          "oldPassword": string
       }

      ' \
   https://<mgmt-cluster-ip>/v1/portal/me/password
```

Response content	**MIME type: application/json** `none`
Status codes	204 : Password has been updated. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve owner of an organization

Returns the owner of the organization. The authenticated caller must be a member of the organization.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/owner
```

Parameters	Path Parameters orgID : Organization Id
Response content	MIME type: application/json `{ id: string name: string url: string }`
Status codes	200 : Organization owner has been retrieved. 401 : Authentication failed. Valid username and password are required. 403 : The user must be a member of the developer organization. 404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Change the owner of an organization

Removes the current owner of the organization and replaces with the specified user as the new owner.

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  -d '
      {
         "name": string
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/owner
```

Parameters	Path Parameters orgID : Organization Id
Response content	MIME type: application/json `none`

Status codes	204 : Organization owner has been changed. 400 : The new owner must be an active member of the organization 401 : Authentication failed. Valid username and password are required. 403 : The user must be the owner of the developer organization. 404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve the list of developer organizations in which the authenticated user is a member

Retrieve the list of developer organizations in which the authenticated user is a member. If search filter is supplied, only return the list of developer organizations with exact matching name.

Sample method invocation

```
 curl -X GET \
   -H 'Content-Type: application/json' \
   https://<mgmt-cluster-ip>/v1/portal/orgs
```

Parameters	**Query Parameters** name : Search filter

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      id: string`
	`      name: string`
	`      url: string`
	`      owner: boolean`
	`      roles: array[string]`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : The organization list has been retrieved. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve the developer organization specified by its ID

Retrieve the developer organization specified by its ID in which the authenticated user is a member.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}
```

Parameters	**Path Parameters** orgID : Organization Id

Response content	**MIME type: application/json**
	`{`
	`   id: string`
	`   name: string`
	`   url: string`
	`}`

Status codes	200 : The organization has been retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Unable to retrieve the organization because the user is not a member of that organization. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Create a new developer organization owned by the caller

The authenticated user must already exist and be active in the system.

Sample method invocation

```
curl -X POST \
  -H 'Content-Type: application/json' \
  -d '
      {
         "name": string
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs
```

Response content	**MIME type: application/json**
	`{`
	`   id: string`
	`   name: string`
	`   url: string`
	`}`

Status codes	201 : The requested organization was created successfully. 400 : Validation for developer organization name failed. 401 : Authentication failed. Valid username and password are required.

	404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update the developer organization

Update the developer organization specified by id. Only organization owner can perform this action.

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  -d '
      {
         "name": string
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}
```

Parameters	**Path Parameters** orgID : Organization Id
Response content	**MIME type: application/json** `none`
Status codes	200 : The developer organization has been updated. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 403 : The user must be the owner of the developer organization. 404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return the number of notifications

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/notifications/count
```

Parameters	**Path Parameters** orgID : Organization ID
Response content	**MIME type: application/json**

	`{`
	`    alertCount: integer`
	`    allCount: integer`
	`    eventCount: integer`
	`    unreadCount: integer`
	`}`

Status codes	200 : Notification counts retrieved 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return a list of notifications

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/notifications
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** first : The first notification to receive (e.g., 10th notification), used for pagination max : The maximum number of notifications to receive type : The type of notifications to receive

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      id: string`
	`      datetime: string`
	`      message: string`
	`      type: enum['alert' or 'event']`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Notifications retrieved 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve the settings for an organization member.

Retrieve the settings for an organization member.

The settings are specific to the authenticated user and are in the context of the organization.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Accept: application/json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/members/{memberID}/settings
```

Parameters	**Path Parameters** orgID : Organization id memberID : Member id

Response content	**MIME type: application/json**
	`{`
	`    appNotificationSettings: array[`
	`        {`
	`            appURL: string`
	`            emailEnabled: boolean`
	`            enabled: boolean`
	`            events:`
	`            {`
	`            }`
	`            interval: enum['day' or 'hour' or 'minute']`
	`        }`
	`    ]`
	`}`

Status codes	200 : Successfully retrieved organization member's settings. 401 : Authentication failed. Valid username and password are required. 403 : The authenticated user is not allowed to access the resource. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Save the settings for an organization member.

Save the settings for an organization member.

The settings are specific to the authenticated user and are in the context of the organization.

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
    -H 'Content-Type: application/json' \
    -H 'Accept: application/json' \
    -d '
        {
           "appNotificationSettings": array[
              {
                 "appURL": string,
                 "emailEnabled": boolean,
                 "enabled": boolean,
                 "events": enum['rateLimitPercent50' or ' rateLimitPercent75' or ' rateLimitPercent90' 
or ' rateLimitPercent100'],
                 "interval": enum['minute' or ' hour' or ' day']
              }
           ]
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/members/{memberID}/settings
```

Parameters	Path Parameters orgID : Organization id memberID : Member id

Response content	MIME type: application/json
	`{`
	`   appNotificationSettings: array[`
	`      {`
	`         appURL: string`
	`         emailEnabled: boolean`
	`         enabled: boolean`
	`         events:`
	`         {`
	`         }`
	`         interval: enum['day' or 'hour' or 'minute']`
	`      }`
	`   ]`
	`}`

Status codes	200 : Saved organization member settings. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 403 : The authenticated user is not allowed to access the resource. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Remove a user from the developer organization.

Members who are not the owner may remove themselves from a developer organization. Only the owner can remove other members from the organization and he cannot remove himself.

Sample method invocation

```
  curl -X DELETE \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/members/{memberID}
```

Parameters	**Path Parameters** orgID : Organization Id memberID : Member Id
Response content	**MIME type: application/json** `none`
Status codes	204 : The user has been removed from the organization. 400 : A user can delete himself if, and only if, he is not the organization owner. To delete another user, the authentciated user must be the organization owner. 401 : Authentication failed. Valid username and password are required. 403 : The user must be a member of the developer organization. 404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List members of a developer organization

Returns all members of a developer organization including the owner. The authenticated caller must be a member of the organization.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/members
```

Parameters	**Path Parameters** orgID : Organization Id

Response content	**MIME type: application/json**
	`{`
	`   id: string`
	`   name: string`
	`   url: string`
	`   roleUrls: array[string]`
	`   status: string`
	`}`

Status codes	200 : The member list has been retrieved. 401 : Authentication failed. Valid username and password are required. 403 : The user must be a member of the developer organization. 404 : The requested organization does not exist. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Invite a user to join my developer organization

The inviter must be the owner of the developer organization.

Sample method invocation

```
curl -X POST \
  -u userid:pw \
  -H 'Content-Type: application/json' \
  -d '
      {
         "name": string,
         "email": string,
         "roles": array[string]
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/members
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization Id</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre>
 {
    id: string
    name: string
    url: string
    additionalFields:
    {
    }

    email: string
    firstName: string
    lastLoginTime: string
    lastName: string
    roles: array[string]
    status: string
 }
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>201 : User has been created and added to the developer organization successfully.
204 : User has been added to the developer organization successfully.
401 : Authentication failed. Valid username and password are required.
403 : The user must be the owner of the developer organization.
403 : Invitations have been disabled for this portal.
404 : The requested organization does not exist.
409 : The user already exists in the developer organization.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List roles available for this Portal

One or more of these roles can be assigned to members of developer organizations in this Portal

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/roles
```

Parameters	**Path Parameters** orgID : Organization ID

Response content	**MIME type: application/json**
	`[`
	`{`
	`id: string`
	`name: string`
	`url: string`
	`displayName: string`
	`}`
	`.`
	`.`
	`.`
	`]`

Status codes	200 : The collection of resources was retrieved. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a role

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/roles/{roleID}
```

Parameters	**Path Parameters** orgID : Organization ID roleID : Role id or name

Response content	**MIME type: application/json**
	`{`
	`id: string`
	`name: string`
	`url: string`
	`displayName: string`
	`}`

Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Activate user

Activate the user and developer organization memberships after a user signs up or is invited to the portal.

Note: The url/path for this API is subject to change. Do not hard-code this path in your application. Instead use the "url" field as provided in the activation token. For more details please refer to the IBM API Connect online documentation.

Sample method invocation

```
  curl -X POST \
    -d '
        {
           "additionalFields":
           {
           }
           ,
           "firstName": string,
           "lastName": string,
           "password": string,
           "username": string
        }

      ' \
    https://<mgmt-cluster-ip>/v1/portal/users/{userID}/activate
```

Parameters	**Path Parameters** userID : Username or ID
Response content	**MIME type: application/json** `none`
Status codes	204 : User has been activated. 400 : User has already been activated. 401 : Authentication failed. Valid username and password are required. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Browsing Products

Using APIs to browse public Products published to a portal.

- **List products**
  List products that are available to the specified organization
- **List public products**
  List public products
- **Retrieve a product in JSON**
  The product must be available to the specified organization. Use this REST API to fetch the definition of a Product in JSON format. You must pass the header "Accept: application/vnd.ibm-apim.product+json" to receive the contents formatted as JSON.
- **Retrieve a product in YAML**
  The product must be available to the specified organization. Use this REST API to fetch the definition of a Product in YAML format. You must pass the header "Accept: application/vnd.ibm-apim.product+yaml" to receive the contents formatted as YAML.

- **Retrieve a public product in JSON**
  Use this REST API to fetch the definition of a Product in JSON format. You must pass the header "Accept: application/vnd.ibm-apim.product+json" to receive the contents formatted as JSON.
- **Retrieve a public product in YAML**
  Use this REST API to fetch the definition of a Product in YAML format. You must pass the header "Accept: application/vnd.ibm-apim.product+yaml" to receive the contents formatted as YAML.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List products

List products that are available to the specified organization

Sample method invocation

```
 curl -X GET \
   -u userid:pw \
   https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/products
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b><pre>
 [
   {
      id: string
      url: string
      info:
      {
      }

      status: enum['TRANSIENT_REMOVED' or 'STOPPED' or 'RUNNING' or 'SUSPENDED' or
'ARCHIVED' or 'SET_MIGRATION_TARGET' or 'RETIRED' or 'DEPRECATED' or 'PUBLISHED' or
'DEPLOYED' or 'PENDING']
   }

   .
   .
   .
 ]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : The collection of resources was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List public products

List public products

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/products
```

Response content	**MIME type: application/json**
	[
	{
	id: string
	url: string
	info:
	{
	}
	status: enum['TRANSIENT_REMOVED' or 'STOPPED' or 'RUNNING' or 'SUSPENDED' or 'ARCHIVED' or 'SET_MIGRATION_TARGET' or 'RETIRED' or 'DEPRECATED' or 'PUBLISHED' or 'DEPLOYED' or 'PENDING']
	}
	.
	.
	.
	]

Status codes	200 : The collection of resources was retrieved. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a product in JSON

The product must be available to the specified organization. Use this REST API to fetch the definition of a Product in JSON format. You must pass the header "Accept: application/vnd.ibm-apim.product+json" to receive the contents formatted as JSON.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Accept: application/vnd.ibm-apim.product+json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/products/{productID}
```

Parameters	**Path Parameters** productID : product ID orgID : Organization ID

Response content	**MIME type: application/json**
	{
	apis:
	{
	}
	plans:
	{
	}
	product: string
	}

Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required.

	403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a product in YAML

The product must be available to the specified organization. Use this REST API to fetch the definition of a Product in YAML format. You must pass the header "Accept: application/vnd.ibm-apim.product+yaml" to receive the contents formatted as YAML.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Accept: application/vnd.ibm-apim.product+yaml' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/products/{productID}
```

Parameters	**Path Parameters** productID : product ID orgID : Organization ID

Response content	**MIME type: application/json**
	`{`
	`   apis:`
	`   {`
	`   }`
	`   plans:`
	`   {`
	`   }`
	`   product: string`
	`}`

Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a public product in JSON

Use this REST API to fetch the definition of a Product in JSON format. You must pass the header "Accept: application/vnd.ibm-apim.product+json" to receive the contents formatted as JSON.

Sample method invocation

```
curl -X GET \
  -H 'Accept: application/vnd.ibm-apim.product+json' \
  https://<mgmt-cluster-ip>/v1/portal/products/{productID}
```

Parameters	**Path Parameters** productID : product ID

Response content	**MIME type: application/json**
	`{`
	`    apis:`
	`    {`
	`    }`
	`    info:`
	`    {`
	`    }`
	`    plans:`
	`    {`
	`    }`
	`    product: string`
	`}`

Status codes	200 : The resource was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a public product in YAML

Use this REST API to fetch the definition of a Product in YAML format. You must pass the header "Accept: application/vnd.ibm-apim.product+yaml" to receive the contents formatted as YAML.

Sample method invocation

```
curl -X GET \
  -H 'Accept: application/vnd.ibm-apim.product+yaml' \
  https://<mgmt-cluster-ip>/v1/portal/products/{productID}
```

Parameters	**Path Parameters** productID : product ID

Response content	**MIME type: application/json**
	`{`
	`    apis:`
	`    {`
	`    }`
	`    info:`
	`    {`
	`    }`
	`    plans:`
	`    {`
	`    }`

	`    product: string`
	`}`

Status codes	200 : The resource was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Browsing APIs

Using REST APIs to browse public APIs published to a portal and APIs available to a developer organization.

When browsing public APIs published to a portal, the response includes the subset of resources that are publicly exposed by one or more Plans.

When using APIs available to a Developer Organization, the response includes public APIs as well as those published specifically to a developer organization.

- **Retrieve API documentation file for a public API**
  Retrieve API documentation file for a public API, by API's ID and document's ID. Caution: Do not invoke this API in Swagger UI directly.
- **Retrieve the WSDL for a public SOAP API**
- **Retrieve the WSDL for the specified SOAP API**
  Retrieve the WSDL for the specified SOAP API. The API must be available to the specified orgs/{orgID}
- **Retrieve an API in Swagger 2.0 JSON format**
  Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 JSON format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+json" to receive the contents formatted as Swagger 2.0 JSON.
- **Retrieve an API in Swagger 2.0 JSON format**
  The API must be available to the specified orgs/{orgID}. Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 JSON format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+json" to receive the contents formatted as Swagger 2.0 JSON.
- **Retrieve an API in Swagger 2.0 YAML format**
  Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 YAML format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+yaml" to receive the contents formatted as Swagger 2.0 YAML.
- **Retrieve an API in Swagger 2.0 YAML format**
  The API must be available to the specified orgs/{orgID}. Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 YAML format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+yaml" to receive the contents formatted as Swagger 2.0 YAML.
- **List public APIs**
- **List APIs**
  List APIs that are available to the specified orgs/{orgID}
- **Retrieve API documentation file for an API**
  Retrieve API documentation file for an API, by API's ID and document's ID. Caution: Do not invoke this API in Swagger UI directly.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve API documentation file for a public API

Retrieve API documentation file for a public API, by API's ID and document's ID. Caution: Do not invoke this API in Swagger UI directly.

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/apis/{apiID}/documents/{docID}/file
```

Parameters	**Path Parameters** apiID : docID :
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve the WSDL for a public SOAP API

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/apis/{apiID}/wsdl
```

Parameters	**Path Parameters** apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 400 : The API associated with the request does not have a SOAP service. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve the WSDL for the specified SOAP API

Retrieve the WSDL for the specified SOAP API. The API must be available to the specified orgs/{orgID}

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apis/{apiID}/wsdl
```

Parameters	**Path Parameters** orgID : Organization ID apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 400 : The API associated with the request does not have a SOAP service. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an API in Swagger 2.0 JSON format

Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 JSON format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+json" to receive the contents formatted as Swagger 2.0 JSON.

Sample method invocation

```
 curl -X GET \
   -H 'Accept: application/vnd.ibm-apim.swagger2+json' \
   https://<mgmt-cluster-ip>/v1/portal/apis/{apiID}
```

Parameters	**Path Parameters** apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an API in Swagger 2.0 JSON format

The API must be available to the specified orgs/{orgID}. Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 JSON format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+json" to receive the contents formatted as Swagger 2.0 JSON.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Accept: application/vnd.ibm-apim.swagger2+json' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apis/{apiID}
```

Parameters	**Path Parameters** orgID : Organization ID apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an API in Swagger 2.0 YAML format

Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 YAML format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+yaml" to receive the contents formatted as Swagger 2.0 YAML.

Sample method invocation

```
curl -X GET \
  -H 'Accept: application/vnd.ibm-apim.swagger2+yaml' \
  https://<mgmt-cluster-ip>/v1/portal/apis/{apiID}
```

Parameters	**Path Parameters** apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an API in Swagger 2.0 YAML format

The API must be available to the specified orgs/{orgID}. Use this REST API to fetch the definition of an API, and its operations, in Swagger 2.0 YAML format. You must pass the header "Accept: application/vnd.ibm-apim.swagger2+yaml" to receive the contents formatted as Swagger 2.0 YAML.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  -H 'Accept: application/vnd.ibm-apim.swagger2+yaml' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apis/{apiID}
```

Parameters	**Path Parameters** orgID : Organization ID apiID : API ID
Response content	**MIME type: application/json** `none`
Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List public APIs

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/apis
```

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     id: string`
	`     url: string`
	`     info:`
	`     {`
	`     }`
	`     protocol: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : The collection of resources was retrieved. 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List APIs

List APIs that are available to the specified orgs/{orgID}

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apis
```

Parameters	**Path Parameters** orgID : Organization ID

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      id: string`
	`      url: string`
	`      info:`
	`      {`
	`      }`
	`      protocol: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : The collection of resources was retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve API documentation file for an API

Retrieve API documentation file for an API, by API's ID and document's ID. Caution: Do not invoke this API in Swagger UI directly.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apis/{apiID}/documents/{docID}/file
```

Parameters	**Path Parameters** orgID : apiID : docID :
Response content	**MIME type: application/json** `none`

Status codes	200 : The resource was retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Application Management APIs

Using APIs to manage applications and subscriptions.

Applications are scoped to a developer organization. All members of a developer organization can perform these actions on any application:

- List all applications
- Create applications
- Get applications
- Update applications
- Delete applications

Likewise, all members of a developer organization can perform these actions on any application credentials:

- Verify the client secret
- Reset the client secret
- Specify user-selected client/application ID and secret.

All members of a developer organization can modify Plan subscriptions for any application. Applications subscribe to Plans to get access to the resources and rate limits published in a Plan. All members of a developer organization can perform these actions on Plan subscriptions:

- Get the list of Plans that an application is subscribed to
- Get the list of applications subscribed to a Plan
- Subscribe an application to a Plan
- Get details of a specific subscription
- Unsubscribe an application to a Plan
- Migrate application subscriptions to another Plan

Application management APIs enable members of a developer organization to be notified when applications reach specified thresholds of rate limits.

- **Create an application**
  Create an application resource associated with orgs/{orgID}. The application is owned by the developer organization, and not by the user that created it. Note that the clientSecret field within the credentials is the plain-text secret, and should be saved immediately by the user. This is the only time when the clientSecret field is given as plain-text to the user. If user forgets, or loses the clientSecret, or it is compromised, the clientSecret can be reset to a new value using the application credentials API.
- **Retrieve an application**
  Retrieve the specified application.
- **Update an application**
  Update an existing application. The only fields that can be updated by the user are: name, description, oauthRedirectURI and public. If other read-only fields of the application are passed during the update operation, they are simply ignored by the server.
- **Delete an application**
  Delete a developer application.
- **List applications**
  Retrieve the collection of applications within the scope of orgs/{orgID}.
- **Reset application client secret**
  Reset the client secret of an application. This API has been superseded by the more generic credentials reset API. The clientSecret in the response is the new plain-text client secret for the application.
- **Verify application client secret**
  Verify whether a client secret is correct.

- **Retrieve application credentials**
  Retrieve application credentials
- **Update application credentials**
  Use this method to update the initial clientID/clientSecret credentials that are defined by default when an application is created. You can update the clientID, clientSecret, or both. Note that when updating the clientID, the API checks that there is no other application credentials resource with the same clientID.
- **Creates new application credentials (clientID/secret pair). A maximum of 20 key pairs can be generated per app.**
  Creates new application credentials (clientID/secret pair). A maximum of 20 key pairs can be generated per app.
- **Resets a specific application's credentials denoted by ID.**
  Resets the client secret and/or client ID of a specific application's credentials. If reset, the clientSecret in the response is the new plain-text client secret for the specific application's credential.
- **Reset application credentials**
  Reset the client secret and/or client ID of an application. If reset, the clientSecret in the response is the new plain-text client secret for the application.
- **Reset a client secret of an application's specific credentials**
  Resets the client secret of an application's specific credential, denoted by Id. This API has been superseded by the more generic credentials reset API. The clientSecret in the response is the new plain-text client secret for the application.
- **Verify client secret for a specific credential id of an application**
  Verify whether a client secret is correct, for a specific application credentialId
- **Deletes a specific credentials of the application**
  Deletes a specific credentials of the application
- **Retrieve a specific credentials of the application**
  Retrieve a specific credentials of the application
- **Update a specific application credential**
  Where an application has multiple pairs of clientID/clientSecret credentials, use this method to update a specific pair of credentials by supplying its credential Id. You can update the clientID, clientSecret, or both. Note that when updating the clientID, the API checks that there is no other credentials resource with the same clientID. The input clientSecret value for this API must be a Base64 encoded SHA-256 hash of the original plain-text clientSecret. Also, note that many tools will automatically encode a SHA-256 value as a hex string so that it is human readable. This API requires that the SHA-256 value be encoded using Base64 instead. You can use the following Linux command to convert a hex encoded SHA-256 value to Base64 encoding: echo "HEX_ENCODED_SHA256_VALUE" | xxd -r -p | base64
- **Delete application image**
  Delete the image associated with a developer application. All GET calls to retrieve the image after its deletion will result in HTTP 204 status code. The imageURL property of the application resource will be shown as null.
- **Replace an application image**
  This API is provided mainly for backwards compatibility with non-scripted browser based clients that have to use HTML forms for file uploads.
- **Replace an application image**
  Replace the application image with a new image
- **Unsubscribe an application from a plan**
  Delete a plan subscription for an application.
- **Retrieve an application subscription**
  Retrieve details of an application plan subscription.
- **List the plan subscriptions for an application**
  Retrieve the collection of plan subscriptions for an application.
- **List subscriptions for a product or a plan of the product**
  Retrieve the list of application subscriptions associated with a product, or a plan of the product.
- **Subscribe an application to a plan**
  An application plan subscription allows the application to invoke API resources that are exposed by the plan. To create a new subscription, specify the plan's name, and either product's ID or product's name and version, but not both.
- **Update an application subscription**
  Update details of an application plan subscription. This API can be used for migrating to new plan version. Specify the plan name, and either product ID or product name and version, but not both. You can migrate a subscription to another plan within the same product, which may require approval. Alternatively, you can migrate a subscription to a superseding plan if available. If the superseding plan requires approval, the subscription will be in the pending state unless previously approved.

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Create an application

Create an application resource associated with orgs/{orgID}. The application is owned by the developer organization, and not by the user that created it. Note that the clientSecret field within the credentials is the plain-text secret, and should be saved immediately by the user. This is the only time when the clientSecret field is given as plain-text to the user. If user forgets, or loses the clientSecret, or it is compromised, the clientSecret can be reset to a new value using the application credentials API.

Sample method invocation

```
 curl -X POST \
   -u userid:pw \
   -H 'Content-Type: application/json' \
   -d '
       {
          "name": string,
          "appImageURL": string,
          "credentials":
          {
             "clientID": enum['true' or ' false'],
             "clientSecret": enum['true' or ' false'],
             "description": string
          }
          ,
          "description": string,
          "oauthRedirectURI": string,
          "public": boolean
       }

      ' \
   https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>

```
 {
    id: string
    name: string
    url: string
    appCredentials: array[
       {
          id: string
          url: string
          clientID: string
          clientSecret: string
          description: string
       }
    ]
    appImageURL: string
    credentials:
    {
       url: string
       clientID: string
       clientSecret: string
       description: string
    }

    description: string
    enabled: boolean
    imageURL: string
    oauthRedirectURI: string
    orgID: string
    promoteTo: enum['PRODUCTION' or 'DEVELOPMENT']
    public: boolean
    state: enum['SUSPENDED' or 'ACTIVE']
    type: enum['PRODUCTION' or 'DEVELOPMENT']
    createdAt: string
    updatedAt: string
 }
```

</td></tr>
<tr><td><b>Status codes</b></td><td>201 : The application was created.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}</td></tr>
</table>

	404 : Resource not found. 409 : There is already an application with the same name. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an application

Retrieve the specified application.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID
appID : Application ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>

```
{
    id: string
    name: string
    url: string
    appCredentials: array[
        {
            id: string
            url: string
            clientID: string
            clientSecret: string
            description: string
        }
    ]
    appImageURL: string
    credentials:
    {
        url: string
        clientID: string
        clientSecret: string
        description: string
    }

    description: string
    enabled: boolean
    imageURL: string
    oauthRedirectURI: string
    orgID: string
    promoteTo: enum['PRODUCTION' or 'DEVELOPMENT']
    public: boolean
    state: enum['SUSPENDED' or 'ACTIVE']
    type: enum['PRODUCTION' or 'DEVELOPMENT']
    createdAt: string
    updatedAt: string
}
```

</td></tr>
<tr><td><b>Status codes</b></td><td>200 : The application was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}</td></tr>
</table>

	404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update an application

Update an existing application. The only fields that can be updated by the user are: name, description, oauthRedirectURI and public. If other read-only fields of the application are passed during the update operation, they are simply ignored by the server.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    -H 'Content-Type: application/json' \
    -d '
        {
           "name": string,
           "appImageURL": string,
           "description": string,
           "oauthRedirectURI": string,
           "public": boolean
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json**

```
 {
    id: string
    name: string
    url: string
    appCredentials: array[
       {
          id: string
          url: string
          clientID: string
          clientSecret: string
          description: string
       }
    ]
    appImageURL: string
    credentials:
    {
       url: string
       clientID: string
       clientSecret: string
       description: string
    }

    description: string
    enabled: boolean
    imageURL: string
    oauthRedirectURI: string
    orgID: string
    promoteTo: enum['PRODUCTION' or 'DEVELOPMENT']
    public: boolean
    state: enum['SUSPENDED' or 'ACTIVE']
    type: enum['PRODUCTION' or 'DEVELOPMENT']
```

	`createdAt: string`
	`updatedAt: string`
	`}`

Status codes	200 : The application was updated. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 409 : There is already an application with the same name. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Delete an application

Delete a developer application.

Sample method invocation

```
curl -X DELETE \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application was deleted. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List applications

Retrieve the collection of applications within the scope of orgs/{orgID}.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre>
[
  {
     id: string
     name: string
     url: string
     appCredentials: array[
        {
           id: string
           url: string
           clientID: string
           clientSecret: string
           description: string
        }
     ]
     appImageURL: string
     credentials:
     {
        url: string
        clientID: string
        clientSecret: string
        description: string
     }

     description: string
     enabled: boolean
     imageURL: string
     oauthRedirectURI: string
     orgID: string
     promoteTo: enum['PRODUCTION' or 'DEVELOPMENT']
     public: boolean
     state: enum['SUSPENDED' or 'ACTIVE']
     type: enum['PRODUCTION' or 'DEVELOPMENT']
     createdAt: string
     updatedAt: string
  }

  .
  .
  .
]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : The collection of applications was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reset application client secret

Reset the client secret of an application. This API has been superseded by the more generic credentials reset API. The clientSecret in the response is the new plain-text client secret for the application.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/reset-secret
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

Response content	**MIME type: application/json**
	`{`
	`   clientSecret: string`
	`}`

Status codes	200 : The application client secret and/or client ID were reset. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Verify application client secret

Verify whether a client secret is correct.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    -H 'Content-Type: application/json' \
    -d '
        {
           "clientSecret": string
        }

      ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/verify-secret
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application credentials' client secret is correct. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 409 : The client secret provided does not match the client secret of the application credentials. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve application credentials

Retrieve application credentials

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

Response content	**MIME type: application/json**
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`
	`}`

Status codes	200 : The application credentials were retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update application credentials

Use this method to update the initial clientID/clientSecret credentials that are defined by default when an application is created. You can update the clientID, clientSecret, or both. Note that when updating the clientID, the API checks that there is no other application credentials resource with the same clientID.

The plain text clientID and clientSecret should contain only unreserved URL characters. Unreserved URL characters are defined by:

Characters that are allowed in a URI but do not have a reserved purpose are called unreserved. These include uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde.

unreserved = ALPHA / DIGIT / "-" / "." / "" / "~"

In regular expression, the unreserved URL characters are [A-Za-z0-9.-~]. For more information, refer to https://tools.ietf.org/html/rfc3986#section-2.3.

The maximum length allowed for plain text clientID or plain text clientSecret is 128 characters.

The input clientSecret value for this API must be a Base64 encoded SHA-256 hash of the original plain-text clientSecret. Also, note that many tools will automatically encode a SHA-256 value as a hex string so that it is human readable. This API requires that the SHA-256 value be encoded using Base64 instead. You can use the following Linux command to convert a hex encoded SHA-256 value to Base64 encoding: echo "HEX_ENCODED_SHA256_VALUE" | xxd -r -p | base64

Sample method invocation

```
 curl -X PUT \
   -u userid:pw \
   -d '
       {
          "clientID": string,
          "clientSecret": string,
          "description": string
       }

      ' \
   https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

Response content	**MIME type: application/json**
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`
	`}`

Status codes	200 : The application credentials were updated. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 409 : There is already another application credentials resource with same clientID. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Creates new application credentials (clientID/secret pair). A maximum of 20 key pairs can be generated per app.

Creates new application credentials (clientID/secret pair). A maximum of 20 key pairs can be generated per app.

Sample method invocation

```
 curl -X POST \
   -u userid:pw \
   -d '
       {
          "clientID": enum['true' or ' false'],
          "clientSecret": enum['true' or ' false'],
          "description": string
```

```
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials
```

Parameters	Path Parameters orgID : Organization ID appID : Application ID

Response content	MIME type: application/json
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`
	`}`

Status codes	201 : Application credential has been created 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Resets a specific application's credentials denoted by ID.

Resets the client secret and/or client ID of a specific application's credentials. If reset, the clientSecret in the response is the new plain-text client secret for the specific application's credential.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    -d '
        {
           "clientID": enum['true' or ' false'],
           "clientSecret": enum['true' or ' false']
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}/reset
```

Parameters	Path Parameters orgID : Organization ID appID : Application ID credentialId : Application Credential ID

Response content	MIME type: application/json
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`

	`}`
Status codes	200 : The specific credential's client secret and/or client ID were reset. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reset application credentials

Reset the client secret and/or client ID of an application. If reset, the clientSecret in the response is the new plain-text client secret for the application.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    -d '
        {
           "clientID": enum['true' or ' false'],
           "clientSecret": enum['true' or ' false']
        }

      ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/reset
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

Response content	**MIME type: application/json**
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`
	`}`

Status codes	200 : The application client secret and/or client ID were reset. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Reset a client secret of an application's specific credentials

Resets the client secret of an application's specific credential, denoted by Id. This API has been superseded by the more generic credentials reset API. The clientSecret in the response is the new plain-text client secret for the application.

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}/reset-
secret
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID credentialId : Application Credential ID

Response content	**MIME type: application/json**
	`{`
	`    clientSecret: string`
	`}`

Status codes	200 : The application credentialId's client secret was reset. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Verify client secret for a specific credential id of an application

Verify whether a client secret is correct, for a specific application credentialId

Sample method invocation

```
  curl -X PUT \
    -u userid:pw \
    -H 'Content-Type: application/json' \
    -d '
        {
           "clientSecret": string
        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}/verify-
secret
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

	credentialId : Application Credential ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application credentialId's client secret is correct. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 409 : The client secret provided does not match the application credentialId's client secret. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Deletes a specific credentials of the application

Deletes a specific credentials of the application

Sample method invocation

```
curl -X DELETE \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID credentialId : Application Credential ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application credentials specific to a credential Id are deleted 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve a specific credentials of the application

Retrieve a specific credentials of the application

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID credentialId : Application Credential ID

Response content	**MIME type: application/json**
	`{`
	`id: string`
	`url: string`
	`clientID: string`
	`clientSecret: string`
	`description: string`
	`}`

Status codes	200 : The application credentials specific to a credential Id were retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update a specific application credential

Where an application has multiple pairs of clientID/clientSecret credentials, use this method to update a specific pair of credentials by supplying its credential Id. You can update the clientID, clientSecret, or both. Note that when updating the clientID, the API checks that there is no other credentials resource with the same clientID. The input clientSecret value for this API must be a Base64 encoded SHA-256 hash of the original plain-text clientSecret. Also, note that many tools will automatically encode a SHA-256 value as a hex string so that it is human readable. This API requires that the SHA-256 value be encoded using Base64 instead. You can use the following Linux command to convert a hex encoded SHA-256 value to Base64 encoding: echo "HEX_ENCODED_SHA256_VALUE" | xxd -r -p | base64

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
  -d '
      {
         "clientID": string,
         "clientSecret": string,
         "description": string
      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/credentials/{credentialId}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID

	credentialId : Application Credential ID

Response content	**MIME type: application/json**
	`{`
	`    url: string`
	`    clientID: string`
	`    clientSecret: string`
	`    description: string`
	`}`

Status codes	200 : The specific application credentials were updated. 400 : Invalid or missing request payload properties. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 409 : There is already another application credentials resource with same clientID. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Delete application image

Delete the image associated with a developer application. All GET calls to retrieve the image after its deletion will result in HTTP 204 status code. The imageURL property of the application resource will be shown as null.

Sample method invocation

```
 curl -X DELETE \
   -u userid:pw \
   -H 'Content-Type: multipart/form-data' \
   https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/image
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application image was deleted. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Replace an application image

This API is provided mainly for backwards compatibility with non-scripted browser based clients that have to use HTML forms for file uploads.

Sample method invocation

```
curl -X POST \
  -u userid:pw \
  -H 'Content-Type: multipart/form-data' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/image
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application image was updated 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Replace an application image

Replace the application image with a new image

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
  -H 'Content-Type: multipart/form-data' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/image
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The application image was updated 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.

Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Unsubscribe an application from a plan

Delete a plan subscription for an application.

Sample method invocation

```
curl -X DELETE \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/subscriptions/{subID}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID subID : Subscription ID
Response content	**MIME type: application/json** `none`
Status codes	204 : The plan subscription was deleted. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Retrieve an application subscription

Retrieve details of an application plan subscription.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/subscriptions/{subID}
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID subID : Subscription ID
Response content	**MIME type: application/json** `{`

<table>
<tr><td></td><td><pre>
    id: string
    url: string
    apis: array[
       {
          basePath: string
          info:
          {
          }

          operations: array[
             {
                operation: string
                path: string
             }
          ]
       }
    ]
    app:
    {
       id: string
       name: string
    }

    approved: boolean
    plan: string
    product:
    {
       id: string
       name: string
       version: string
    }

    supersededBy:
    {
       plan: string
       product: null
    }

    createdAt: string
    updatedAt: string
 }
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>201 : The subscription was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List the plan subscriptions for an application

Retrieve the collection of plan subscriptions for an application.

Sample method invocation

```
 curl -X GET \
   -u userid:pw \
   https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/subscriptions
```

Parameters	**Path Parameters** orgID : Organization ID

<table>
<tr><td></td><td>appID : Application ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre>
[
  {
     id: string
     url: string
     apis: array[
        {
           basePath: string
           info:
           {
           }

           operations: array[
              {
                 operation: string
                 path: string
              }
           ]
        }
     ]
     app:
     {
        id: string
        name: string
     }

     approved: boolean
     plan: string
     product:
     {
        id: string
        name: string
        version: string
     }

     supersededBy:
     {
        plan: string
        product: null
     }

     createdAt: string
     updatedAt: string
  }

  .
  .
  .
]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : The subscription was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

List subscriptions for a product or a plan of the product

Retrieve the list of application subscriptions associated with a product, or a plan of the product.

Sample method invocation

```
curl -X GET \
  -u userid:pw \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/products/{productID}/subscriptions
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
productID : product ID
orgID : Organization ID
<b>Query Parameters</b>
plan : plan's name</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre>
[
  {
     id: string
     url: string
     apis: array[
        {
           basePath: string
           info:
           {
           }

           operations: array[
              {
                 operation: string
                 path: string
              }
           ]
        }
     ]
     app:
     {
        id: string
        name: string
     }

     approved: boolean
     plan: string
     product:
     {
        id: string
        name: string
        version: string
     }

     supersededBy:
     {
        plan: string
        product: null
     }

     createdAt: string
     updatedAt: string
  }

  .
  .
  .
]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : The resource was retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
405 : Consumer org is currently not allowed to subscribe the plan.
500 : An internal error has occurred.</td></tr>
</table>

Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Subscribe an application to a plan

An application plan subscription allows the application to invoke API resources that are exposed by the plan. To create a new subscription, specify the plan's name, and either product's ID or product's name and version, but not both.

Sample method invocation

```
  curl -X POST \
    -u userid:pw \
    -d '
        {
           "plan": string,
           "product": 
           {
              "id": string,
              "name": string,
              "version": string
           }

        }

       ' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/subscriptions
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID

appID : Application ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>

```
 {
    id: string
    url: string
    apis: array[
       {
          basePath: string
          info: 
          {
          }

          operations: array[
             {
                operation: string
                path: string
             }
          ]
       }
    ]
    app: 
    {
       id: string
       name: string
    }

    approved: boolean
    plan: string
    product: 
    {
       id: string
       name: string
       version: string
    }
```

</td></tr>
</table>

<table>
<tr><td></td><td><pre>
    supersededBy:
    {
       plan: string
       product: null
    }

    createdAt: string
    updatedAt: string
 }
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>201 : The application has subscribed to the plan successfully.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Update an application subscription

Update details of an application plan subscription. This API can be used for migrating to new plan version. Specify the plan name, and either product ID or product name and version, but not both. You can migrate a subscription to another plan within the same product, which may require approval. Alternatively, you can migrate a subscription to a superseding plan if available. If the superseding plan requires approval, the subscription will be in the pending state unless previously approved.

Sample method invocation

```
curl -X PUT \
  -u userid:pw \
  -d '
      {
         "plan": string,
         "product":
         {
            "id": string,
            "name": string,
            "version": string
         }

      }

     ' \
  https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/apps/{appID}/subscriptions/{subID}
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID
appID : Application ID
subID : Subscription ID</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b><pre>

 {
    id: string
    url: string
    apis: array[
       {
          basePath: string
          info:
          {
</pre></td></tr>
</table>

```
         }
         operations: array[
            {
               operation: string
               path: string
            }
         ]
      }
   ]
   app:
   {
      id: string
      name: string
   }

   approved: boolean
   plan: string
   product:
   {
      id: string
      name: string
      version: string
   }

   supersededBy:
   {
      plan: string
      product: null
   }

   createdAt: string
   updatedAt: string
}
```

Status codes	200 : The subscription was updated. 400 : Plan subscription cannot be updated because there is already a pending request for another plan. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Configuration APIs

Using APIs to obtain details about a portal configuration.

Use the Configuration APIs to obtain details about a portal configuration. You can perform the following tasks.

- Query a portal configuration.
- **Get portal configuration by originURL**
  This API is used for looking up the configuration details of a portal based on its originURL. The originURL is whatever the user enters in the URL browser address bar to arrive at the portal. The search algorithm is lenient enough to allow for variations of the originURL. Before searching for matches, the query, and fragment sections are removed. The algorithm also looks for variations with, and without the default port numbers. (i.e. 80 for http://, and 443 for https://)

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Get portal configuration by originURL

This API is used for looking up the configuration details of a portal based on its originURL. The originURL is whatever the user enters in the URL browser address bar to arrive at the portal. The search algorithm is lenient enough to allow for variations of the originURL. Before searching for matches, the query, and fragment sections are removed. The algorithm also looks for variations with, and without the default port numbers. (i.e. 80 for http://, and 443 for https://)

Sample method invocation

```
curl -X GET \
  https://<mgmt-cluster-ip>/v1/portal/config
```

<table>
<tr><td><b>Parameters</b></td><td><b>Query Parameters</b>

originURL : The url of the portal</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre>
{
   applicationLifecycleEnabled: boolean
   customGatewayURL: string
   customPortalURL: string
   envID: string
   envName: string
   expired: boolean
   gatewayURL: string
   invitationEnabled: boolean
   orgDisplayName: string
   orgID: string
   orgName: string
   paymentGateways: array[
      {
         id: string
         url: string
         envId: string
         orgId: string
         type: enum['stripe' or ' custom']
         createdAt: string
         createdBy: string
         updatedAt: string
         updatedBy: string
      }
   ]
   portalAPIHost: string
   portalURL: string
   selfSignUpEnabled: boolean
   userRegistry:
   {
      id: string
      inviteViaEmailOnly: boolean
      isCaseSensitive: boolean
      realm: string
      type: string
      usernameIsAlwaysEmail: boolean
      writable: boolean
   }

}
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : The configuration resource was successuflly retrieved

204 : No configuration found associated with the specified originURL

400 : The originURL query parameter is not valid.

500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Analytics

Analyzing Catalog APIs.

Catalog APIs allow members of a developer organization to analyze applications and APIs in the following ways:

- Get five most used APIs
- Get five most active Applications
- Get health status for Applications
- Get success-rates, latencies, and data usage for applications, APIs and Plans

- **Return top five APIs**
- **Return data usage information for all the resources used by a given application**
- **Return data usage information for all the resources used by all applications in the given organization**
- **Return latency information for all the resources used by a given application**
- **Return latency information for all the resources used by all applications in the given organization**
- **Return success and failure rates for all the resources used by a given application**
- **Return success and failure rates for all the resources used by all applications in the given organization**
- **Return a list of applications with their health status (green, yellow, red)**
- **Return top five applications**
- **Return call histograms for top five APIs.**
- **Return call histograms for top five apps.**
- **Return combined data usage for all resources used by a given application**
- **Return combined data usage information for all the resources used by a given organization**
- **Return combined latency information for all the resources used by a given application**
- **Return combined latency information for all the resources used by a given organization**
- **Return combined success and failure rates for all the resources used by a given application**
- **Return combined success and failure rates for all the resources used by a given organization**

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return top five APIs

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apis/top?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** timeRange : Time range to retrieve analytics data for (week, month)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     id: string`
	`     name: string`
	`     hits: integer`

<table>
<tr><td></td><td><pre>
        href: string
    }

    .
    .
    .
]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : Top five APIs retrieved.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return data usage information for all the resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/data-usage?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID
appID : Application ID
<b>Query Parameters</b>
units : Data unit (B, KB, MB, GB)
V5.0.8 + useBytesSent: Beginning in version 5.0.8.3, when set to <i>true</i>, the analytics field <b>bytes_sent</b> is used to calculate the data usages instead of the analytics field <b>bytes_received</b>.
timeZoneOffset : Time zone offset (minute)
timeRange : Time range to retrieve analytics data for (week, month)</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b><pre>
[
    {
        id: string
        url: string
        dataUsages: array[
            {
                avg: string
                datetime: string
                max: string
                min: string
            }
        ]
        path: string
        verb: string
    }

    .
    .
</pre></td></tr>
</table>

<table>
<tr><td></td><td><pre> .
]</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : Data usage information retrieved for all resource used by this application.

401 : Authentication failed. Valid username and password are required.

403 : Authenticated user is not a member of orgs/{orgID}

404 : Resource not found.

500 : An internal error has occurred.</td></tr>
<tr><td><b>Available since</b></td><td>API Connect 5.0.0</td></tr>
</table>

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return data usage information for all the resources used by all applications in the given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/data-usage?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID

<b>Query Parameters</b>

units : Data unit (B, KB, MB, GB)

V5.0.8 + useBytesSent: Beginning in version 5.0.8.3, when set to <i>true</i>, the analytics field <code>bytes_sent</code> is used to calculate the data usages instead of the analytics field <code>bytes_received</code>.

timeZoneOffset : Time zone offset (minute)

timeRange : Time range to retrieve analytics data for (week, month)</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b>
<pre> [
   {
      id: string
      url: string
      dataUsages: array[
         {
            avg: string
            datetime: string
            max: string
            min: string
         }
      ]
      path: string
      verb: string
   }

   .
   .
   .
 ]</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : Data usage information retrieved for all resources used by all applications in the given organization.

401 : Authentication failed. Valid username and password are required.</td></tr>
</table>

	403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return latency information for all the resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/latencies?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

<table>
<tr><td><b>Parameters</b></td><td><b>Path Parameters</b>
orgID : Organization ID
appID : Application ID
<b>Query Parameters</b>
resourceID : Resource ID
timeZoneOffset : Time zone offset (minute)
timeRange : Time range to retrieve analytics data for (week, month)
interval : Interval for granularity of histogram results (second, minute, hour, day)</td></tr>
<tr><td><b>Response content</b></td><td><b>MIME type: application/json</b><pre>
 [
   {
      id: string
      url: string
      latencies: array[
         {
            avg: string
            datetime: string
            max: string
            min: string
         }
      ]
      path: string
      verb: string
   }

   .
   .
   .
 ]
</pre></td></tr>
<tr><td><b>Status codes</b></td><td>200 : Latency information retrieved for all resource used by this application.
401 : Authentication failed. Valid username and password are required.
403 : Authenticated user is not a member of orgs/{orgID}
404 : Resource not found.
500 : An internal error has occurred.</td></tr>
</table>

Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return latency information for all the resources used by all applications in the given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/latencies?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** resourceID : Resource ID timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      id: string`
	`      url: string`
	`      latencies: array[`
	`          {`
	`              avg: string`
	`              datetime: string`
	`              max: string`
	`              min: string`
	`          }`
	`      ]`
	`      path: string`
	`      verb: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Latency information retrieved for all resources used by all applications in the given organization. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return success and failure rates for all the resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/success-rates?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID **Query Parameters** resourceID : Resource ID timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     id: string`
	`     url: string`
	`     path: string`
	`     successRates: array[`
	`        {`
	`           datetime: string`
	`           failure: string`
	`           success: string`
	`           total: string`
	`        }`
	`     ]`
	`     verb: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Success and failure rates retrieved for all resources used by this application. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return success and failure rates for all the resources used by all applications in the given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/success-rates?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** resourceID : Resource ID timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     id: string`
	`     url: string`
	`     path: string`
	`     successRates: array[`
	`        {`
	`           datetime: string`
	`           failure: string`
	`           success: string`
	`           total: string`
	`        }`
	`     ]`
	`     verb: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Success and failure rates retrieved for all resources used by all applications in the given organization. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return a list of applications with their health status (green, yellow, red)

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/health
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** timeZoneOffset : Time zone offset (minute)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     url: string`
	`     appId: string`
	`     appName: string`
	`     health: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Application health retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details. For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return top five applications

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/top?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** timeRange : Time range to retrieve analytics data for (week, month)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     id: string`
	`     name: string`
	`     hits: integer`
	`     href: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Top five apps retrieved. 401 : Authentication failed. Valid username and password are required.

	403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return call histograms for top five APIs.

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apis/top/histogram?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** timeRange : Time range to retrieve analytics data for (week, month) timeZoneOffset : Time zone offset (minute) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      id: string`
	`      name: string`
	`      histogram: array[`
	`         {`
	`            count: integer`
	`            datetime: string`
	`         }`
	`      ]`
	`      href: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Call histograms for top five APIs retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return call histograms for top five apps.

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/top/histogram?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** timeRange : Time range to retrieve analytics data for (week, month) timeZoneOffset : Time zone offset (minute) interval : Interval for granularity of histogram results (second, minute, hour, day)
Response content	**MIME type: application/json** `[ { id: string name: string histogram: array[ { count: integer datetime: string } ] href: string } . . . ]`
Status codes	200 : Call histograms for top five apps retrieved. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined data usage for all resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/data-usage/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) V5.0.8 + useBytesSent: Beginning in version 5.0.8.3, when set to *true*, the analytics field **`bytes_sent`** is used to calculate the data usages instead of the analytics field **`bytes_received`**.

Response content	**MIME type: application/json**
	[
	  {
	      datetime: string
	      failure: string
	      success: string
	      total: string
	  }
	  .
	  .
	  .
	]

Status codes	200 : Combined data usage information retrieved for all resources used by this application. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined data usage information for all the resources used by a given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/data-usage/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month)

	interval : Interval for granularity of histogram results (second, minute, hour, day) V5.0.8 + useBytesSent: Beginning in version 5.0.8.3, when set to *true*, the analytics field **`bytes_sent`** is used to calculate the data usages instead of the analytics field **`bytes_received`**.

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     datetime: string`
	`     failure: string`
	`     success: string`
	`     total: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Combined data usage information retrieved for all resources used by this organization. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined latency information for all the resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/latencies/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     datetime: string`
	`     failure: string`
	`     success: string`
	`     total: string`
	`  }`

	`.`
	`.`
	`.`
	`]`

Status codes	200 : Combined latency information retrieved for all the resources used by this application. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined latency information for all the resources used by a given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/latencies/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`      datetime: string`
	`      failure: string`
	`      success: string`
	`      total: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Combined latency information retrieved for all resources used by this organization. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined success and failure rates for all the resources used by a given application

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/apps/{appID}/success-rates/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID appID : Application ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     datetime: string`
	`     failure: string`
	`     success: string`
	`     total: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Combined success and failure rates retrieved for all resources used by this application. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0

Note: IBM API Connect 5.0.x was EOS after 30 April 2022. See support policy for details.
For a more recent version, see the IBM API Connect 10.0.5.x product documentation.

Return combined success and failure rates for all the resources used by a given organization

Sample method invocation

```
  curl -X GET \
    -u userid:pw \
    -H 'Accept: application/json' \
    https://<mgmt-cluster-ip>/v1/portal/orgs/{orgID}/analytics/success-rates/all?timeRange=
<start_datetime>%20TO%20<end_datetime>
```

Parameters	**Path Parameters** orgID : Organization ID **Query Parameters** units : Data unit (B, KB, MB, GB) timeZoneOffset : Time zone offset (minute) timeRange : Time range to retrieve analytics data for (week, month) interval : Interval for granularity of histogram results (second, minute, hour, day)

Response content	**MIME type: application/json**
	`[`
	`  {`
	`     datetime: string`
	`     failure: string`
	`     success: string`
	`     total: string`
	`  }`
	`  .`
	`  .`
	`  .`
	`]`

Status codes	200 : Combined success and failure rates retrieved for all resources used by this organization. 401 : Authentication failed. Valid username and password are required. 403 : Authenticated user is not a member of orgs/{orgID} 404 : Resource not found. 500 : An internal error has occurred.
Available since	API Connect 5.0.0