Informatica

Informatica® Intelligent Cloud Services
July 2022

API Portal Guide

Informatica Intelligent Cloud Services API Portal Guide
July 2022
July 2022

Publication Date: 2022-07-06

Table of Contents

Preface

Use the *API Portal Guide* to learn how to use API Portal. Learn how to perform API administrative tasks and how to view API groups and group details.

CHAPTER 1

API Portal Overview

API Portal provides API consumers with secure access to managed APIs and custom APIs. API consumers can see details of an API such as status, authentication type, and applicable access control policy, and drill down to further details with Swagger or WSDL.

API consumers can use API Portal to generate software development kit (SDK) packages to facilitate rapid integration of managed APIs and custom APIs into applications. Consumers can generate SDK packages that provide a set of resources to enable integration for Java, Android, Javascript, Nodejs, Python, Ruby-on-Rails, C#.NET, ASP.NET5, or C# applications.

API consumers can also use API Portal to interact with managed APIs and custom APIs, and view API usage analytics.

Before You Start

Before you use API Portal, ensure that you have an active Informatica Intelligent Cloud Services account for the organization and have an API Manager license.

To access API Portal, the organization administrator must assign you the **Service Consumer** privilege. You must also be assigned the **View API Portal** privilege in API Manager.

For more information about registration and roles, refer to the *Administrator* help.

Accessing API Portal

Access API Portal from the Informatica Intelligent Cloud Services **My Services** page.

1. On the Informatica Intelligent Cloud Services login page, enter your user name and password.
2. Click **Log In**.

   The **My Services** page appears.
3. Select **API Portal**.

   API Portal appears.

CHAPTER 2

Analytics

API analytics provide a graphical overview of activity and API usage, and appear on the **Home** page of the API Portal. The analytics dashboard contains panels with reports about managed APIs and custom APIs. Use the dashboard to view visual summary information about APIs, such as trends in usage over time and APIs with the most invocations.

You can view your API usage activity for any authenticated managed API. You can see which authenticated managed APIs are most frequently invoked on your behalf.

Note: Analytics are based on the user name that is provided as part of the Authorization HTTP header in an authenticated API. The statistics reflect the number of API invocations executed for each API which contains your user name in the Authorization header.

API usage trends

The API Portal **Home** page shows API usage trends for a selected period, for 7, 30, or 90 days.

You can refresh the data by clicking the refresh icon. The last time that you refreshed the data appears near the icon. Data from the current day appears after a delay of half an hour.

Top APIs

The API Portal **Home** page shows the APIs most frequently invoked in the selected period, ranked by number of invocations.

The following table describes the properties of the My Top APIs report:

Panel	Description
API Name	Name of the managed API.
API URL	Identifies the URL of the managed API that was invoked.

Panel	Description
Protocol	Protocol of the managed API. A managed API can use one of the following protocols: - REST - SOAP
Invocations	Number of times the invoked URLs for authenticated managed APIs. **Note:** Analytics are based on the user name that is provided as part of the Authorization HTTP header in an authenticated API.

CHAPTER 3

API administration

When you work with managed APIs and custom APIs, you might want to view the APIs that are available in API Portal, and perform tasks that help you integrate the managed APIs and custom APIs in your applications and invoke them.

The API registry contains the managed APIs and custom APIs that you can view in API Portal with their details.

You can perform the following tasks in the API registry:

- View the managed APIs and custom APIs that are available in API Portal.
- View the metadata of a managed API.
- Copy the URL of a managed API or custom API and use it in your applications to invoke the API.
- Generate and export an SDK package for a managed API.
- Search for a managed API or custom API.
- Interactively test a managed API.

Viewing available APIs

View the managed APIs and custom APIs that are available in API Portal on the **API Registry** page.

The following table describes the properties that the **API Registry** page shows for each API:

Property	Description
Icon	Identifies whether the entity is a managed API or custom API: - ⚙: Designates a managed API. - ⚙: Designates a custom API.
Name	Name of the API.
Version	Version of the API. Different versions of a managed API point to different Cloud Application Integration processes.
Service Name	Name of the Informatica Cloud Application Integration process that the managed API points to. This field is empty for custom APIs.

Property	Description
Protocol	Protocol of the API. An API can use one of the following protocols: - REST - SOAP
Status	Status of the API. An API can have the following statuses: - ✅ Active. The API is active. - ❌ Inactive. The API is currently not available. - ✓ Service not available. The Informatica Cloud Application Integration process that the managed API points to is unavailable or deleted. The managed API is greyed out.
Authentication Method	API authentication method. A managed API can use one of the following authentication methods: - Anonymous. You invoke the managed API without authentication. - Basic. To invoke the managed API, you authenticate with an Informatica Intelligent Cloud Services user name and password. - OAuth 2.0. You receive OAuth 2.0 client credentials from the API Portal administrator and use them to generate an OAuth 2.0 authorization token that you then use to invoke the managed API. - JSON Web Token (JWT). You generate a token in API Portal and use it to invoke the managed API. You can generate a token for a single managed API, for multiple managed APIs, and for a group of APIs. A custom API uses anonymous authentication.
Group	Group that the API belongs to.
Description	Description of the API.

Viewing managed API details and metadata

View API details, copy the API URL, and copy the URL of the page that contains API metadata details on the managed API page.

1. On the **API Registry** page, click the managed API.

   The managed API page appears, showing the managed API details.

2. To copy the URL of the managed API, click **Copy URL**.
3. To obtain the URL to view metadata details for the managed API, copy the URL of the page that contains API metadata details. One of the following buttons appears on the page, based on the protocol of the managed API:
   - If the managed API is a REST API, click **Copy Swagger URL**.
   - If the managed API is a SOAP API, click **Copy WSDL URL**.

   The URL is copied to the clipboard.

4. Use one of the following methods to view metadata details for the managed API:
   - If the managed API uses anonymous or basic authentication, paste the URL in your browser.
   - If the managed API uses OAuth 2.0 or JSON Web Token (JWT) authentication, copy the URL to an application or service that supports token authentication such as Postman or SoapUI.

API properties

View the API properties on the API page.

The following table describes the properties of the API page:

Property	Description
Name	Name of the API.
Version	Version of the API. Different versions of a managed API point to different Cloud Application Integration processes.
Service Name	Name of the Informatica Cloud Application Integration process that the managed API points to. This field is empty for custom APIs.
Protocol	Protocol of the API. An API can use one of the following protocols: - REST - SOAP
Status	Status of the API. An API can have the following statuses: - ☑ Active. The API is active. - ⊗ Inactive. The API is currently not available. - ✓ Service not available. The Informatica Cloud Application Integration process that the managed API points to is unavailable or deleted. The managed API is greyed out.
Authentication Method	API authentication method. A managed API can use one of the following authentication methods: - Anonymous. You invoke the managed API without authentication. - Basic. To invoke the managed API, you authenticate with an Informatica Intelligent Cloud Services user name and password. - OAuth 2.0. You receive OAuth 2.0 client credentials from the API Portal administrator and use them to generate an OAuth 2.0 authorization token that you then use to invoke the managed API. - JSON Web Token (JWT). You generate a token in API Portal and use it to invoke the managed API. You can generate a token for a single managed API, for multiple managed APIs, and for a group of APIs. A custom API uses anonymous authentication.
Group	Group that the API belongs to.
Description	Description of the API.

Obtaining an API URL

To invoke a manage API or custom API, copy the API URL and use it in your applications.

Perform one of the following tasks to copy the URL:

- On the API details page click **Copy URL**.

- On the **API Registry** page perform the following actions:
  1. Select a managed API or custom API.
  2. Click to open the Actions menu, and select **Copy URL**.

The URL is available for pasting in the clipboard.

Generating and exporting an SDK package

You can generate and export an SDK package for a managed API. You can use the SDK package to integrate the managed API into your applications.

1. On the **API Registry** page, click to select a managed API.

   The API details window appears.
2. Select the type of client SDK package that you want to generate.
3. Click **Download**.

   The **API Registry** window appears.
4. Enter your API Portal user authorization details.

   The SDK package downloads to your host machine. To obtain information about the SDK package, read the `readme.md` file.

Searching for an API

You can search for a managed API or custom API by sorting columns or searching for specific text.

Perform one of the following tasks:

- To sort the APIs according to a specific property, on the **API Registry** tab, click the column picker icon to the left of the **Find** field and then select the column to sort. The API Registry table shows the sorted APIs.
- To search for APIs according to specific text, in the **Find** field, type the text for which to search. The search is performed on all columns. The API Registry table shows the relevant APIs.

Interactively testing a managed API

You can interactively test a managed API created for a REST API in a Swagger interface. You can view the API URL, the HTTP status codes, the request parameters, and the response parameters. You can also execute the API for testing purposes, or get a sample cURL command.

1. In the **API Registry** page, click to select a managed API, or click to open the Actions menu and select **View API Details**.

   The API details window appears.
2. Select the **Swagger** tab.

3. If the managed API requires authentication, the **Authorization** dialog box appears. You might need to perform one of the following actions based on the authentication type that the managed API uses:
    - If the managed API uses JSON Web Token (JWT) authentication, no action is required. API Manager generates a JWT token and authenticates to the managed API.
    - If the managed API doesn't use JWT authentication and uses basic authentication, enter the username and password of a user who is authorized to access the managed API.
    - If the managed API uses only OAuth 2.0 authentication, enter the credentials of an OAuth 2.0 client that allows access to the managed API.
4. To expand the view in the **Swagger** tab, click the arrows in the upper right corner.
5. To view the API request body and response codes, click any button that displays an API method.

    For example, for an API with a POST method, a **POST** button is displayed. Click the **POST** button to view the API request body and response code. The input fields are displayed in the request body and the output fields are displayed in the response body with the data types and field descriptions.
6. To view the request body in JSON format, select **application/json**. To view the request body in XML format, select **application/xml**.
7. To test the API semantics, in the request body panel, perform the following steps:
    a. Click **Try it out**.
    b. Edit the request body. Replace any parameter type with a value.

        Note: If a query parameter in a managed API URL includes spaces, the access request fails. Use `POST` to pass a query parameter that includes spaces.
    c. To test the updated request body, click **Execute**.

        The **Server response** panel displays the response body, response headers, and request duration time.
    d. To clear the server response, click **Clear**.
    e. To cancel the request body changes, click **Cancel**. To change the request body again, click **Edit**.
8. To view the request or response syntax, in the **Models** panel, click the right arrow near the request or response entry.

    The model request or response body is displayed. A red asterisk next to an element indicates a required element. The input fields are displayed in the request body and the output fields are displayed in the response body with the data types and field descriptions.

CHAPTER 4

API groups

API groups sort managed APIs and custom APIs into logical groups for easy management of the APIs in API Portal. An API can belong to only one group.

You can view API groups and group details, including details about the APIs that belong to the group.

You can generate JSON web tokens for managed API groups. When you generate a token for a group, you can use the token to invoke the managed APIs in the group that use JSON Web Token (JWT) authentication. For more information, see Chapter 5, "Authentication and authorization" on page 15.

Viewing API groups and group details

You can view API groups and group details in the **API Groups** page.

The **API Groups** page shows general details about each API group, including the following properties:

Property	Description
Name	Name of the group.
Context	Group context. API Manager adds the context to the API URLs of the APIs that belong to the group.
Description	Description of the group.

To view more details about an API group, click the group name to open the group page. The group page shows the general group details, and a list of the APIs that belong to the group. If the group uses JSON web tokens, the group page includes a **JWT Access Token** area, where you can generate a token for the group.

The following table describes the properties that appear on the group page for each API that belongs to the group:

Property	Description
Name	Name of the API.
Version	Version of the API. Different versions of a managed API point to different Cloud Application Integration processes.
Service Name	Name of the Informatica Cloud Application Integration process that the managed API points to. This field is empty for custom APIs.

Property	Description
Protocol	Protocol of the API. An API can use one of the following protocols: - REST - SOAP
Status	Status of the API. An API can have the following statuses: - ✅ Active. The API is active. - ❌ Inactive. The API is currently not available. - ✓ Service not available. The Informatica Cloud Application Integration process that the managed API points to is unavailable or deleted. The managed API is greyed out.
Description	Description of the API.

CHAPTER 5

Authentication and authorization

Authentication and authorization control access to managed APIs that are available in API Portal.

A managed API can use the following types of authentication and authorization:

- Basic. To invoke the managed API, you authenticate with an Informatica Intelligent Cloud Services user name and password.
- OAuth 2.0. You receive OAuth 2.0 client credentials from the API Portal administrator and use them to generate an OAuth 2.0 authorization token that you then use to invoke the managed API.
- JSON Web Token (JWT). You generate a token in API Portal and use it to invoke the managed API. You can generate a token for a single managed API, for multiple managed APIs, and for a group of APIs.

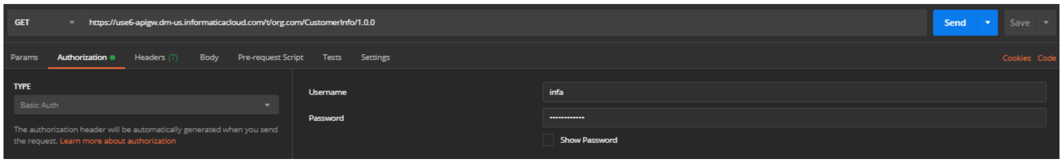
You can view the authentication type or types that a managed API uses on the **API Registry** page. When a managed API uses more than one authentication type, you can use any of the relevant methods to invoke the managed API.

A custom API does not use authentication.

Invoke a managed API with basic authentication

To invoke a managed API that uses basic authentication, you authenticate to the API with an Informatica Intelligent Cloud Services user name and password.

The following image shows an API invoked through Postman with the authorization type set to **Basic Auth**, and the user name and password specified:



GET https://use6-apigw.dm-us.informaticacloud.com/t/org.com/CustomerInfo/1.0.0

TYPE: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization

Username: infa

Password: ············

☐ Show Password

Invoke a managed API with OAuth 2.0 authentication and authorization

To invoke a managed API that uses OAuth 2.0 authentication, you generate an OAuth 2.0 authorization token and send the token to the managed API.

The following sections describe the stages of invoking a managed API that uses OAuth 2.0 authentication:

Generating an OAuth 2.0 authorization token

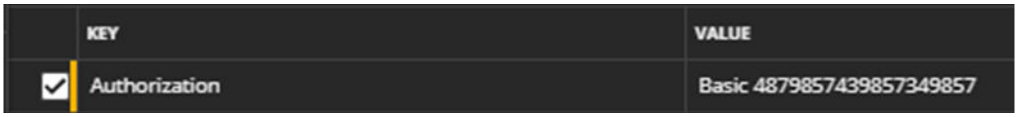
To generate the token, you authenticate to the Informatica Intelligent Cloud Services OAuth 2.0 server using the server URL and the OAuth 2.0 client credentials that you receive from the API Portal administrator.

Note: The token is valid for 15 minutes. After 15 minutes you have to generate a new token.

You use one of the following methods to provide the client credentials to the OAuth 2.0 server, based on the application or software package that you use to invoke the API:

- Enter the OAuth 2.0 client ID and secret separately, as plain text. For example, in Postman, enter the client name in the **Username** field and the client secret ID in the **Password** field.
- Enter the client credentials as an authentication header value in a Basic authorization header.

The following image shows an API invocation through Postman with a Basic authorization header:

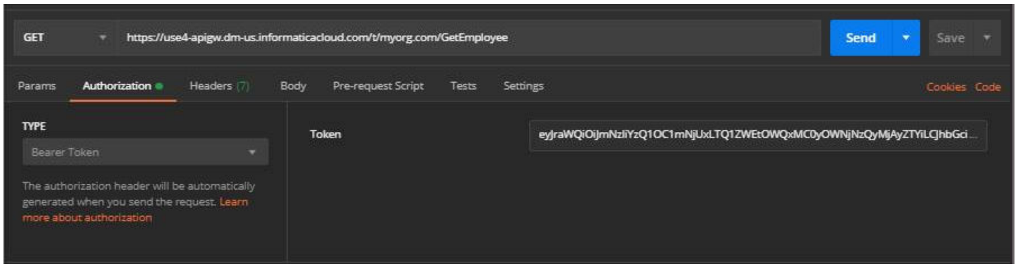


	KEY	VALUE
☑	Authorization	Basic 4879857439857349857

Sending the token to the managed API

You pass the token that you receive from the OAuth 2.0 server to the managed API as an Authorization header with the prefix `Bearer` followed by the token.

The following image shows an API invoked through Postman with a Bearer Token authorization type and the token specified:



https://use4-apigw.dm-us.informaticacloud.com/t/myorg.com/GetEmployee

TYPE: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization

Token: eyJraWQiOiJmNzliYzQ1OC1mNjUxLTQ1ZWEtOWQxMC0yOWNjNzQyMjAyZTYiLCJhbGci...

Python example: Invoke a managed API with OAuth 2.0 authentication

You can invoke a managed API where OAuth 2.0 authentication is enabled in Python 3.

In order to invoke a managed API with an OAuth 2.0 authentication method, you must request an OAuth 2.0 token from the Informatica Intelligent Cloud Services OAuth 2.0 server. The token is valid for 15 minutes. After 15 minutes, you must request a new token.

You can use any OAuth 2.0 library, tool, or programming language to run the OAuth 2.0 authentication sequence. Before you run the OAuth 2.0 authentication, verify that you have the following information:

- URL of the Informatica Intelligent Cloud Services OAuth 2.0 server.
- OAuth 2.0 client ID and secret with permissions to run the managed API.

The following example shows the codes used for invoking a managed API with OAuth 2.0 authentication in Python 3:

```
import sys
import requests
import json
import logging
import time

logging.captureWarnings(True)

test_api_url = "https://apigw-pod1.dm-us.informaticacloud.com/t/apim.usw1.com/
get_employee_details"

##
##    function to obtain a new OAuth 2.0 token from the authentication server
##
def get_new_token():

auth_server_url = "https://dm-us.informaticacloud.com/authz-service/oauth/token"
client_id = 'Jl88QzqE3GYvaibOVb1Fx'
client_secret = '9xy23jdl'

token_req_payload = {'grant_type': 'client_credentials'}

token_response = requests.post(auth_server_url,
data=token_req_payload, verify=False, allow_redirects=False,
auth=(client_id, client_secret))

if token_response.status_code !=200:
            print("Failed to obtain token from the OAuth 2.0 server", file=sys.stderr)
            sys.exit(1)

            print("Successfuly obtained a new token")
            tokens = json.loads(token_response.text)
            return tokens['access_token']

##
##     obtain a token before calling the API for the first time
##     the token is valid for 15 minutes
##
token = get_new_token()

while True:

##
##   call the API with the token
##
api_call_headers = {'Authorization': 'Bearer ' + token}
api_call_response = requests.get(test_api_url, headers=api_call_headers, verify+False)

##
##
if    api_call_response.status_code == 401:
            token = get_new_token()
else:
print(api_call_response.text)

time.sleep(30)
```

Java example: Invoke a managed API with OAuth 2.0 authentication

You can invoke a managed API where OAuth 2.0 authentication is enabled in Java.

In order to invoke a managed API with an OAuth 2.0 authentication method, you must request an OAuth 2.0 token from the Informatica Intelligent Cloud Services OAuth 2.0 server. The token is valid for 15 minutes. After 15 minutes, you must request a new token.

You can use any OAuth 2.0 library, tool, or programming language to run the OAuth 2.0 authentication sequence. Before you run the OAuth 2.0 authentication, verify that you have the following information:

- URL of the Informatica Intelligent Cloud Services OAuth 2.0 server.
- OAuth 2.0 client ID and secret with permissions to run the managed API.

The following example shows the codes used for invoking a managed API with OAuth 2.0 authentication in Java:

```
import com.google.gson.Gson;
import com.squareup.okhttp.";

import java.io.IOException;
import java.util.Map;
import java.util.concurrent.Timeunit;

public class OAuthClientSample
(
public static String TEST_API_URL = "https://apigw-pod1.dm-us.informaticacloud.com/t/
apim.usw1.com/get_employee_details";
public static String OAUTH_SERVER_URL = "https://dm-us.informaticacloud.com/authz-
service/oauth/token";
public static String CLIENT_CREDENTIALS =
"YwliT1ZlMWJGRUpsOOhftenFFM8dZdjpRUjQzQXcwbes-";

OkHttpClient client = new OkHttpClient();

public static void main(String [] args) throws Exception
{
    new OAuthClientSample().runApi();
    }

    //
 //  run an OAuth 2.0 in a loop
    //
 public void runApi() throws Exception
{
    //
    //  obtain an OAuth 2.0 token for running the API
    //
 String token = getNewToken();

    while (true)

    //
    //  run the API using the OAuth 2.0 token
    //
    Request request = new Request.Builder()
    -url(TEST_API_URL)
    -method("GET", null)
    -addHeader("Authorization", Bearer + token)
    -build();
    Response response = client.newCall(request).execute();

    //
    // If the token expired, obtain a new token (token is valid for 15 min)
    //
    if (response.code() --401)
                token = getNewToken ();
    else
        System.out.printIn(response.body().string());

        Thread.sleep(TimeUnit.SECONDS.toMillis(30));
        }
}

    /**
        * @return a new OAuth 2.0 token from the authentication server
        * @throws IOException
        */
        String getNewToken() throws IOException
        {
        String authHeader = "Basic    " + CLIENT_CREDENTIALS;

        Request request = new request.Builder()
                -url(AUTH_SERVER_URL + "?grant_type =client_credentials")
                -method("POST", RequestBody.create(MediaType.parse("text/plain"), ""))
                -addHeader("Authorization", authHeader)
                -build();

        Response response = client.newCall(request).execute();

        if (response.code() != 200)
        {
                    System.err.printIn(response.code());
                    System.exit(1);
                    return null;
                }

        Map <String, Object> jsonResponse = new
Gson().fromJson(response.body().string(), Map.class);
        return (String)jsonResponse.get("access_token");

                }
}
```

Invoke a managed API with JSON Web Token authentication

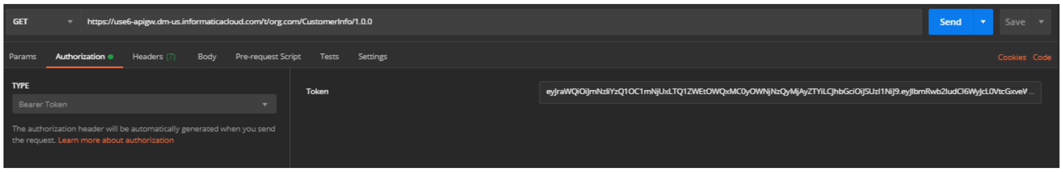
To invoke a managed API that uses JSON Web Token (JWT) authentication, you generate a token and pass the token as a bearer token in the HTTP Authorization header.

API Portal uses the Coordinated Universal Time (UTC) time zone for the JWT token expiration and uses the current time on your computer as the baseline time for the token expiration. The token expires on the expiration date you configure and a minute earlier than the time at which you generated the token. After a token expires you have to generate a new token.

For example, if you generate the token on January 10 at 2:30 p.m. and set the expiration date as January 11, the token expires on January 11 at 2:29 p.m. If you set the expiration date as January 15, the token expires on January 15 at 2:29 p.m.

You can generate a token for a single managed API, for up to 15 APIs simultaneously, and for a managed API group. When you generate a token for a group, you can use the token to invoke the managed APIs in the group that use JWT authentication.

The following image shows an API invoked through Postman with a Bearer Token authorization type and the token:



GET https://use6-apigw.dm-us.informaticacloud.com/t/org.com/CustomerInfo/1.0.0

TYPE: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization

Token: eyJraWQiOiJmNzliYzQ1OC1mNjUxLTQ1ZWEtOWQxMC0yOWNjNzQyMjAyZTYiLCJhbGciOiJSUzI1NiJ9.eyJlbmRwb2ludCI6WyJcL0VtcGxveV ...

Generating and getting JWT tokens for managed APIs

Generate a token for a single managed API or for multiple managed APIs on the **API Registry** page. You can generate a token for up to 15 APIs simultaneously.

1. Use one of the following methods to generate the token:
   - To generate a token for a single managed API, perform the following actions:
     1. Click a managed API that uses JWT authentication.
        The managed API page appears.
     2. Select an expiration date for the token and click **Generate**.
   - To generate a token for multiple managed APIs, perform the following actions:
     1. Select managed APIs that use JWT authentication.
     2. Click the down arrow above the list of APIs and select **Generate Token**.
        The **Generate JWT Access Token** dialog box appears.
     3. Select an expiration date for the token and click **Generate**.

   API Portal generates a JWT token. The token appears in the **JWT Access Token** area. The name of the **Generate** button changes to **Generate New Token**. You can click this button to generate a new token if the current token expires. You cannot revoke a token after you generate it.
2. Click **Copy Token**.

   Use the token in the HTTP Authorization header to invoke the managed API or APIs that you generated the token for.

Generating and getting a JWT token for a managed API group

Generate a token for a managed API group on the **API Groups** page. You can use the token to invoke the managed APIs in the group that use JWT authentication.

1. On the **API Groups** page click the API group.

   The group page appears.
2. Select an expiration date for the token and click **Generate**.

   API Portal generates a JWT token. The token appears in the **JWT Access Token** area. The name of the **Generate** button changes to **Generate New Token**. You can click this button to generate a new token if the current token expires. You cannot revoke a token after you generate it.
3. Click **Copy Token**.

   Use the token in the HTTP Authorization header to invoke managed APIs in the group that use JWT authentication.

INDEX