



Informatica® API Center
October 2025

Manage APIs

© Copyright Informatica LLC 2022, 2025

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2025-10-03

Table of Contents

Preface	5
Chapter 1: API Console.....	6
REST APIs.	7
SOAP APIs.	7
Chapter 2: Published APIs.....	8
Creating a published API.	8
Viewing published history of an API.	9
Viewing a published API.	9
Unpublishing an API.	10
Deleting an unpublished API.	10
Chapter 3: Custom APIs.....	11
Path and query parameters.	11
Publishing a custom API.	12
Rules and guidelines for custom APIs.	13
Chapter 4: Managed APIs.....	14
Managed API lifecycle.	14
Creating a managed API.	16
Editing a managed API.	17
Activating a managed API.	17
Editing policies of active managed APIs and managed API groups.	18
Viewing a managed API.	18
Generating a JWT token.	19
Testing a managed API.	19
Sharing a managed API.	20
Deprecating a managed API.	21
Deactivating a managed API.	21
Deleting a managed API.	22
Downloading a managed API specification.	22
Managing contacts.	23
Chapter 5: API Groups.....	24
Creating an API group.	25
Editing an API group.	25
Adding operations to an API group.	26
Configuring API policies for an API group.	26
Viewing an API group.	27

Deleting an API group.	27
Rules and guidelines for API groups.	28
Chapter 6: Managed API Groups.....	29
Creating a managed API group.	29
Editing a managed API group.	30
Activating a managed API group.	30
Viewing a managed API group.	30
Generate JWT token.	30
Sharing a managed API group.	31
Deprecating a managed API group.	31
Deactivating a managed API group.	32
Deleting a managed API group.	32
Downloading a managed API group specification.	33
Managing contacts for API groups.	33
Rules and guidelines for managed API groups.	34

Preface

Read *Managed APIs* to learn how to manage the lifecycle of a managed API and managed API group in your organization.

CHAPTER 1

API Console

If you are assigned the Deployer or Admin role, you can use the **API Console** page to view the published APIs and managed APIs in your organization.

The **API Console** page displays the following tabs:

Services

The **Services** tab displays all the API implementations that are available in Informatica Intelligent Cloud Services Application Integration. It lists all the processes that you published in Application Integration. You can also view the service name, service provider, provider source name, status, and protocol of the published processes. API Center supports REST and SOAP protocols.

The **Service Name** column indicates the API name of the service provider process. **Provider Source Name** indicates the name of the service provider process. Only the published service provider processes appear on the **Services** tab. **Active** status of the service provider process indicates that the APIs are published and are available for consumption.

The services are listed based on ascending order of the service name. Use the filter to search for the required service name.

You can use the **Services** tab to create a published API using the bottom-up technique.

Published APIs

The **Published APIs** tab displays the published APIs in your organization. It also lists all the REST APIs and SOAP APIs published directly from Application Integration processes that are part of the same organization as the API organization.

You can view the published APIs according to the published API name, version, description, provider, provider source name, protocol, managed API count, and last modified date.

You can use the **Published APIs** tab to create managed APIs. You can create a managed API from any version of a published API. Any changes to one version of the API doesn't impact the API definition of the other versions of the same API.

You can use the **Publish Custom API** button to create and publish custom APIs.

Managed APIs

The **Managed APIs** tab displays the managed APIs in your organization. You can expand a managed API to view the API operations.

You can view the managed APIs according to the managed API name, operation name, published API name, version, status, protocol, context name, API endpoint URL, API policy count, and last modified date. For more information about the different statuses of a managed API, see [“Managed API lifecycle” on page 14](#).

You can click on the number in the **API Policy Count** column to view details of the policies applied to each operation of a managed API.

You can use the **Managed APIs** tab to copy the endpoint URLs of API operations and share them with API consumers. You can also download Swagger 2.0 and OpenAPI 3.0 specifications of an API to share with API consumers.

REST APIs

REST API assets contain API operations that conform to the constraints of REST architectural style and that allow for interaction with RESTful web services. You use REST API assets to create managed APIs that you can share with API consumers.

For more information about REST APIs in API Center, see *REST API Assets*.

SOAP APIs

You can view a list of SOAP API implementations that are available in Application Integration on the **Services** tab of the the **API Console** page in API Center.

You can see the SOAP APIs on the **Services** tab only after you migrate to the new API gateway post July 2024 release.

You can create a published API from an available SOAP API. The steps to create a managed API and activate the managed API remain the same as REST APIs. However, you can't test a managed API created from a SOAP API.

You can download a WSDL file of the SOAP API and share it with API consumers. When you download the WSDL file, you must be logged in as the user who is part of the Allowed Users list in the associated Application Integration process.

The following image shows few sample REST and SOAP APIs on the **Services** tab:

Informatica API Center

API Console				
<div>ServicesPublished APIsManaged APIs</div>				
Service Name	Provider	Provider Source Name	Status	Protocol
aBasSgVWev_POST_JIO_Process	Application Integration	aBasSgVWev_POST_JIO_Process	Active	REST
aBasSgVWev_POST_JIO_Process	Application Integration	aBasSgVWev_POST_JIO_Process	Active	SOAP
aerf	Application Integration	aerf	Active	REST
aerf	Application Integration	aerf	Active	SOAP
AK_Process1	Application Integration	AK_Process1	Active	REST
AK_Process1	Application Integration	AK_Process1	Active	SOAP

CHAPTER 2

Published APIs

You can create a published API using either a top-down approach or a bottom-up approach.

Top-down approach

Using the top-down approach, use the REST API page to create an operation for an API, and then publish the API. Add the operation details, map an Application Integration process to the operation, and configure operational policies and response timeout. After you create the API, save and publish the API.

To create a published API using the top-down approach, you must be assigned the Designer or Admin role.

For more information about creating and publishing an API using the top-down approach, see *Creating an API operation* topic in the *REST API Assets* help.

Bottom-up approach

Using the bottom-up approach, you can create published APIs from existing Application Integration processes.

To create a published API using the bottom-up approach, you must be assigned the Deployer or Admin role.

Use the **Services** tab in the **API Console** page to create a published API.

You can create published APIs and subsequent managed APIs for both active and inactive Application Integration processes. However, you can't invoke a managed API of an inactive Application Integration process.

For more information about creating a published using the bottom-down approach, see [“Creating a published API” on page 8](#).

Creating a published API

1. On the **API Console** page, click the **Services** tab.
The **Services** tab lists all the processes that you published in Application Integration.
2. Hover on the required service name, click the **Actions** menu on the row of the service, and then click **Create Published API**.
The **Publish** dialog box appears.
The **Published API Name** field is pre-populated based on the API name of the Application Integration process.
3. In the **Published API Name** field, edit the name of the published API, if required.
The published API name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, hyphens, dashes, and underscores.

4. Optionally, in the **Notes** field, enter a publishing note.
5. Click **Publish**.

The published API appears on the **Published APIs** tab in the **API Console** page.

By default, the newly created and published API inherits the API name and the authentication type or policy of the service provider process. The operation path is retrieved from the process Swagger file of the service provider. API Center creates a POST operation request by default.

You can then create a managed API from the published API. For more information, see [“Creating a managed API” on page 16](#).

Viewing published history of an API

You can view a comprehensive published history of an API. You can view a detailed log that lists the users who made changes to the API and published the API.

1. On the **API Console** page, click the **Published APIs** tab.
2. Click the **Actions** menu on the row of the published API and select **View Published History**.

The published history details of the API appears.

Viewing a published API

If you are assigned the Admin or Deployer role, you can view and verify the details of a published API before you create a managed API from it.

To view the details of a published API, perform the following steps:

1. On the **API Console** page, click the **Published APIs** tab.
2. Click the **Actions** menu on the row of the published API and select **View Published API**.

The published API details page appears.

You can view the API-level rate limit policy and operation-level policies in read-only mode. You can view the API-level security and privacy policies under the operational-level policies if the operational-level policies inherit the API-level policies.

You can view the details of the following types of published APIs:

- REST APIs that were published using the top-down approach or bottom-up approach
- SOAP APIs that were published using the bottom-up approach. However, you can't view the request and responses details for SOAP APIs.
- Custom REST and SOAP APIs. However, parameter details don't appear for existing custom SOAP APIs.

Unpublishing an API

You can unpublish an API that was published using a top-down approach or a bottom-up approach. You can also unpublish a custom API or any API published through an Application Integration process.

If you no longer need a published API, you can unpublish the API and permanently delete the unpublished API from API Center. The API that you want to unpublish must not contain any managed API in created, active, shared, deprecated, or inactive state.

If you unpublish and delete the latest version of an API, the previous version of the API automatically becomes the latest version of the API.

Unpublishing an API doesn't impact any API group. Any operation of the unpublished API that is referenced in an API group doesn't get unpublished when the API is unpublished.

1. On the **API Console** page, click the **Published APIs** tab.
The **Published APIs** tab shows all the published APIs according to the **Last Modified Date** of the published API.
2. Click the **Actions** menu on the row of the selected published API and select **Unpublish**.
The **Confirm Unpublish** dialog box appears.
3. Click **Unpublish**.
A confirmation message about successful unpublish appears.
4. Go to **Explore > All Projects**, and select your project.
All APIs and assets that you created using the top-down approach appears. The **Published** column displays the status of an unpublished API as **No**.

Deleting an unpublished API

If you are assigned the Deployer or Admin role, you can delete an unpublished API.

You can delete unpublished APIs under the following conditions:

- The API is created using the top-down approach.
 - No other published API in the organization references the same designed API of the published API to be deleted.
1. On the **Explore** page, click **All Projects** and select your project.
 2. Click the **Actions** menu on the row of the unpublished API and select **Delete**.
A warning message appears.
 3. Click **Delete** to delete the unpublished API.
The published API gets deleted from API Center.

CHAPTER 3

Custom APIs

Custom APIs are APIs that aren't based on Informatica Cloud Application Integration processes. Custom APIs help you manage APIs that are external to the Informatica environment.

You can create, manage, and govern custom APIs that you add to your organization's list of APIs. You can't create custom APIs based on Informatica internal and Informatica public URLs.

Path and query parameters

When you create and activate a custom managed API, you can choose to dynamically change the path parameter and query parameter values.

Path parameters and query parameters are request parameters attached to a URL that point to a specific REST and SOAP API resource.

In a custom managed API, you can use the following format for a query parameter:

Query key and value (`?param=value`)

Path parameters are part of the endpoint and are required. For example, in `/users/{id}`, `{id}` is the path parameter of the endpoint `/users`, which points to a specific user's record. An endpoint can have multiple path parameters. For example, `/organizations/{orgId}/members/{memberId}`. The endpoint is pointing to a specific member's record within a specific organization, where both `{orgId}` and `{memberId}` are mandatory variables.

Use path parameters when you need to identify a specific resource within the managed API. The path parameter is separated from the URL by a `/` and is contained within curly braces `{ }`.

Query parameters are separated from the URL by a question mark (`?`). A query parameter defines sort, pagination, search, or filter operations. A query is passed as a variable in the query parameter. For example, `/users?sort=name&limit=10`. This URL consists of two query parameters, `sort` and `limit`. Query parameters can modify the behavior of an API request by adding more information.

Use query parameters when you need to provide additional information or filter criteria without changing the core resource being accessed.

For more information about setting up path and query parameters in a custom managed API, see [“Publishing a custom API” on page 12](#)

Publishing a custom API

You can publish a custom API, and then create a managed API from it.

1. On the **API Console** page, click the **Published APIs** tab.

The **Published APIs** tab shows all the published APIs according to the last modified date of the published API.

2. Click **Publish Custom API**.

The **Publish Custom API** dialog box appears.

3. On the **API Details** tab, enter the required values.

Field	Description
Name	The name that you assign to the custom API is part of the API URL. The published API name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, hyphens, dashes, and underscores.
Version	Version of the published API. You can't change the version.
API Type	Select a protocol from the list. You can select REST or SOAP.
API URL	URL for the API. The URL format must conform to the W3C standard. You can add path parameters and query parameters to the URL. Sample URL: <code>https://reqres.in/api/users/{User_ID}?query=20</code> The sample URL contains the following path parameters and query parameters: <ul style="list-style-type: none">- {User_ID}: Path parameter for a specific user's ID, which will be replaced by an actual user ID when making the request. For example, <code>https://reqres.in/api/users/2</code> If you have multiple path parameters that you would define later, you can represent them with an asterisk (*) in the URL structure. For example: <code>https://reqres.in/api/users/{User_ID}/*?query=20</code> In this URL, {User_ID} indicates the first path parameter, such as a user ID and * indicates any additional path parameters you might add later.- query=20. A query parameter that consists of a key (query) and a value (20). Query parameters come after the question mark (?) and are used to provide additional data for the request. Note: API Center doesn't validate query parameters. For all custom APIs that were created prior to October 2024 release, you can optionally create another custom managed API to update the path parameter and query parameter values dynamically.
Method	Select an HTTP method for the API from the available list. Default is GET.
Description	(Optional) Enter a publishing note.

4. Click **Next**.

The **Test** tab appears.

Optionally, you can change the values of the path parameters and query parameters.

5. Click **Try it Out!** to verify if the URL is valid and reachable.
6. You can choose to skip the verification for the custom API, if required.

If you do not want to test the API URL provided you can enable this option. For example, API Center doesn't verify URLs for SOAP APIs that are imported from Application Integration.

7. Click **Next**.

The **Summary** tab displays all the configured details.

8. Click **Publish**.

The published custom API appears on the **Published APIs** tab in the **API Console** page.

You can then create a managed API from the published custom API. For more information, see ["Creating a managed API" on page 16](#). You can't test a custom managed API or its operations.

Rules and guidelines for custom APIs

Consider the following rules and guidelines when you work with custom APIs:

- When you create a custom API, the API URL that you specify must not be a URL that is restricted by Informatica and must not contain any potential vulnerability or malicious script. The URL must use a certificate that is signed by a well-known Certificate Authority. You can't create a custom API for an HTTPS URL that uses a self-signed certificate or a certificate that isn't signed by a Certificate Authority.
- When you configure a response caching policy for a custom managed API, do not select the `Accept-Encoding` request HTTP header. If you select the `Accept-Encoding` header, the response caching policy might not be honored.
- You can't configure a Personally Identifiable Information (PII) policy for a custom managed API that contains the `Accept-Encoding` request HTTP header as `br(Brotli)` or response as `br(Brotli) compressed`.

CHAPTER 4

Managed APIs

Managed APIs are APIs with unified authentication methods that you can monitor access to with usage policies. Using managed APIs ensures that API consumers can safely and securely use and re-use the organization APIs with set policies, access control, and well-defined endpoints.

Use the **API Console** page to edit, view, activate, share, deprecate, deactivate, and delete the managed APIs in the organization. To create a managed API, you select a published API and create a managed API from it.

When you create a managed API for enterprise services and processes, you can copy the API URL and send it to API consumers. You can also download the API specification for a managed API and send it to API consumers.

You can choose the domain name when you log in to API Center the first time, and then provision the domain.

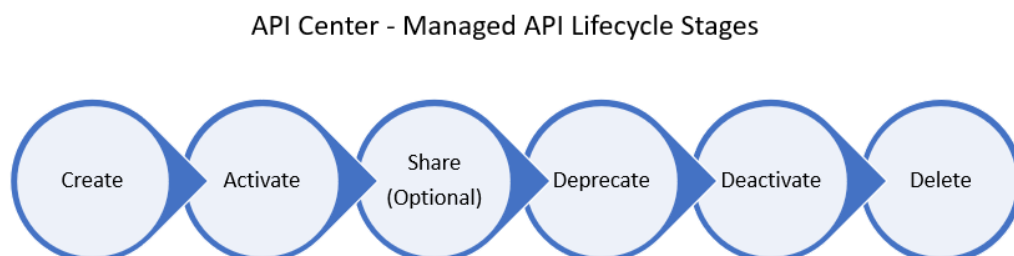
Use the **API Monitor** page to monitor the APIs in the organization and to download API activity logs.

Managed API lifecycle

API Center supports the complete lifecycle management of managed APIs. You can create a managed API from a published API, transition from one state to the next state, revise the managed API, and control the deprecation and removal of APIs. This API center managed API lifecycle ensures smooth transition of managed APIs from one state to another and backward compatibility as APIs evolve over time.

The following image shows the basic lifecycle of a managed API:

Figure 1.



The following table shows the different states of a managed API:

Managed API status	Description
Created	<p>Default status of the managed API. Every managed API when created is always in the Created state. You can edit a managed API that is in the Created state. You can't share a managed API with API consumers if it is in the Created state.</p> <p>You can perform the following actions on a managed API that is in the Created state:</p> <ul style="list-style-type: none"> - Edit the managed API - View the managed API - Activate the managed API - Delete the managed API <p>For more information, see "Creating a managed API" on page 16.</p>
Active	<p>Indicates that the managed API is active and deployed on the API gateway. You can share the API endpoint URLs for the API operations with API consumers.</p> <p>You can edit the policies of a managed API that is in active state.</p> <p>You can perform the following actions on a managed API that is in the Active state:</p> <ul style="list-style-type: none"> - View the managed API - Edit policies associated with the managed API - Test the managed API - Share the managed API - Deprecate the managed API - Download the Swagger 2.0 specification - Download the Open API 3.0 specification <p>For more information, see "Activating a managed API" on page 17.</p>
Shared	<p>(Optional) Indicates that the managed API and its operations are shared.</p> <p>You can perform the following actions on a managed API that is in the Shared state:</p> <ul style="list-style-type: none"> - View the managed API - Edit policies associated with the managed API - Test the managed API - Deprecate the managed API - Download the Swagger 2.0 specification - Download the Open API 3.0 specification <p>For more information, see "Sharing a managed API" on page 20.</p>
Deprecated	<p>Indicates that the managed API is deprecated. You can deprecate a managed API to inform the API consumers that the deprecated managed API might not be available for consumption in the future.</p> <p>You can perform the following actions on a managed API that is in the Deprecated state:</p> <ul style="list-style-type: none"> - View the managed API - Edit policies associated with the managed API - Deactivate the managed API <p>For more information, see "Deprecating a managed API" on page 21.</p>
Deactivated	<p>Indicates that the managed API is deactivated. You can deactivate a deprecated managed API.</p> <p>You can perform the following actions on a managed API that is in the Deactivated state:</p> <ul style="list-style-type: none"> - Edit the managed API - View the managed API - Activate the managed API - Delete the managed API <p>For more information, see "Deactivating a managed API" on page 21.</p>

Creating a managed API

Create a managed API from a published API. To create a managed API, you must be assigned the Deployer or Admin role. When you create a managed API, API Center generates API endpoint URLs for the API operations, which you can share with API consumers to access the operations with.

1. On the **API Console** page, select the **Published APIs** tab.

The **Published APIs** tab shows all the published APIs according to the **Last Modified Date** of the published API.

2. Click the **Actions** menu on the row of the selected published API and select **Create a Managed API**.

3. Enter a name for the managed API.

The name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, hyphens, dashes, and underscores.

4. Optionally, enter a description.

5. In the **URL Context** field, enter the context that API Center adds to the API URL.

The context can contain up to 80 characters, including ASCII letters, Japanese characters, digits, hyphens, and underscores.

The context can be shared between different versions of a managed API that belongs to the same published API.

6. Optionally, enable the JSON web token (JWT) authentication policy at the managed API level.

You can enable JSON web token authentication at the API level. You can use this JSON web token to authenticate the API and all its operations.

- a. Optionally, add required notes, and then click **Save**.

Notes can contain a maximum of 500 characters.

- b. If both API-level and operation-level authentications are selected, the **Warning** dialog box appears. Click **OK** to confirm overriding of all the existing operation-level authentication.

If no authentications were applied to the API operations previously, this dialog box doesn't appear.

For more information about using an API-level authentication policy, see *API Policies*.

7. Optionally, create an IP filtering policy at the managed API level.

For more information about IP filtering policies, see *API Policies*.

8. Optionally, create and associate a user-level rate limit policy for a specific user of the managed API.

For more information about user-level rate limit policies, see *API Policies*.

9. Optionally, create and associate a CORS policy at the managed API.

For more information about CORS policies, see *API Policies*.

10. Optionally, add or update operation level authentication, rate limit, response caching, or privacy policies for both managed APIs or custom managed APIs.

When you deploy an API, you can override the policies that are assigned to it. The following table shows the optional metadata that you can modify for an operation:

Policies	Options
Authentication	Choose one or more of the following applicable policy type: <ul style="list-style-type: none">- Anonymous- Basic- OAuth 2.0- JWT - JSON Web Token- Session ID To invoke a managed API using session ID, pass the session ID header value as <code>IDS-SESSION-ID</code> and then run the API.
Rate Limit	Choose from the available rate limits. Default is system-org-level-rate-limit .
Response Caching	Define the caching timeout in seconds.
Privacy Settings	Define the privacy policy for the operation.

For more information about authentication policies, see *API Policies*.

11. Click **Save**.

If any validation errors occur, the **Validation** panel appears. Fix all errors listed on the **Validation** panel and click **Save** again.

API Center saves the managed API.

Editing a managed API

If you are assigned the Deployer or Admin role, you can edit managed APIs. For example, you can update the policies that the managed API uses. You can't edit the name of the managed API.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API to edit and select **Edit Managed API**.
3. Edit the managed API, and then click **Save**.

API Center applies the updated policies to the API and API operations.

Activating a managed API

If you are assigned the Deployer or Admin role, you can activate a managed API with a created status.

API consumers can invoke only the activated APIs and API operations. You can edit the policies of a managed API that is in active state. For more information, see [“Editing policies of active managed APIs and managed API groups” on page 18](#).

1. On the **API Console** page, click the **Managed APIs** tab.

2. Click the **Actions** menu on the row of a managed API with a created status and select **Activate Managed API**.

API Center activates the managed API.

When you activate a managed API, you can choose to dynamically change the path parameter and query parameter values.

In a managed API, you can use the following formats for query parameters:

- Query key (?param)
 - Query key with equals (?param=)
 - Query key and value (?param=value)
3. Click the **Managed APIs** tab, expand the managed API, and then click **Copy** under the **API Endpoint URL** column of an operation.
 4. Share the API endpoint URLs of the operations with API consumers to access the operations with.

Editing policies of active managed APIs and managed API groups

If you are assigned the Deployer or Admin role, you can update the policies of your managed APIs and managed API groups that are in the active, shared, and deprecated state.

You can edit both API-level and operation-level policies while the managed API or managed API group is in the active, shared, or deprecated state. When you update the policies of a managed API or managed API group, the update will immediately impact the deployed managed API at run time.

Example

If your API's rate limit is 1,000 requests per 60 minutes, you can increase it to 2,000 instantly without downtime or token invalidation.

You want to change the basic authentication to anonymous authentication of an active managed API at run time. Changing the authentication method updates the managed API access immediately.

Previously, deactivating a managed API broke the OAuth associations, requiring client reassignment after reactivation. Now, OAuth associations remain intact during policy edits, so existing tokens continue working.

Viewing a managed API

If you are assigned the Deployer or Admin role, you can view the configurations of a managed API. You can view the managed API level policies and operation level policies in read-only mode.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **View Managed API**.

The managed API details appear.

Generating a JWT token

If you are assigned the Deployer or Admin role, you can generate an API-level JSON web token (JWT). You can use the JSON web token to invoke the API and all its operations.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Generate JWT token**.
The **Generate JWT token** option is enabled for the managed APIs with API-level JSON web tokens (JWT).
The **Generate API Level JSON Web Token** dialog box appears.
3. Optionally, edit the expiration date and time.
You can choose a maximum of 180 days for the expiration date.
4. Click **Generate New Token**.
5. Copy the generated token and use the token to invoke the API and all its operations.
For more information about using an API-level authentication policy, see *API Policies*.

Testing a managed API

If you are assigned the Deployer or Admin role, you can test a managed API and its operations.

You can either click **Test Managed API** at the managed API level or test one particular operation directly by clicking **Test Operation**.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Test Managed API**.
To test an operation of an API, click the API to expand the list of operations for the API, and then click **Test Operation**. You can test an operation of a managed API only when the API is in the **Active** state.
3. Click **API Policies > Security**.
4. In the Authentication panel, click **Generate New Token**.
The **Generate New Token** option is enabled only if the API-level JSON web token is selected.
Optionally, edit the expiration date and time. You can choose a maximum of 180 days for the expiration date.
For more information about using an API-level authentication policy, see *API Policies*.
5. Click **Operations** and select an operation.
6. On the **Test** tab, click **Authorize**.
If the managed API requires authentication, the **Available authorizations** dialog box appears.
7. Based on the authentication type, enter the respective details to authenticate the managed API:
 - For basic authentication, enter the IDMC user name and password.
 - For OAuth 2.0 authentication, enter the `client_id` and `client_secret`.
 - For JWT authentication, enter the JSON web token value.
 - For session ID authentication, enter the session ID.

Note: All Data Quality operations must contain a session ID based authentication policy.

For more information about the authentication methods, see [Security policies](#).

8. Click **Authorize**.
9. To view the API request body and response code, click anywhere on the Swagger endpoint.
10. To test the API, in the request body panel, perform the following steps:
 - a. Click **Try it out**.
 - b. Enter the parameter values if required.

Edit the request body and replace the example values of request fields with any value of your choice.
 - c. To test the managed API, click **Execute**.

The server response panel displays the response body and response headers.
To clear the server response, click **Clear**.
To exit the edit mode of the request body, click **Cancel**. The changes done to the request body are retained.
To reset the changes made to the request body, click **Reset**.
To change the request body again, click **Try it Out**.

Note: When you run a managed API endpoint that has response caching configured, the response caching doesn't work if the upstream API response is in compressed format. The `Accept-Encoding` request header typically contains a comma-separated list of encoding formats, such as `gzip`, `br`, and `deflate`. The `Accept-Encoding` request header is added by default in Postman or web browser. To resolve the issue of honoring the response caching, use Postman to disable the `Accept-Encoding` header in your request, and then invoke the API. However, if you invoke the API from a web browser, `Accept-Encoding` is added automatically. As a result, the response caching policy might not be honored.

Sharing a managed API

If you are assigned the Deployer or Admin role, you can share a managed API and its operations. You can share a managed API if the API is in the active state.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Share Managed API**.

The **Update <managed_API_name> Lifecycle Status** dialog box appears.
3. Optionally, on the **Shared** tab, enter one or more valid email addresses.

No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.

An email address with Japanese characters in the domain name is considered as an invalid email address.
4. Optionally, add required notes.
5. Click **Confirm**.

The managed API status changes to **Shared**. You can view, test, or deprecate a shared managed API.

Deprecating a managed API

If you are assigned the Deployer or Admin role, you can deprecate a managed API and its operations. You can deprecate a managed API if the API is in the active state or shared state.

If you plan to deactivate a managed API, deprecate the managed API first to inform the API consumers that the deprecated managed API might not be available for consumption in the future. The existing API consumers can continue using the deprecated managed API and its operations until the API is deactivated.

After you deprecate a managed API, the API endpoint URL doesn't appear on the **Managed APIs** tab as the API. You can't download the API specification for a deprecated API.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Deprecate Managed API**.
The **Update <managed_API_name> Lifecycle Status** dialog box appears.
3. Optionally, on the **Deprecated** tab, enter one or more valid email addresses.
No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.
4. Optionally, add required notes.
5. Click **Confirm**.
The managed API status changes to **Deprecated**.
You can view or deactivate a deprecated managed API.

Deactivating a managed API

If you are assigned the Deployer or Admin role, you can deactivate a managed API and its operations. You can view, edit, activate, or delete a deactivated managed API.

You can deactivate a managed API when the API is in the deprecated state. You can activate and use an inactive managed API.

After you deactivate a managed API, the API is no longer available in the API gateway and all calls to the API will fail. Ensure that you notify the API consumers about the impending deactivation of the managed API before deactivating the API to prevent disruptions and enable them to make necessary adjustments.

If the deactivation fails, the status of the managed API doesn't change. You can't view the operation level status of a managed API.

You can use the **Show Inactive APIs** toggle button to see the list of inactive or deactivated APIs.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Deactivate Managed API**.
The **Update <managed_API_name> Lifecycle Status** dialog box appears.
3. On the **Inactive** tab, enter one or more valid email addresses.
No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.
4. Optionally, add required notes.

5. Click **Confirm**.

The managed API status changes to **Inactive**.

The managed API gets deactivated and the managed API count reduces by 1.

Similarly, on the **Published APIs** tab, the managed API count of the respective published API reduces by 1.

Deleting a managed API

If you are assigned the Deployer or Admin role, you can delete a managed API and its operations.

You can delete a managed API when the API is in the deactivated state or created state. If you no longer need a managed API, you can permanently delete the managed API from API Center. All information related to the deleted managed API is removed from API Center. However, you can reuse the context of the deleted managed API for other APIs.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Delete Managed API**.
The **Delete <managed_API_name>** dialog box appears.
3. Click **Yes** to delete the managed API.
The managed API gets deleted from API Center.

Downloading a managed API specification

If you are assigned the Deployer or Admin role, you can download the API specification of a managed API. You can download a Swagger 2.0 specification or an OpenAPI 3.0 specification and share it with API consumers.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API to download the required API specification.
3. Perform one of the following steps:
 - To download a Swagger 2.0 specification, select **Download Swagger 2.0**.
 - To download an OpenAPI 3.0 specification, select **Download OpenAPI 3.0**.
4. Share the API specification with API consumers.

You can't download the API specification for a deprecated API.

Note: You can't download API specifications for managed custom APIs.

Managing contacts

If you are assigned the Deployer or Admin role, you can enter the email addresses of the API consumers whom you would want to notify about any status change of the managed API.

No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.

1. On the **API Console** page, click the **Managed APIs** tab.
2. Click the **Actions** menu on the row of the managed API and select **Manage Contacts**.
The **Manage Contacts** dialog box appears.
The **Manage Contacts** option is available only for **Shared** and **Deprecated** managed APIs.
3. Enter one or more valid email addresses.
4. Optionally, add required notes.
5. Click **OK**.

CHAPTER 5

API Groups

You can categorize published REST, SOAP, and custom APIs and their operations into logical API groups to easily manage the APIs in your organization. To create an API group, you must be assigned the Deployer or Admin role.

Using API groups, you can package a set of related APIs and assets for easy discovery. You can customize the API package with different Service Level Agreements (SLAs) and rate limit policies to meet diverse requirement scenarios. You can enhance API management by allowing the API group administrators to create, update, and manage API groups and their SLAs within an organization.

Managing APIs in a group helps avoid API duplication. If you can't find existing APIs to reuse in part or as a whole, APIs are likely to be regularly duplicated, increasing the development time and costs.

With API groups, you can engage more roles in your API ecosystem and share your APIs with extended users to reap the benefits of an API-first approach. This approach enables product owners and business technologists to easily find, use, and work with your APIs.

API group examples

Some example scenarios of API groups include:

Employees APIs

If your application holds "Employees" records, then you can create one group named "Employee APIs". Include all required actions to get information, add, modify, and delete employee records. Similarly, you can create a "Salaries APIs" group to manage related salaries information.

Online food delivery APIs

An online food delivery service can have various APIs, such as,

- User API
- Restaurant location API
- Payments API
- Transaction API
- Rewards API

These APIs can be grouped into packages like user-centric group which can include User and Rewards API, and a Financial group which can include Payments and Transactions API.

Weather API

A weather company that forecasts atmospheric conditions can have different APIs, such as,

- Locations API
- Forecast API
- Current Conditions API

- Weather Alarms API
- Alerts API
- Imagery API
- Tropical API

Using API groups, the weather company can package different sets of APIs to target diverse consumer requirements with specific SLAs that meet those needs. API consumers can access related APIs to simplify asset discovery. Administrators can modify SLAs with ease, ensuring governance.

Creating an API group

Create an API group on the **API Groups** page. After you publish a REST API, SOAP API, or custom API, you can add the API operations to an API group. An operation of a published API can belong to several API groups.

1. On the **API Groups** page, click **Create API Group**.
The **Create API Group** dialog box appears.
2. In the **Name** field enter a group name.
The API group name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, n-dash, and the following special characters: \$ () [] . ? ` - _
3. Optionally, enter a category for the API group.
You can create a new category or select an existing category.
4. Optionally, enter a description of the group.
5. Click **Create**.
API Center saves and publishes the API group. The published API group page opens where you can add operations to the API group and configure API policies for the group.
6. On the **Groups Operations** tab, click **Add Operations** to add operations from different published APIs.
For more information, see [“Adding operations to an API group” on page 26](#).
7. On the **Policies** tab, you can associate an user-level rate limit policy, add IP filtering rules, and add an API-level JSON web token (JWT) authentication or an OAuth 2.0 authentication to the API group.
You can configure a rate limit policy for all operations in an API group. You can configure a user-level rate limit policy for all users listed under **User-Level Rate Limit** area.
For more information, see *API Policies*.
8. Click **Save**.
You can then create a managed API group from the API group. For more information, see [“Creating a managed API group” on page 29](#).

Editing an API group

You can edit an API group from the API definitions page.

1. On the **API Groups** page, select the **API Groups** tab.

2. Click the **Actions** menu on the row of the API group and select **Edit API Group**.
Alternatively, click on the API group to open the group in view mode. Click the **Edit** button to edit the API group.
3. Edit the API group and then click **Save**.
You can change the description, category, or operations as needed. You can't change the name of the API group as the group gets published as soon as it is created.

Adding operations to an API group

Create an API group on the **API Groups** page. You can add operations from different published APIs to an API Group. An operation of a published API can belong to several API groups.

1. On the **API Groups** page, select the **API Groups** tab.
2. Click the **Actions** menu on the row of the API group and select **Edit API Group**.
3. On the **Groups Operations** tab, click **Add Operations** to add operations from different published APIs.
The **Add Operations** dialog box appears. You can view all the published APIs and their operations in this dialog box. Use the **Published API Name** filter to quickly locate your published APIs.
4. Select the published APIs and operations that you want to add to the API group, and click **Add**.
The selected published API operations appear in the API group page.
You can group different operations from different published APIs. You can view the operation-level policies applied to each operation.
5. Click **Add**.
The selected operations appear in the **Group Operations** tab of the **API Group** page.
You can add only one operation with the same operation type and operation endpoint to one API group.
Note: You can't add Data Quality operations to an API group.
6. Click **Save**.

Configuring API policies for an API group

You can configure and assign rate limit policy, JSON web token authentication, and IP filtering policy to API groups.

Rate limit configuration for API groups

You can associate a rate limit policy with any operation in an API group.

1. On the **API Groups** page, select the **API Groups** tab.
2. Click the **Actions** menu on the row of the API group and select **Edit API Group**.
3. On the **Policies** tab, click **Rate Limit**.
4. Select a rate limit for the managed API group. The organization level rate limit policy is applied to the managed API group by default.
5. Optionally, add user-level rate limit.

6. Click **Save**.

IP filtering rules configuration for API groups

Create an IP filtering policy at the managed API group level.

1. On the **API Groups** page, select the **API Groups** tab.
2. Click the **Actions** menu on the row of the API group and select **Edit API Group**.
3. On the **Policies** tab, click **IP Filtering Rules > Add IP Filtering Rule..**
4. Select to allow or deny a range of addresses, and then fill in the IP range.
5. Optionally, enter a description of the rule. Add additional rules as required to define the policy.
6. Click **Save**.

JWT access token configuration for API groups

Enabling the API-level JSON web token authentication while creating an API group overrides all the operation-level authentications. You can't change the authentication type while creating a managed API group. After you activate the managed API with API-level JSON web token authentication, you can't remove the JSON web token authentication from that particular API group.

OAuth 2.0 security authentication for API groups

When you create or edit an API group or managed API group, you can enable OAuth 2.0 authentication at the API group level. You can then invoke the APIs in the managed API group with OAuth 2.0 or third-party authorization access tokens. When you enable API-level authentication for an API group, all the operation-level authentications are overridden. However, this behavior doesn't apply to custom API operations.

For more information about API policies, see *API Policies*.

Viewing an API group

If you are assigned the Deployer or Admin role, you can view the configurations of an API group. You can view the API-level policies and operation level policies in read-only mode.

1. On the **API Groups** page, select the **API Groups** tab.
2. Click the **Actions** menu on the row of the API group and select **View API Group**.

Deleting an API group

If you are assigned the Deployer or Admin role, you can delete an API group and its operations.

1. On the **API Groups** page, select the **API Groups** tab.
2. Click the **Actions** menu on the row of the API group and select **Delete API Group**.
A delete confirmation dialog box appears.
3. Click **Delete**.

The API group gets deleted from API Center. You can't recover deleted API groups.

Rules and guidelines for API groups

Consider the following rules and guidelines when you work with API groups:

- You can group different operations from different published APIs. You can view the operation-level policies applied to each operation.
- You can add only one operation with the same operation type and operation endpoint to one API group.
- After you delete an API operation from an API group, save the API group to successfully remove the operation from the API group. Otherwise, the API operation is not removed from the API group.
- After you create and save an API group, if any associated authentication policy changes any operation in that API group, you must add the operations again for the policies to take effect. Authentication policies doesn't refresh automatically for any operation in an API group.
- If you associate a rate limit policy at the API level, operations level, and user level, the rate limit policy with the minimum value takes precedence and the other values are ignored.
- If you associate an API-level rate limit policy with an API, but do not assign policies to individual API operations, the API-level rate limit policy is automatically applied to the operations when you add the published API operation to an API group. The active managed API groups also enforces the applied rate limit.

CHAPTER 6

Managed API Groups

You can create subsequent managed API groups from an API group.

You can add a URL context to a managed API group, create an IP filtering policy for a managed API group, and generate a JSON web token for a group of managed APIs. You can assign a rate limit policy to a managed API group.

Creating a managed API group

Create a managed API group from an API group. To create a managed API, you must be assigned the Deployer or Admin role.

When you create a managed API group, API Center generates API endpoint URLs after activating the managed API group. You can share API endpoint URLs with API consumers to access the APIs and operations in the group.

1. On the **API Groups** page, select the **API Groups** tab.

The **API Groups** tab shows all the published API groups according to the last modified date of the published API group. You can filter the API groups based on the group name.

2. Click the **Actions** menu on the row of the selected published API group and select **Create Managed API Group**.

Alternatively, you can create a managed API group from any specific API group page.

The **Create Managed API Group** dialog box appears.

3. Enter a name for the managed API group.

The managed API group name can contain up to 80 characters, including ASCII letters, digits, Japanese characters, n-dash, and the following special characters: \$ () [] . ? ` - _

4. Optionally, enter a category and a description.

5. In the **Context** field, enter the context that API Center adds to the API group URL.

The context can contain up to 80 characters, including ASCII letters, Japanese characters, digits, hyphens, and underscores.

The context must be unique across an organization. You can't use the same context for managed APIs and managed API groups.

6. Click **Create**.

The managed API group is created and the managed API group will be in created state.

Editing a managed API group

If you are assigned the Deployer or Admin role, you can edit managed API groups. For example, you can update the policies that the managed API group uses. You can't edit the name of the managed API group.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group to edit and select **Edit Managed API Group**.
3. Edit the managed API, and then click **Save**.

API Center applies the updated policies to the group operations.

Activating a managed API group

If you are assigned the Deployer or Admin role, you can activate a managed API group with a created status.

API consumers can invoke only the activated managed API groups and operations. You can edit the policies of a managed API group that is in active state. For more information, see [“Editing policies of active managed APIs and managed API groups” on page 18](#).

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of a managed API group with a created status and select **Activate Managed API Group**.

API Center activates the managed API group.

3. Click the managed API group name that you activated to open the managed API group.
4. On the **Groups Operations** tab, click **Copy** under the **API Endpoint URL** column of an operation. Share the API endpoint URLs of the group operations with API consumers for accessing the API group operations.

Viewing a managed API group

If you are assigned the Deployer or Admin role, you can view the configurations of a managed API group. You can view the managed API-level policies and operation level policies in read-only mode.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **View Managed API Group**.

Generate JWT token

If you are assigned the Deployer or Admin role, you can generate an API-level JSON web token (JWT). You can use the JSON web token to invoke the API groups and all its operations.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Generate JWT token**.

The **Generate JWT token** option is enabled for the managed APIs groups with API-level JSON web tokens (JWT).

The **Generate API Level JSON Web Token** dialog box appears.

3. Optionally, edit the expiration date and time.

You can choose a maximum of 180 days for the expiration date.

4. Click **Generate New Token**.
5. Copy the generated token and use the token to invoke the API groups and all its operations.

For more information about using an API-level authentication policy, see *API Policies*.

Sharing a managed API group

If you are assigned the Deployer or Admin role, you can share a managed API group and its operations. You can share a managed API group if the API group is in the active state.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Share Managed API Group**.

The **Update <managed_API_groupname> Lifecycle Status** dialog box appears.

3. Optionally, on the **Shared** tab, enter one or more valid email addresses.

No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.

4. Optionally, add required notes.
5. Click **Confirm**.

The managed API group status changes to **Shared**. You can view, manage contact, or deprecate a shared managed API group.

Deprecating a managed API group

If you are assigned the Deployer or Admin role, you can deprecate a managed API group and its operations. You can deprecate a managed API group if the API is in the active state or shared state.

If you plan to deactivate a managed API group, deprecate the managed API group first to inform the API consumers that the deprecated managed API group might not be available for consumption in the future. The existing API consumers can continue using the deprecated managed API group and its operations until the API group is deactivated.

After you deprecate a managed API group, the API endpoint URL doesn't appear on the **Group Operations** tab of the managed APIs group page. You can't download the API group specification for a deprecated managed API group.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Deprecate Managed API Group**.

The **Update <managed_API_Group_name> Lifecycle Status** dialog box appears.

3. Optionally, on the **Deprecated** tab, enter one or more valid email addresses.
No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.
4. Optionally, add required notes.
5. Click **Confirm**.
The managed API group status changes to **Deprecated**.
You can view or deactivate a deprecated managed API group.

Deactivating a managed API group

If you are assigned the Deployer or Admin role, you can deactivate a managed API group and its operations. You can view, edit, activate, or delete a deactivated managed API.

You can deactivate a managed API group when the API is in the deprecated state. You can activate and use an inactive managed API group.

After you deactivate a managed API group, the API group is no longer available in the API gateway and all calls to the API group will fail. Ensure that you notify the API consumers about the impending deactivation of the managed API group before deactivating the API to prevent disruptions and enable them to make necessary adjustments.

If the deactivation fails, the status of the managed API group doesn't change. You can't view the operation level status of a managed API group.

You can use the **Show Inactive API Groups** toggle button to see the list of inactive or deactivated APIs.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Deactivate Managed API Group**.
The **Update <managed_API_Group_name> Lifecycle Status** dialog box appears.
3. On the **Inactive** tab, enter one or more valid email addresses.
No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.
4. Optionally, add required notes.
5. Click **Confirm**.
The managed API group status changes to **Inactive**.
The managed API group gets deactivated.

Deleting a managed API group

If you are assigned the Deployer or Admin role, you can delete a managed API group and its operations.

You can delete a managed API group when the API is in the deactivated state or created state. If you no longer need a managed API group, you can permanently delete the managed API group from API Center. All

information related to the deleted managed API group is removed from API Center. However, you can reuse the context of the deleted managed API group for other APIs.

1. On the **API Groups** page, click the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Delete Managed API Group**. A delete confirmation dialog box appears.
3. Click **Delete**.
The managed API group gets deleted from API Center.

Downloading a managed API group specification

If you are assigned the Deployer or Admin role, you can download the API specification of a managed API group. You can download a Swagger 2.0 specification or an OpenAPI 3.0 specification and share it with API consumers.

1. On the **API Groups** page, select the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the active state managed API group to download the required API specification.
3. Perform one of the following steps:
 - To download a Swagger 2.0 specification, select **Download Swagger 2.0**.
 - To download an OpenAPI 3.0 specification, select **Download OpenAPI 3.0**.
4. Share the API specification with API consumers.

You can't download the API specification for a deprecated API.

If a managed API group contains APIs other than REST APIs, you can't download the API specification of that managed API group.

Managing contacts for API groups

If you are assigned the Deployer or Admin role, you can enter the email addresses of the API consumers whom you would want to notify about any status change of the managed API group.

No email communications are sent to these email addresses. You can use the **Email Address** field as a placeholder to add the email addresses. You can refer to the placeholder email addresses when you want to send a notification to the API consumers externally.

1. On the **API Groups** page, click the **Managed API Groups** tab.
2. Click the **Actions** menu on the row of the managed API group and select **Manage Contacts**. The **Manage Contacts** dialog box appears.
The **Manage Contacts** option is available only for **Shared** and **Deprecated** managed API groups.
3. Enter one or more valid email addresses.
4. Optionally, add required notes.
5. Click **OK**.

Rules and guidelines for managed API groups

Consider the following rules and guidelines when you work with managed API groups:

- You can edit the policies of a managed API group that is in active state. For more information, see [“Editing policies of active managed APIs and managed API groups” on page 18](#).
- Enabling the API-level JSON web token authentication while creating an API group overrides all the operation-level authentications. You can't change the authentication type while creating a managed API group.
- If the managed API group is in the created or inactive state, you can remove the JSON web token authentication from that particular API group. Select an operation-level authentication policy to activate the managed API group.

If the managed API group is not in the created or inactive state, you can't remove the JSON web token authentication from that particular API group. You must create another managed API group without JSON web token authentication.

- If an associated rate limit policy is disabled for an operation in an API group, the disabled policy is removed automatically from the operation while activating its managed API group.
- You can't activate a managed API group if it contains a disabled API-level rate limit policy. Edit the managed API group to associate a valid API-level rate limit policy, and then activate the managed API group.
- If a user-defined rate limit policy is selected at an API level and an operation inherits the API-level rate limit policy, API Center displays the selected policy name along with its configuration during the design of a REST API, managed API, and managed API group. If no policy is selected at the API level or operation level, API Center displays the organization-level rate limit policy as selected along with its configuration.