



Informatica® API Center  
July 2025

# REST API Assets

© 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](mailto: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: 2025-08-28

# Table of Contents

<b>Preface .....</b>	<b>4</b>
<b>Chapter 1: REST API assets.....</b>	<b>5</b>
Define REST API asset metadata. ....	5
API policy configuration. ....	6
Configure an API security policy. ....	6
Configure an API privacy policy. ....	6
Configure an API rate limit policy. ....	7
Creating an API operation. ....	7
Step1. Define the operation metadata. ....	8
Step 2. Define operation parameters. ....	8
Step 3. Define the operation request. ....	9
Step 4. Define the operation response. ....	10
Step 5. (Optional) Map an Application Integration process to the operation. ....	11
Step 5. (Optional) Map Data Quality assets to an operation. ....	13
Step 6. Configure operational policies. ....	13
Step 7. Configure a response timeout. ....	14
API definition. ....	15
Importing an API definition. ....	15
Importing and mapping an Application Integration process. ....	16
Importing Data Quality rule specifications. ....	17
Validating an API. ....	18
Publishing an API. ....	19
Versioning an API. ....	19
Editing an API. ....	20

# Preface

Read *REST API Assets* to learn how to design APIs to share with API consumers.

# CHAPTER 1

## REST API assets

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.

When you create a REST API asset, you configure the metadata of the API. You then create and define API operations, or import existing API definitions to the API.

Operations specify API consumer requests for access to a data collection. If you create an operation, you map a process to the operation. You can assign policies to the API and to the operations.

You must validate that an API doesn't contain any errors before you can publish it. You create managed APIs from published APIs.

For example, you can create a REST API asset that contains an API with a GET operation and a PUT operation. You then provide API consumers with access to the API endpoints of the operations.

## Define REST API asset metadata

Define the metadata of the REST API asset on the REST API page.

1. On the navigation bar, click **New > APIs > REST API > Create**.

The **REST API** page appears.

2. In the **API Name** field, enter a name for the REST API asset.

The **API Name** is the actual name of the REST API asset that is stored in the database. The name must be unique in the selected location. You can't edit the **API Name** after you create the API.

The **API Name** can contain up to 128 characters, including ASCII letters, digits, Japanese characters, hyphens, dashes, and underscores.

The **Name** field is autogenerated based on the **API Name** that you enter followed by the current version number of the API.

You can't edit the **Name** field.

3. From the **Location** field, click **Browse**, select a project or folder, and then click **Select**.

4. Optionally, in the **Description** field, enter a description.

5. The **Version** field is autogenerated.

The version number starts with v1 and is incremented when you create a new version of the API. You can't edit the **Version** field.

6. Click **Save**.

The REST API gets created.

# API policy configuration

You can configure and assign security and privacy policies to APIs.

A security policy defines the authentication methods that can be used to access the API. A privacy policy defines the sensitive data that API Center protects for the API.

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

## Configure an API security policy

Configure a security policy for an API on the REST API page.

1. Expand the **Policies** area and click **Security**.  
The **Authentication** panel appears.
2. Select one of the following options:
  - **None**. Select an authentication at the time of creating a managed API.
  - **Use existing**. Assign an existing security policy to the API.
  - **Create new**. Select one or more authentication methods to create a new security policy for the API.  
You can't use anonymous authentication with any other authentication method.
3. Optionally, in the **Notes** field, enter a description of the policy.
4. Click **Save**.

## Configure an API privacy policy

Configure a privacy policy for an API on the REST API page.

1. Expand the **Policies** area and click **Privacy**.  
The **Personally Identifiable Information (PII)** panel appears.
2. Select one of the following options:
  - **None**. Select a privacy policy at the time of creating a managed API.
  - **Use existing**. Assign an existing PII policy to the API.
  - **Create new**. Create a new policy for the API. For each type of information to protect, select the action to take for the request and the response. You can select different actions for the request and the response.  
Select one of the following actions:
    - **No action**. Don't take any action.
    - **Warning**. Issue a warning message that there was a privacy policy leakage in the request or the response. Don't block the request or response.
    - **Block**. Block the request or response and issue a message that the message was blocked because of a potential privacy policy breach in the request or the response.
3. Optionally, in the **Notes** field, enter a description of the policy.
4. Click **Save**.

## Configure an API rate limit policy

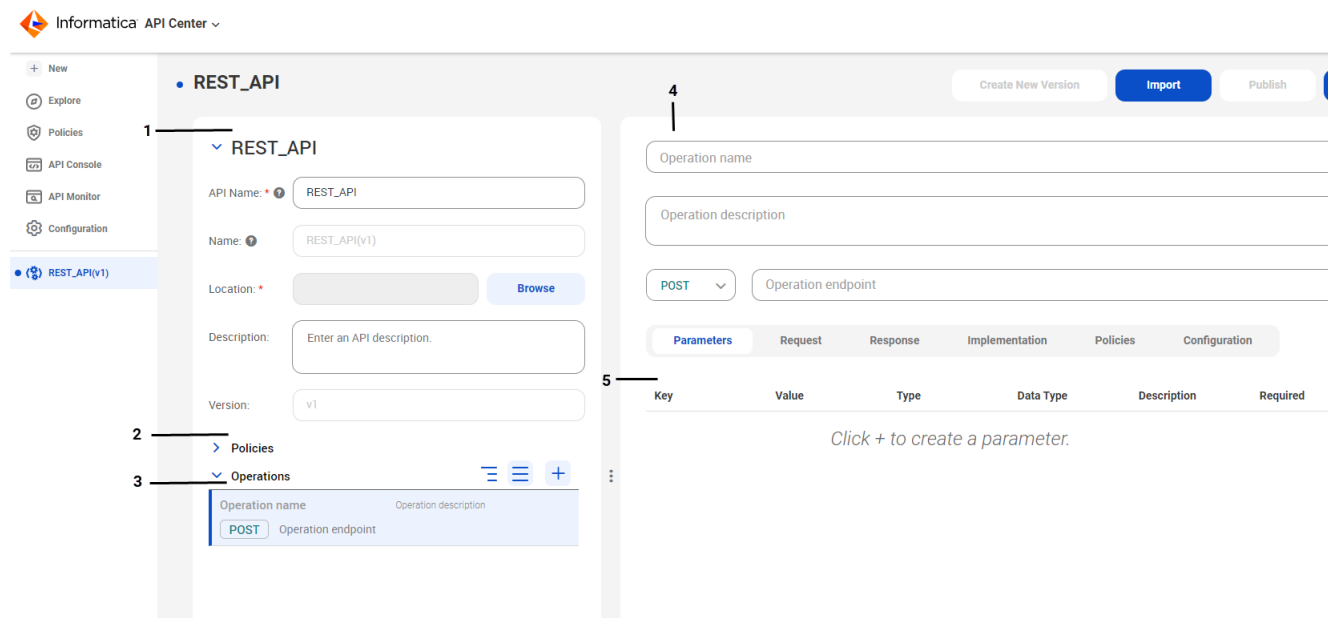
Configure a rate limit policy for an API on the REST API page.

1. Expand the **Policies** area and click **Privacy**.  
The **Rate Limit** panel appears.
2. Select a rate limit policy from the available list.
3. Optionally, enter notes.
4. Click **Save**.

## Creating an API operation

Use the REST API page to create an operation for an API.

The following image shows the REST API page with the operations area selected and the operations panel displayed:



1. REST API metadata area. Define the metadata of the REST API asset.
2. API Policies area. Configure the policies of the API.
3. Operations area. Create operations.
4. Operation panel. Define the metadata of an operation.
5. Operation tabs. Define the parameters, request, and response, map an Application Integration process, and configure operational policies and a response timeout for an operation.

To create an operation, you perform the following tasks:

1. Define the operation metadata.
2. Define the operation parameters.
3. Define the operation request.

4. Define the operation response.
5. Map an Application Integration process to the operation.
6. Configure operational policies.
7. Configure a response timeout.

Repeat the steps to create as many operations as needed for an API.

## Step1. Define the operation metadata

Define the metadata of an API operation on the operation panel.

1. In the **Operations** area of the REST API page, click **+**, or on the right panel of the REST API page, click **Create a New Operation**.
2. In the **Operation name** field, enter a name for the operation.  
The name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, hyphens, and underscores.
3. Optionally, in the **Operation description** field, enter a description.  
The operation description can contain up to 4,000 characters.
4. From the request methods list, select the request method.
5. In the **Operation endpoint** field, enter a URL in the following format: `/ {operation name} / {path parameter key} /`  
The operation path can contain alphanumeric characters, Japanese characters, and the following special characters: `.?&=_{ }~*`  
The operation endpoint path must be unique for an HTTP verb. The operation path must start with `/`, but can't end with `/`.  
**Note:** Note that you can use `*` only at the end of the operation path.
6. Click **Save**.

## Step 2. Define operation parameters

Define the parameters of an API operation on the **Parameters** tab of the operation panel.

1. On the operation panel, click **Parameters**.
2. Click **+**.
3. In the **Key** field, enter a name for the parameter.
4. From the **Type** list, select the parameter type.  
If you select **Path**, you can't add a value, the **Required** field is **True** by default and you can't change it.  
If you select **Query**, you can add a value, the **Required** field is **True** by default and you can change it.  
The path and query parameters do not get appended to the **Operation endpoint** field automatically. You need to add them manually to the **Operation endpoint** field.  
Query parameters are separated from the URL by a question mark (`?`). You can use the following formats for query parameters:
  - Query key (`?param`)
  - Query key with equals (`?param=`)



- Query key and value (?param=value)

Multiple query parameters can be included by separating them with an ampersand (&).

5. From the **Data Type** list, select a data type.
6. Optionally, in the **Description** field, enter a description.
7. Add as many parameters to the operation as required.  
To delete a parameter, click **Delete** on the row of the parameter.
8. Click **Save**.

## Step 3. Define the operation request

Define the operation request on the operation panel, including the header type and the request attributes. You can define the request body by adding attributes or by defining attributes in a JSON script.

1. On the operation panel, click **Request**.
2. In the **Headers** area, click **+**.
3. In the **Name** field, enter a name for the header.  
The request header name can contain only the following characters: ! # \$ % & ' \* + - . ^ \_ ` | ~  
A-Z a-z 0-9
4. From the type list, select a header type.
5. Optionally, in the **Description** field, enter a description.
6. Optionally, in the **Default Value** field, enter a value.  
The request header value must adhere to the HTTP specifications.
7. To add request body attributes, in the **Body** area, click **+**.
8. In the **Field Name** field, enter a name for the attribute. Assign a meaningful name, such as Order, Quantity, or Status.
9. From the **Data Type** list, select one of the following data types:
  - Array
  - Boolean
  - Date
  - Date Time
  - File
  - Integer
  - Number
  - Object
  - String

If you select **Array**, from the **Array Type** list, select an array type.

If you select **Object**, you can add as many child objects or fields under it as required.
10. Optionally, in the **Description** field, enter a description.
11. Optionally, select **Required** to make the attribute required. By default, the attribute isn't required.
12. Click **Add**.

13. Add as many attributes to the request body as required. The order of the attributes determines their order in the request body. Move up or move down the attributes as needed. To delete a request header or an attribute, click the **Actions** menu on the row of the header or attribute and then click **Delete**.
14. To define the request attributes in a JSON script, click **JSON** and add the request body fields as required. You must provide the fields according to the JSON format displayed on the **JSON** tab.
15. Click **Save**.

## Step 4. Define the operation response

Define the operation response on the operation panel, including the header type and the response attributes. You can define the response body by adding attributes or by defining attributes in a JSON script.

1. On the operation panel, click **Response**.
2. In the **Headers** area, click **+**.
3. In the **Name** field, enter a name for the header.  
The header name can contain only the following characters: ! # \$ % & ' \* + - . ^ \_ ` | ~ A-Z a-z 0-9
4. From the type list, select a header type.
5. Optionally, in the **Description** field, enter a description.
6. Optionally, in the **Default Value** field, enter a value.  
The response header value must adhere to the HTTP specifications.
7. To add response body attributes, in the **Body** area, click **+**.
8. In the **Field Name** field, enter a name for the attribute.  
You can't assign objects with the same name in the response body as the objects in the request body.
9. From the **Data Type** list, select one of the following data types:
  - Array
  - Boolean
  - Date
  - Date Time
  - File
  - Integer
  - Number
  - Object
  - String

If you select **Array**, from the **Array Type** list, select an array type.

If you select **Object**, you can add as many child objects or fields under it as required.

When you include the file data type in the request and response body fields during REST API creation, API Center stores API responses that contain content types such as multipart/form-data and application/octet-stream.
10. Optionally, in the **Description** field, enter a description.
11. Optionally, select **Required** to make the attribute required.  
By default, the attribute isn't required.
12. Click **Add**.

13. Add as many attributes to the response body as required. The order of the attributes determines their order in the response body. Move up or move down the attributes as needed.

To delete a response header or an attribute, click the **Actions** menu on the row of the header or attribute and then click **Delete**.

14. Optionally, to edit a response code, click **Edit Responses**. Select a response code from the **Response Code** list, edit the description in the **Description** field as needed, and click **OK**.

The default response code is 200 OK.

15. To define the response attributes in a JSON script, click **JSON** and add the response body fields as required.

You must provide the fields according to the JSON format displayed in the **JSON** tab.

16. Click **Save**.

**Note:** When you import an API definition with multiple response codes, you can view and edit all existing response codes. However, you can't add new response codes to the imported API definition. For more information about importing an API definition into API Center, see ["Importing an API definition" on page 15](#).

When you generate a new Application Integration process in API Center, the response code with the lowest configuration value is considered and populated in Application Integration. If you add or update any response code in the Application Integration process after mapping the process, the updated response codes appear in API Center after you refresh the process.

If you update an Application Integration process that was mapped to API Center prior to April 2025 release, you need to refresh the process to synchronize the data.

## Step 5. (Optional) Map an Application Integration process to the operation

Map an Application Integration process to the operation on the operation panel. You can create a new process and map it to the operation or map an existing process.

If you map an existing process, the process input fields must match the API request fields, the process output fields must match the API response fields, and the response code of the Application Integration process must match the response code of the API response.

**Note:** If you choose to generate a new process and use the same name for a request field and response field of a primitive data type, the process gets generated. However, when you edit the process, an error occurs. You must update the field names and generate the process again. If the conflicting field names are of a complex data type, the process does not get generated. You must update the field names and generate the process again. To avoid issues, always use unique names for the request fields and response fields.

1. On the operation panel, click **Implementation**.
2. Select to generate a new process or update an existing process.
  - **Generate a new process.**
    1. Select **Generate new**.
    2. In the **Name** field, enter a name for the process.  
The name can't exceed 80 characters, start with a number, or contain special characters.
    3. Click **Browse**, select a project or folder where the process must be created, and then click **Select**.
    4. Optionally, in the **Notes** field, enter a description of the process.

5. Click **Generate**.

API Center maps the API request to the process input in the **Input Mapping** area and the API response to the process output in the **Output Mapping** area.

After you generate a new process in API Center and publish the API, the changes reflect in Application Integration. When you update the **Input Mapping** or **Output Mapping** fields in Application Integration and publish the process, the data synchronizes in API Center and the updates appear in the **Request** and **Response** tabs in API Center.

- **Select an existing process.**

1. Select **Map existing**.
2. Click **Browse**, select the Application Integration process that you want to map, and then click **Select**. If any input or output mappings are updated after you map the process, click the **Refresh** button to refresh the process.

API Center maps the API request to the process input in the **Input Mapping** area and the API response to the process output in the **Output Mapping** area.

3. Optionally, in the **Notes** field, enter a description of the process.

When you select **Map existing**, you can choose to override the existing process. When you override an existing process, the following changes occur:

- The operation path doesn't change. You can change the operation path, if required.
- All existing details in the **Request** and **Response** tabs are updated based on the Application Integration process.
- All existing path and query parameters and request headers are removed from the **Parameters** tab.

3. Click **Save**.

## API Center and Application Integration asset conversion matrix

The following table shows the asset conversion matrix between API Center and Application Integration processes when you generate a new process, refresh a mapped process, or select an existing process:

API Center Asset	Generate New	Refresh Mapped Process	Map Existing
Request method	Retain API Center operation endpoint information for all HTTP methods.	Retain API Center operation endpoint information for all HTTP methods.	Synchronize details with the Application Integration process.
Path parameter	Retain API Center path parameter details.	Retain API Center path parameter details.	Clear all existing properties of API Center path parameter.
Query parameter	Retain API Center query parameter details.	Retain API Center query parameter details for all HTTP methods other than GET. For GET, synchronize details with the Application Integration process.	Clear all existing properties of API Center query parameter.
Request headers	Retain API Center request header details.	Retain API Center request header details.	Clear all existing properties of API Center request header.

API Center Asset	Generate New	Refresh Mapped Process	Map Existing
Request body	Retain API Center request body details.	GET method: NA All other methods: Synchronize details with the Application Integration process.	Synchronize details with the Application Integration process.
Response header	Retain API Center response header details.	Synchronize details with the Application Integration process.	Synchronize details with the Application Integration process.
Response body	Retain API Center response body details.	Synchronize details with the Application Integration process.	Synchronize details with the Application Integration process.
Response codes	Retain API Center response code details.	Synchronize details with the Application Integration process.	Synchronize details with the Application Integration process.

## Step 5. (Optional) Map Data Quality assets to an operation

You can map a Data Quality asset to an operation.

1. On the operation panel, click **Implementation**.
2. Select **Data Quality** as the provider.
3. In the **Mapped Rule** field, click **Browse** to select the Data Quality rule that you want to map, and then click **Select**.

If any input or output mappings are updated in API Center after you map the Data Quality rule, click the refresh button to refresh the rule specification.

4. Select the rule specification asset that has been deployed as a Data Quality API.
5. Click **Save**.

## Step 6. Configure operational policies

Configure policies for an API operation on the operation panel. You can configure security, privacy, rate limit, and response caching policies for an operation.

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

1. On the operation panel, click **Policies**.
2. To associate a security policy, in the **Authentication** tab, select one of the following options:
  - **Inherit**. Selected by default, the operation uses the same security policy as the API.
  - **Use existing**. Assign an existing security policy to the operation.
  - **Create new**. Create a new security policy for the operation. Assign one or more authentication methods to the operation to use as the security policy. You can't use anonymous authentication with any other authentication method.
  - **None**. Select an authentication at the time of creating a managed API.
3. To associate a rate limit policy to an operation while creating or updating a managed API, in the **Rate Limit** tab, select a rate limit policy from the existing list.

The values of the selected rate limit policy appear. If you leave the field blank, the organization-level rate limit policy values are shown and applied.

**Note:** 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.

4. To associate a response caching policy, in the **Response Caching** tab, select one of the following options:
  - Use existing. Assign an existing response caching policy to the operation.
  - Create new. Create a new response caching policy for the operation. In the **Caching Timeout** field, enter a timeout value from 1 through 3600 seconds.
  - None. Select a response caching policy at the time of creating a managed API.
5. To associate a privacy policy, in the **Privacy Settings** tab, select one of the following options:
  - Inherit. Selected by default, the operation uses the same Personally Identifiable Information (PII) policy as the API.
  - Use existing. Assign an existing Personally Identifiable Information (PII) policy to the operation.
  - Create new. Create a new Personally Identifiable Information (PII) policy for the operation. For each type of information that you want to protect, select the action to take for the request and the response. You can select different actions for the request and the response. Select one of the following actions:
    - Block. Block the request or response and issue a message that the message was blocked because of a potential privacy policy breach in the request or the response.
    - No action. Don't take any action.
    - Warning. Issue a warning message that there was a privacy policy leakage in the request or the response. Don't block the request or response.
  - None. Select a privacy policy at the time of creating a managed API.
6. Optionally, in the **Notes** fields, describe the policies.
7. Click **Save**.

## Step 7. Configure a response timeout

Configure a response timeout for an operation on the REST API page. If a response isn't received within the specified time, the request times out. The operation response timeout takes precedence over the organization response timeout.

1. On the operation panel, click **Configuration**.
2. In the **Timeout** field, enter a timeout value between 1 and 180 seconds.
3. Click **Save**.

# API definition

If you have an existing API definition in a file, you can import the API definition to quickly import multiple operations to the REST API on the REST API page.

When you import an API definition, you select the file that contains the definition, the URL that points to the definition, or APIs of IDMC services. You can import an API definition from the following sources:

- OpenAPI 3.0 JSON
- OpenAPI 3.0 YAML
- Swagger 2.0 JSON
- Swagger 2.0 YAML

After you specify the source file or URL, API Center parses the source and loads the operations. You select the operations to import.

## Importing Application Integration processes to API Center

You can import all the Application Integration processes to API Center. The processes on the **Service Name** corresponds to the **API Name** and the **Source Name** corresponds to the **Name** in Application Integration.

This import capability helps you easily transition the REST APIs of published Application Integration processes into API Center. You can easily design, publish, and create managed APIs in API Center.

## Importing an API definition

On the **REST API** page, import an API definition to create a REST API.

1. On the right panel of the REST API page, click **Import an API Definition** or **Import**.
2. Select the API definition type to import. In the **Source** field, select one of the following source types:
  - **File**. Select an API definition file from your local system.
  - **URL**. Specify a URL that points to the location of the API definition file.  
**Note:** If the URL points to a specification that requires authentication, you can't select the URL option. You must download the specification as a file by providing the authentication details, save it as local file, and then import it as a file.
  - **IDMC Services**. Select the IDMC service from which you want to import the APIs.
3. Click **Next**.  
The **Select Assets** tab appears. API Center parses the file and displays the operations on the **Select Operations** tab. If there are conflicts between the operations and the API, or any issue with the source file, API Center displays the warning details. Fix the API definition file and try again.  
When you select **Application Integration** as the source, all published processes in Application Integration appear. You can filter the data by service name.  
When you select **Data Quality** as the source, the deployment or API created on the Data Quality rule specifications appear. Select the deployed rule specification that you want to import.  
**Note:** If your organization is not provisioned for Data Quality, you can't import the rule specifications. Contact your organization administrator for information about provisioning your organization.
4. Select the operations to import and click **Next**.  
The **Summary** tab displays a summary of the assets that you selected.
5. Click **Import**.

API Center imports the selected policies and operations. You can update the policies and operations if needed. For more information, see [“Creating an API operation” on page 7](#).

**Note:** You can import operations from only one type of IDMC provider for an API. For a single API, you add operations either from an Application Integration process or from a Data Quality rule specification.

6. Click **Save**.

## Importing and mapping an Application Integration process

You can import all the published REST APIs of Application Integration to API Center. You can import all published processes in Application Integration.

1. On the right panel of the REST API page, click **Import an API Definition**, or click **Import**.
2. Select the API definition type to import. In the **Source** field, select IDMC Services.
3. Select **Application Integration** as the provider.
4. Click **Next**.

In the **Select Assets** tab, all published processes in Application Integration appear. The list displays the service name, source name, status, protocol, and description of each published Application Integration process. You can filter the data by service name.

The following table shows the fields in API Center with their corresponding fields in Application Integration:

API Center Field	Application Integration Field
Source Name	Process Name
Service Name	API Name
Description	Description in the <b>General</b> tab of <b>Start Properties</b> .

5. Select an asset to import and click **Next**.

You can select only one asset at a time.

The **Summary** tab displays a summary of the assets that you selected.

6. Click **Import**.

API Center imports the selected assets and the related Swagger definitions from Application Integration and populates the operation metadata area.

The following table explains the correlation between the API Center and imported Application Integration assets:

API Center Tabs	Application Integration Field
Request method	Auto-populated from the Swagger definition.
Parameters	NA. Application Integration doesn't support any query or path parameters.
Request	All properties defined in the <b>Input Fields</b> of the Application Integration process. Required fields in Application Integration appear as True in API Center.



API Center Tabs	Application Integration Field
Response	All properties defined in the <b>Output Fields</b> of the Application Integration process. Required fields in Application Integration appear as <b>True</b> in API Center. Response headers present in Application Integration appear in API Center. When you import an API definition with multiple response codes, you can view all the response codes. You can edit the response codes. However, you can't add any new response on the <b>Response</b> tab.
Implementation	The imported Application Integration process is already mapped and all the relevant fields are populated accordingly.
Policies	NA. API Center retains its default policy configurations.

- Click **Save**.
- Validate and publish the API.

## Importing Data Quality rule specifications

You can import rule specifications that you enabled in Data Quality to API Center to design an endpoint. After you publish the API endpoint, you can integrate the rule specification capabilities within your application.

- On the right panel of the REST API page, click **Import an API Definition** or **Import**.
- Select the API definition type to import. In the **Source** field, select **IDMC Services**.
- Select **Data Quality** as the provider.
- Click **Next**.

On the **Select Assets** tab, all deployed rule specifications that you enabled in Data Quality appear.

- Select an asset to import and click **Next**.

You can select only one asset at a time.

The **Summary** tab displays a summary of the assets that you selected. The summary includes the source name, provider name, asset name, description, operation name, and warnings, if any.

- Click **Import**.

API Center imports the selected assets and populates the operation metadata area based on the Data Quality rule's Open API specifications.

The following table explains the correlation between the API Center and imported Data Quality rule specification assets:

API Center Tabs	Data Quality Field
Request method	Operation path auto-populated from the Data Quality rule specification.
Parameters	NA. Data quality doesn't support query parameters and path parameters.
Request	All properties defined in the <b>Inputs</b> field of the Data Quality rule set. Required fields in Data Quality appear as <code>False</code> in API Center.

API Center Tabs	Data Quality Field
Response	All properties defined in the <b>Outputs</b> field of the Data Quality rule set. When you import an API definition with multiple response codes, you can view all the response codes. You can edit the response codes. However, you can't add a new response on the <b>Response</b> tab.
Implementation	The imported Data Quality rule specification is already mapped and all the relevant fields are populated accordingly. The <b>Mapped Rule</b> field indicates the mapped Data Quality rule specification. You can click on the rule to navigate to the configuration workspace for the rule specification in Data Quality. For more information about configuring a rule specification in Data Quality, see <a href="#">Rule specification assets</a> .
Policies	All Data Quality operations must contain a session ID based authentication policy. You can either create a new session ID or inherit and use an existing session ID. For more information about session ID authentication, see <a href="#">API Policies</a> .

- Click **Save**.
- Validate and publish the API.

**Note:** Effective in the July 2025 release, import of Data Quality rule specifications directly into API Center is available for preview.

Preview functionality is supported for evaluation purposes but is unwarranted and is not supported in production environments or any environment that you plan to push to production. Informatica intends to include the preview functionality in an upcoming release for production use, but might choose not to in accordance with changing market or technical circumstances. For more information, contact Informatica Global Customer Support.

## Validating an API

If an API contains errors, fix the errors listed on the REST API page. You can only publish valid APIs.

- On the **Explore** page, navigate to the API.
- On the **Actions** menu, click **Edit**.
- Make changes as needed and click **Save**.
- If there are validation errors, in the **Validation** area, click the error rows and take the required corrective action.
- Click **Save**.

# Publishing an API

You can publish a valid API from the REST API page. When you update an API, you must republish the API for the changes to take effect. You can create a managed API from the published API and share the API endpoint URLs with API consumers.

1. On the **Explore** page, navigate to the API.
2. On the **Actions** menu, click **Edit**.
3. Click **Publish**.

Alternatively, if you created an API on the **New > APIs > REST API > Create > Create API** page, validate the API, and then click **Publish**.

The **Publish** dialog box appears.

4. Edit the published API name, if required.

The published API name can contain up to 50 characters, including ASCII letters, digits, Japanese characters, hyphens, dashes, and underscores. You can't publish an API if the name of the API starts with a digit.

You can't edit the published API name after you publish the API.

5. Optionally, add notes.
6. Click **Publish**.

API Center publishes the API. The published API appears under the **Published APIs** tab on the **API Console** page.

# Versioning an API

You can create multiple versions of a REST API without impacting the existing API definition. You can create multiple versions of an API. You can create a new version of an API from the most recent version of the API.

When you change or update any parameter of an existing API which impacts the existing definition of the API, create a new version of the API.

When you create a new version of an API, a copy of the existing API version gets created. The API name remains the same and the **Name** field gets updated with the next immediate version number. You can make necessary changes to the new version and then publish the API.

If you edit a previous version of an API, the changes to that particular version doesn't impact the any other versions of the API.

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.

1. On the **Explore** page, navigate to the API.
2. On the **Actions** menu, click **Create New Version**.

Alternatively, if you published an API on the **New > APIs > REST API > Create > Create API** page, select **Create New Version**.

The version number of the API updates to the next immediate version.

If you have an updated version of the API, the **Create New Version** option on the **Actions** menu of a previous version of the API appears disabled.

3. Update the API details, and click **Save**.  
You can now publish and create a managed API.

## Editing an API

You can edit an API from the API definitions page.

1. On the **Explore** page, navigate to the API, and then on the **Actions** menu, click **Edit**.
2. Change the description or operations as needed.

You can also create a new version of the API.

3. Click **Save > Publish**.

The **Publish** dialog box appears.

4. Optionally, add notes.

You can't edit the published API name.

5. Click **Publish**.

API Center publishes the updated API and the API appears under the **Published APIs** tab on the **API Console** page.