Oracle® Digital Experience for Communications Industry Fabric

Implementation Guide

Release 25D
G38460-02
October 2025

ORACLE

Oracle Digital Experience for Communications Industry Fabric Implementation Guide, Release 25D

G38460-02

Contents

About This Content

The guide provides you with the concepts and procedures required to secure the access to your applications, extend these applications, and integrate them with other applications using CX Industries Framework.

Audience

This guide is intended for application developers, system integrators, and system administrators involved in configuration and integration of applications.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview of CX Industries Framework

Overview

This chapter provides an overview of Oracle CX Industries Framework and the tasks you must perform to integrate applications.

CX Industries Framework serves as a platform for seamless integration of applications. You can use this framework to integrate your application with other Oracle applications and third-party applications, which includes cloud services and on-premise systems. It connects the applications and coordinates their interactions to deliver flexible digital experiences. It uses data from various sources, supports TM Forum Open APIs, and provides REST APIs for you to configure and extend tailored experiences.

CX Industries Framework supports both synchronous and asynchronous interactions between the integrated applications. It hosts all the TM Forum Open APIs on Oracle Cloud Infrastructure API Gateway to secure and optimize the API interactions. It also ensures high availability and resiliency by using the Oracle Cloud Infrastructure Load Balancing service. Here are the key tasks that you can accomplish with this framework:

- Add your own built applications to the topology
- Integrate applications with external applications
- Route requests from the TM Forum Open APIs to the corresponding run time applications
- Publish data from events to the corresponding run time applications
- Setup non-TM Forum APIs as custom APIs
- Use custom adapter to connect to non-TM Forum external applications

Before You Start

Here are some points to consider before you get started with your application setup:

- If you're using your existing corporate directory service as the primary identity provider, federate your directory with Oracle Cloud Infrastructure. Refer to the Oracle Cloud Infrastructure Documentation for information about how to federate your directory with other identity providers.
- If you're integrating a non-TM Forum external application, build an adapter for the external application to connect and interact with your application.
- If you're integrating an on-premises system, connect your cloud network to your on-premises network by using a virtual private network connection or FastConnect. Refer to the Oracle Cloud Infrastructure Documentation for information about setting up VPN connect or FastConnect.

> **Note**
>
> Your application is automatically provisioned and deployed in your environment. You can use it as an independentservice or configure it to work with your existing applications based on your business needs. You can contact Oracle Consulting Services for help with building or adding your own applications and configuring your application to work with existing applications.

Where You Perform Setup Tasks

Use the CX Industries Framework REST APIs to perform all your setup tasks.

Here are the REST APIs you can use to set up your implementation:

- **apis:** Use this API to get a list of all the TM Forum (TMF) and user-defined APIs (TMF or non-TMF) that are accessible through CX Industries Framework. You can also use this API to add, modify, or delete user-defined APIs. For example, you can add a user-defined API to:
  - Use a non-TMF API for routing billing requests to your billing software.
  - Configure an adapter for a non-TMF external application to connect and interact with your communications applications. This can be for billing, customer care, customer data management, or other purposes based on your need.
- **systemDescriptors:** Use this API to define the target domain and the target application you want to integrate with your communications applications. For example, you can use this API to integrate a custom tax engine, billing software, or Customer Relationship Management (CRM) software with your communications applications. You can also use this API to update or remove target application details.
- **connectionDescriptors:** Use this API along with the systemDescriptors API to specify how to connect to your target application. You must specify the endpoint for each external application. The endpoint can be an API or an adapter deployed in your target application. You can also use this API to update or remove endpoint details.
- **listenerRegistrations:** Use this API to register the target application to listen to events. You can configure multiple applications to register for the same event. You must specify the event types to be notified to the target application and the endpoint to listen to the events. You can also use this API to modify or delete listeners.
- **gatekeepingRules:** Use this API to review or update gatekeeper rules. This rule provides a mapping to the gatekeeper for routing requests and publishing events. It ensures that only the events generated by gatekeepers are published to event listeners.
  These events aren't republished to the gatekeepers which generated them. The gatekeeper rules are autogenerated when you integrate an application. You can use this API to set an external application as a gatekeeper or to update the API resources for an internal or external application.
- **routingCriteria:** Use this API to update the routing conditions for non-TMF APIs only.
- **utilityConfigurations:** Use this API to view and update the utility configurations. For example, you can update the general settings, such as setting the inactive time period, using this API. You can also view and update other default utility configurations.

Setup Tasks List

This table summarizes the tasks you can do to set up your application and where you can find more information on the task. You can use this table as a checklist to understand the sequence of the tasks.

Table 1-1 Checklist to Setup Tasks List

Setup Activity	Tasks	Read More
Secure access to your application	• Create Confidential application • Set up authentication protocol	Secure Access To Your Application
Create users and associate roles	• Create user accounts. • Review, edit, lock, or delete existing user accounts. • Assign predefined roles to user accounts. • Set password policies.	Create Users and Assign Roles
Integrate external applications	• Integrate external customer data management, billing, or other applications as needed. • Register the external applications to listen to events as needed.	Integrate External Applications
Set up gatekeepers for event publishing	Create rules for the gatekeepers (internal or external) to route requests or publish events to event listeners based on your business needs.	Set Up Gatekeepers

Get Help from Oracle

You can find answers to your questions about CX Industries Framework using the resources in My Oracle Support.

If you still couldn't find answers to your questions, seek help from Oracle by creating a service request that includes all your questions. Sign in to the My Oracle Support Website and create a service request. Be sure to include the necessary details and log files.

2

Secure Access To Your Application

How to Secure Access from External Applications Using OAuth

You can use the OAuth protocol to authorize external applications to access your communication application's REST APIs. Authorization ensures that an application is granted access to a service.

When you submit requests from external applications, the request must include an OAuth access token. To generate that access token, you need OAuth credentials. You can generate these credentials by adding your external application as a confidential application in Oracle Identity Cloud Service. You can then embed the generated OAuth credentials in the external application to generate access token while accessing your communication application's REST APIs.

You can use the OAuth credentials of the DX4C_FABRIC_<Customer_ID> application predefined in Oracle Identity Cloud Service for testing the REST API calls. The <Customer_ID> is a unique identifier of the CX Industries Framework instance. You can test the calls using the cURL command or REST API clients, such as Postman. For instructions on calling REST APIs using cURL command or REST API clients, refer to Quick Start in REST API for Oracle CX Industries Framework. However, you must add a confidential application for every external application that sends request to your communications application.

You must be an administrator to perform this task. Here's how you can create a confidential application:

1. In the Identity Cloud Service console, expand the **Navigation Drawer**, and then click **Applications**.
2. On the Applications page, click **Add**.
   The Add Application page appears.
3. From the list of application types, select **Confidential Application**.
4. On the App Details page, enter a name and a description for your application.
5. At the top of the Add Confidential Application wizard's Details page, click **Next**.

   A confirmation message indicates that the application has been added in a deactivated state.
6. On the Add Confidential Application wizard's Client page, click **Configure this application as a client now**, and enter this information in the Authorization section:
   - **Allowed Grant Types:** Select **Client Credentials**, **JWT Assertion**, **Refresh Token**, and **Authorization Code**.

     > **Note**
     >
     > To generate refresh tokens, you must select **Refresh Token**.

   - **Allow non-HTTPS URLs:** Select this option.

- **Redirect URL:** Enter the URL of the application where responses to authentication requests are sent. The callback URL for the application where the authorization code and authorization token is sent.
- **Post Logout Redirect URL:** Enter the URL where you want to redirect the user after logging out of the application.

7. In the Token Insurance Policy section, under **Authorized Resources**, select Specific.
8. In the Resources section, Click **Add Scope** and follow these steps:
   a. Search for the DX4C_FABRIC_<Customer_ID> application predefined in Oracle Identity Cloud Service.
   b. Click the arrow to the right of the application name.
   c. Select the scope that ends with **all**. The scope looks like this: **`urn:opc:resource:consumer::all`**. You can select additional scopes as needed.
   d. Click **Add**.
9. In the Grant the client access to Identity Cloud Service Admin APIs section, click **Add** and follow these steps:
   a. Select **Authenticator Client**.
   b. Click **Add** and then click **Next**.
10. In the Expose APIs to Other Applications section, leave the default **Skip for later** and click **Next**.
11. In the Web Tier Policy section, leave the default **Skip for later** and click **Next**.
12. On the Add Confidential Application wizard's Authorization page, click **Finish**.

    The Application Added page appears.
13. Make note of the client ID and client secret. You must embed these in the external application to access your communication application's REST APIs.
14. On the Details page for your new application, select **Activate** and confirm the activation.
15. Embed the following in the external application that accesses your communication application's REST APIs:
    - The Oracle Identity Cloud Service URLs for generating authorization codes and requesting OAuth access tokens. For example:

      `https://<idcs_hostname>/oauth2/v1/authorize`

      `https:/<idcs_hostname>/oauth2/v1/token`

      where **`<idcs_hostname>`** is the server of your Oracle Identity Cloud Service instance.
    - The redirect URL to send authorization codes and access tokens.
    - The client ID and client secret generated by the confidential application.

How to Access External Applications Using OAuth

If your external application is secured using an identity provider such as Oracle Identity Cloud Service, use the OAuth protocol to securely access your external server application.

Here are the things you must ensure that you do to access your external application:

- Specify the OIDC client credentials when you integrate your external application with your communications application. When you set up the connection descriptor, specify the

following details about the OIDC client application that's used to secure your external application:

  - Client ID
  - Client Secret
  - OAuth Scope to access your application
  - Identity URI to request the access token
    For more information, see the Integrate External Applications chapter within this guide.

- If your identity provider is Oracle Identity Cloud Service, add your OIDC client application as a trusted application in Oracle Identity Cloud Service.
- Register your communications application's signing certificate in the OIDC client of your external application. To get the signing certificate for your communications application, create a service request on My Oracle Support at https://support.oracle.com.

Related Topics

- How to Set Up Federation With External Identity Provider
- https://www.oracle.com/pls/topic/lookup?ctx=fa22b&id=s20076638

How to Set Up Basic Authentication

You can also use basic authentication to secure access from external applications.

The basic authentication doesn't require any configuration in Oracle Identity Cloud Service. You can use any user account created for your application to access REST APIs. You must provide the account credentials (user name and password) when you integrate the external application with your application. For more information, see the Integrate External Applications chapter in this guide.

For basic authentication, you pass the account credentials in the header of the HTTP request. These user credentials are converted into a basic authentication header and the sign-in URL is called to pass the basic authentication header to the authentication service. On authentication, the external application can access your application's REST APIs.

3

Set Up Federation With External Identity Provider

Identity Providers

You can use Oracle Identity Cloud Service as your primary identity provider for federating users and Oracle Identity and Access Management to manage users. However, if you have an existing identity provider that's compliant with Security Assertion Markup Language (SAML) 2.0, you can use that as your primary identity provider.

To use a third-party identity provider, you must associate that identity provider with Oracle Cloud Infrastructure. This lets you use the existing users in your identity provider or directory service. You don't have to recreate them in Oracle Identity Cloud Service. Instead, enable SAML Just-In-Time Provisioning to automate the user creation in Oracle Identity Cloud Service. Remember that you can't perform bulk migration of users with this setup.

After you set up the federation, configure default access rules and provide mapping to a predefined role in your application based on the user's role in your directory service. You must map restricted roles to limit access to your application. For more information, see the Create Users and Assign Roles chapter.

How to Set Up Federation With External Identity Provider

You can set up federation between Oracle Identity Cloud Service and the external identity provider in one of these ways:

- Set up the external identity provider as the primary identity provider. This enables you to manage users in your external identity provider and synchronize them with Oracle Identity Cloud Service.
- Set up the external identity provider as a service provider. This enables you to manage users in Oracle Identity Cloud Service and synchronize them with your external identity provider.

> **Note**
>
> You must have administrator credentials for your Oracle Identity Cloud Service tenancy and the external identity provider to perform this task.

Set Up External Identity Provider as Primary Identity Provider

Here's a general process that you go through to set up an SAML2 identity provider. The actual steps may vary depending on the external identity provider that you want to set up.

1. In your external identity provider, configure the external identity provider as an SAML2 identity provider and download its metadata file in the XML format. You need this metadata file later in these steps.

2. In the Oracle Identity Cloud Service console, add and configure an SAML2 Identity Provider. While configuring the identity provider, ensure that you do these steps:
   a. Import the metadata file that you downloaded in step 1 into Oracle Identity Cloud Service.
   b. Download the following files from Oracle Identity Cloud Service. You need these files later in the steps. For detailed steps, refer to the Add a SAML Identity Provider topic in the Administering Oracle Identity Cloud Service guide.
      - Service Provider Metadata (in the XML format)
      - Service Provider Signing Certificate (in the PEM format)
      - Service Provider Encryption Certificate (in the PEM format)
3. In your external identity provider, configure Oracle Identity Cloud Service as the remote service provider. While configuring the remote service provider, ensure that you import these files that you downloaded in step 2 into your external identity provider. Ensure that you import Service Provider Signing Certificate and Service Provider Encryption Certificate into your external identity provider's keystore. For example:

   ```
   keytool -import -alias -file IDCS_IDP_FILES/IDCSCertificate.pem -keystore
   IDCS_IDP/keystore.jceks - storetype JCEKS -storepass <storepass>
   ```

4. Verify the federation setup by following these steps:
   a. In the Identity Cloud Service console, navigate to **Security** and then select **Identity Provider**.
   b. Select your external identity provider and then select Test from the right-click menu. Your external identity provider sign-in page appears.
   c. Sign in with the user in your directory service. The connection successful confirmation message appears.

For detailed steps, refer to your external identity provider documentation and the Federating with SAML 2.0 Identity Providers chapter in the Oracle Cloud Infrastructure documentation.

With this setup, you can use SAML Just-In-Time Provisioning to automate user creation in Oracle Identity Cloud Service. For enabling this feature, refer to the Service Request Features for Oracle Identity Cloud Service topic in the Administering Oracle Identity Cloud Service guide. For configuring this feature, see the Configuring SAML JIT Provisioning topic in the REST API for Oracle Identity Cloud Service documentation.

Set Up External Identity Provider as Service Provider

Here's a general process that you go through to set up an SAML2 service provider. The actual steps may vary depending on the external identity provider that you want to set up.

1. In the Oracle Identity Cloud Service console, configure Oracle Identity Cloud Service as the primary identity provider by following these steps:
   a. Add a SAML Application in Oracle Identity Cloud Service. For detailed steps, refer to the Add a SAML Application topic in the Administering Oracle Identity Cloud Service guide. While you add the application, ensure that you download these files:
      - Identity Provider Signing Certificate (in the PEM format). This certificate is used by the SAML application to verify that the SAML assertion is valid.
      - Identity Provider Metadata (in the XML format). You need this metadata file later in the steps. The URL looks like:

        ```
        https://<your_tenancy>.identity.oraclecloud.com/fed/v1/metadata.
        ```

b. Assign the SAML application to your users in Oracle Identity Cloud Service. For detailed steps, refer to the Assign Applications Oracle Identity Cloud User Using Account Form topic in the Administering Oracle Identity Cloud Service guide.

2. In your external identity provider, do the following:

   a. Import the signing certificate that you downloaded in step 1 into your external identity provider's keystore. For example:
   `keytool -import -alias -file IDCS_IDP_FILES/IDCSCertificate.pem -keystore IDCS_IDP/keystore.jceks -storetype JCEKS -storepass <storepass>`

   b. Configure Oracle Identity Cloud Service as the remote identity provider. While configuring the identity provider, ensure that you do these steps:

      - Select SAML as the single sign-on method.
      - Import the Oracle Identity Cloud Service Identity Provider Metadata file that you downloaded in step 1 into your external identity provider.
      - Enter the Oracle Identity Cloud Service console URL as the Sign-on URL.

For detailed steps, refer to your external identity provider documentation.

4

Create Users and Assign Roles

About Roles

You use the Role-Based Access Control model to protect your application. Roles control the access that you have to different features of your application.

This table describes the job roles that you can assign for integrating and configuring applications. You can create the usersand assign only these predefined roles to them. These roles are created as groups in Oracle Identity Cloud Service.

Table 4-1 Job Roles for Integrating and Configuring Applications

Job Role	Job Role Code	Description
DX4C_Configuration_Endpoint_Read	<Workspace-ID>_DX4C_Configuration_Endpoint_Read For example: DX-PROD_DX4C_Configuration_Endpoint_Read	Views the endpoint and system configuration.
DX4C_Configuration_Endpoint_Write	<Workspace-ID>_DX4C_Configuration_Endpoint_Write For example: DX-PROD_DX4C_Configuration_Endpoint_Write	Views or updates the endpoint and system configuration.
DX4C_Configuration_Eventing_Read	<Workspace-ID>_DX4C_Configuration_Eventing_Read For example: DX-PROD_DX4C_Configuration_Eventing_Read	Views the event listener configuration.
DX4C_Configuration_Eventing_Write	<Workspace-ID>_DX4C_Configuration_Eventing_Write For example: DX-PROD_DX4C_Configuration_Eventing_Write	Views or updates the event listener configuration.
DX4C_Configuration_API_Registration_Read	<Workspace-ID>_DX4C_Configuration_Registration_Read For example: DX-PROD_DX4C_Configuration_Registration_Read	Views TM Forum (TMF) Open APIs and the non-TMF custom APIs registered in the application.

Table 4-1 (Cont.) Job Roles for Integrating and Configuring Applications

Job Role	Job Role Code	Description
DX4C_Configuration_API_Registration_Write	<Workspace-ID>_DX4C_Configuration_Registration_Write For example: DX-PROD_DX4C_Configuration_Registration_Write	Views,updates, or removes the non-TMF custom APIs registered in the application.

Table 4-2 Job Roles for Different Users

User	Job Roles	Description
TMF Specialist	• DX4C_Configuration_Endpoint_Read • DX4C_Configuration_Endpoint_Write • DX4C_Configuration_Eventing_Read • DX4C_Configuration_Eventing_Write • DX4C_Configuration_Routing_Read	Performs TM Forum Open APIs specific configuration, such as reviewing routing and gatekeeper rules and setting up event listeners.
System Configuration Viewer	• DX4C_Configuration_Routing_Read • DX4C_Configuration_Endpoint_Read • DX4C_Configuration_Eventing_Read • DX4C_Configuration_API_Registration_ Read	Views the following configuration in a restricted and read-only manner: • Integratedapplications • Gatekeeperrules • Event listeners
System Administrator	• DX4C_Configuration_Routing_Read • DX4C_Configuration_Endpoint_Read • DX4C_Configuration_Endpoint_Write • DX4C_Configuration_Eventing_Read • DX4C_Configuration_Eventing_Write • DX4C_Configuration_API_Registration_ Read • DX4C_Configuration_API_Registration_ Write	An administration user who performs all the configuration required to: • Set up or modify custom APIs (TMF or non-TMF) or gatekeepers. • Route requests to integrated applications. • Publish events to listeners.

Create Users and Assign Roles

As an initial user or administrator, you create users and assign them the predefined roles to perform certain necessary tasks in your application. You use the Security Console to create users.

After you have signed up with your Oracle cloud service, you receive the user name and password for one initial user. This user is provisioned with the job role necessary to perform the necessary setup tasks, including creating users.

Here's how you can create users. You must be an initial user or administrator to do this task.

> ⚠ **Caution**
>
> - Create unique users for each environment.
> - Assign the users only the roles required to perform their assigned tasks.

1. Go to **Navigator Tools** > **Security Console**.
2. In the Users tab, click **Add User Account**.
3. Specify the following details to create a user:
   a. Specify the required details, such as **First Name**, **Last Name**, **Email**, and **User Name**.
   b. Select the **Associated Person Type** as Employee.
4. Enter any user-defined password for the account and then confirm the password.

   The user is now created. You can assign individual roles to this user or assign the user to a group that contains all the relevant roles for this user, for example, DX4C_System_Administrator. If you are assigning individual roles, go to next step. If you are assigning a group, skip adding roles and save the changes. See the following topics for creating groups and assigning roles and users.
5. Click **Add Role**.
6. On the Add Role Membership dialog box, search and select the appropriate roles for the users.
7. Click **Add Role Membership**. A confirmation dialog appears.
8. Click **OK** and then **Done**.
9. Click **Save and Close**.
10. To assign additional roles to the user, select the same account and repeat steps 5 to 9.

    Selected roles or groups are assigned to the user.

Create Groups and Assign Roles

Here's how you create groups and assign roles to that group:

1. In the Oracle Cloud Infrastructure Console, click the navigation menu icon, navigate to identity and security, then under **Identity**, click **Federation**.
2. On the Federation page, click **OracleIdentityCloudService**.
3. On the identity provider details page, click the **Oracle Identity Cloud Service Console** link.
   The Oracle Identity Cloud Service Console opens in a new window.
4. In the Identity Cloud Service Console, click navigation menu icon, and then click **Groups**.
5. To create a group, click **Add**.
6. Enter a name and description for the group that outlines the purpose of this group.

7. To allow users to request access to this group, click **User can request access**.
8. Click **Finish**.

Now you can assign roles to the group you created. Though you can assign roles to the users directly, it's easier to manage assignments when you create a group for roles and then assign roles and users to those groups.

Here's how you assign roles to a group:

1. In the Identity Cloud Service console, click navigation menu icon, and then click Applications.
2. Open the Oracle Identity Cloud Service application defined for your application.
3. Click the Application Roles tab.
4. Next to the role you want to assign, click action menu icon, and then select Assign Groups.
5. Find and select the group you just created, and then click **Assign**. For a description of the predefined roles, see the About Roles topic.

Assign Users to Groups

Assign users to groups to automatically assign them the appropriate roles and permissions.

Here's how you assign users to a group:

1. In the Identity Cloud Service console, click navigation menu icon, and then click **Groups**.
2. Open the group you want to assign users to.
3. Click **Users** and then **Assign**.
4. Select the users you want to add, and then click **OK**.

Remove Users

You use Oracle Identity Cloud Service to remove users from your application.

Only the initial or administrative user can perform this task. You can remove users in either of these ways:

- *Deactivate users*: The user and all related information remain in the application. The user can be reactivated in the future.
- *Delete users*: The user and all related information are removed from the application. The user can't be reactivated.

Refer to the chapter Manage Oracle Identity Cloud Service Users in Administering Oracle Identity Cloud Service for more information.

Set Up Password Policies

Here's the default password policy for your application:

- Minimum length: 8 characters
- Minimum numeric characters: 1
- Minimum alphabet characters: 2
- Minimum uppercase characters: 1

- Minimum lowercase characters: 1
- Minimum special characters: 1

For information about modifying this policy, refer to the chapter Managing Oracle Identity Cloud Service Password Policies in *Administering Oracle Identity Cloud Service*.

5

Set Up Custom APIs

Create a Custom API

You can access the TM Forum (TMF) Open APIs using the proxy endpoints provided by the application. You can use these APIs in various scenarios to support your business needs. You can also deploy TMF or non-TMF APIs for your communications application by using CX Industries Framework APIs.

For example:

- You can add a custom TMF API, which adheres to the TMF Open API structure, to perform address validation for tax calculation by setting it to connect with taxation gateway.
- You can add a Siebel non-TMF API, which follows the open API standards, to get a catalog by setting it to connect with Siebel.

> **Note**
>
> You can use TMF APIs to route requests and publish events. However, you can use the non-TMF APIs only to route requests.

Here's how you create a third-party API or a custom API:

1. Create the payload for registering the custom API with the details described in this table.

Field	Description
`api-name`	The name of the user-defined API.
`api-version`	The version of the API.
`openapi-document-url`	A link to the document that describes the API (optional).
`api-id`	The identifier of the API, for example, `siebel-100`.
`alternative-root-path`	The alternative URL path.
`api-events`	The name of the event that you want to publish through this API, such as individualCreateEvent.
`api-resources`	• The name of the resource you want to associate with this API, such as Individual. • The ID and path to that resource, for example, /individual. **Note:** You must add at least one resource for the API.

2. Call the apis API by running this cURL command or by using a REST API client:
```
curl-H Authorization: Bearer <accessToken> https://<hostName>/admin/
{workspace}/apis/ -X POST -H "Content-Type: application/json" -d @api-
data.json
```

Where:

- <accessToken> is the OAuth access token for your account.
- <hostName> is the URL for your CX Industries Framework API Gateway.
- <workspace> is the path parameter for the production or test workspace. For example, 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

The API returns a response with the custom API ID. You can associate this API with an external application to route requests generated by this API to that application. For more information, see Related Topics.

Here is a sample request payload for registering a custom API:

```
{
"api-name": "siebel",
"api-version": "v1",
"openapi-document-url": "https://wolpertinger/.    ",
"api-id": "siebel-100",
"alternative-root-path": "siebel/v1.0", "api-events" : [],
"api-resources": [
{
"name": "workflow-process-manager", "resource-id": "res-1",
"resource-path": "service/Workflow Process Manager/RunProcess"
},
{
"name": "price-list",
"resource-id": "res-2",
"resource-path": "data/Price List/Price List",
"routing-ambiguity-resolution-strategy": "HTTP400BadRequest" # To mention
the required Routing Ambiguity type, to be defined per resource , if not
defined will be considered a default HTTP400BadRequest
},
{
"name": "price-list-by-id",
"resource-id": "res-3",
"resource-path": "data/Price List/Price List/{id}/Price List Item", "path-
parameters": [
{
"parameter-name": "id", "parameter-type": "string", "optional" : false
}
]
},
{
"name": "swi-product-class", "resource-id": "res-4",
"resource-path": "service/SWI Product Class/QueryByExample"
},
{
"name": "swi-product-attributes", "resource-id": "res-5",
"resource-path": "service/SWI Product Attributes/QueryByExample"
},
{
"name": "swi-catalog-admin", "resource-id": "res-6",
"resource-path": "service/SWI Catalog Admin/QueryByExample"
},
{
```

```
"name": "swi-product",
"resource-id": "res-7",
"resource-path": "service/SWI Product/QueryByExample"
}
]
}
```

Related Topics

- How You Integrate External Back-End Applications

Add Custom API to System Description

If you want the custom API to interact with an existing external application, you must update the system description to add your custom APIs. You use the systemDescriptors API to update this description.

> **Note**
>
> Ensure that you don't update default system descriptors and connection descriptors to integrate any new external application. To replace a pre-integrated target application with an external application, create new descriptors by using the POST method of systemDescriptors and connectionDescriptors APIs. In case, if you lose a default system descriptor or connection descriptor, create a service request on My Oracle Support to revert your changes.

If you're using the custom API in a complete new integration, see Integrate External Applications chapter in this guide. Here's how you can add the custom API to the existing system descriptor:

1. Create a payload with the target application details. You can create the payload by copying the application's systemdescriptor and adding the new custom API in the "offered-apis" section along with the url-prefix for the API.
2. Call the systemDescriptors API by running the following curl command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken https://<hostName>/admin/
{workspace}/systemDescriptors/{id} -
X PUT -H "Content-Type: application/json" -d @api-data.json
```

where:

- <accessToken> is the OAuth access token for your account.
- <hostName> is the URL for your CX Industries Framework API Gateway.
- <workspace> is the path parameter for the production or test workspace. For example, 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

The API returns the response with an accepted status if the descriptor is updated.

Let's consider this example where you're updating the system description for Siebel to add a custom API for getting the product catalog.

Here is a sample request payload for the getCatalog API that you add to the system description by using the systemDescriptors API:

```
{
"target-name": "siebel", "external": {
"apis": [
{
"api-id": "siebel-100",
"api-version": "v1", "api-resources": [
{
"resources": [
{
"resource-id": "res-1"
},
{
"resource-id": "res-2"
},
{
"resource-id": "res-3"
},
{
"resource-id": "res-4"
},
{
"resource-id": "res-5"
},
{
"resource-id": "res-6"
},
{
"resource-id": "res-7"
}
]
}
]
}
]
},
"system" : "siebel", "domain" : "siebel"
}
```

Create Routing Criteria

You can use the existing routing criteria created for the target application to route requests or create a new routing criteria.

Here's how you can create the routing criteria:

1. Create the payload for creating the routing criteria with the required details. For more information, see Routing Criteria.

2. Call the `routingCriteria` API by running this cURL command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
{workspace}/routingCriteria/ -X PATCH BY ID <ID> -H
"Content-Type: application/json" -d @api-data.json
```

   Where:

   - `<accessToken>` is the OAuth access token for your account.
   - `<hostName>` is the URL for your CX Industries Framework API Gateway.
   - `<workspace>` is the path parameter for the production or test workspace. For example, 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.
   - `<ID>` is the identifier of the routing criteria that you are updating.

The API creates the routing criteria and returns its ID. Make note of this ID and enter it as criterion-link in the gatekeeper rule created for the target application.

Routing Criteria

You can use any of these criteria for routing requests using the non-TMF APIs:

Table 5-1 Criteria for Different Routing Requests

Criteria	Description

Boolean	Evaluates whether a condition is true or false. If it's set to true, the application considers that the condition is true. The valid values are true and false.
	Example:
	
	`{`
	`"boolean-criterion": {`
	`"description": "The always passing criterion",`
	`"value": true`
	`}`
	`}`

Table 5-1 (Cont.) Criteria for Different Routing Requests

Criteria	Description

HTTP Header	Evaluates inbound request headers based on the specified criteria. You can use values or regular expressions in this criteria.
	Note: You can't use both values and regular expressions in the same criteria.
	Here are the fields you can use in this criteria:
	• `header-criteria`: The type of the criteria. If you add multiple objects in the same header criteria, all of them must evaluate to true.
	• `request-header-name`: The name of the header key.
	• `header-value-one-of`: A list of strings to validate the header values.
	• `ecma262-regex-value`: The JavaScript regular expression to validate the header values.
	• `header-backend-handling`: The header value handling type. If set to `routing-only`, the application doesn't share the header with the target application. If set to `pass-through`, the application passes the header to the target application.
	Example:
	`{`
	`"header-criteria": [`
	`{`
	`"request-header-name": "end-system", "header-value-one-of": [`
	`"siebel", "fa"`
	`],`
	`"header-backend-handling": "routing-only"`
	`},`
	`{`
	`"request-header-name": "test",`
	`"ecma262-regex-value": "^[a-z]{4,10}$", "header-backend-handling": "routing-only"`
	`}`
	`]`
	`}`

Table 5-1 (Cont.) Criteria for Different Routing Requests

Criteria	Description

Query Parameter	Evaluates the inbound HTTP request query parameters based on the specified criteria. You can use values or regular expressions in this criteria.
	Note: You can't use both values and regular expressions in the same criteria.
	Here are the fields you can use in this criteria:
	• `query-param-criteria`: The type of the parameter.
	• `query-parameters`: A list of supported parameters. If you add multiple objects in the same criteria, all the objects must evaluate to true.
	• `query-parameter-name`: The name of the query parameter.
	• `query-parameter-value-one-of`: A list of string values to validate the query parameter values.
	• `ecma262-query-parameter-regex`: The JavaScript regular expression to validate the query parameter values.
	• `parameter-nature`: The query parameter handling type. If set to `routing-only`, the application doesn't share the header with the target application. If set to `pass-through`, the application passes the header to the target application.
	Example:
	`{`
	`"query-param-criteria": { "description": "Model1",`
	`"query-parameters": [`
	`{`
	`"query-parameter-name": "target-system", "query-parameter-value-one-of": [ "siebel",`
	`"fa"`
	`],`
	`"parameter-nature": "pass-through"`
	`},`
	`{`
	`"query-parameter-name": "test",`
	`"ecma262-query-parameter-regex": "^[a-z]{4,10}$",`
	`"parameter-nature": "routing-only"`
	`}`
	`]`
	`}`
	`}`

Table 5-1 (Cont.) Criteria for Different Routing Requests

Criteria	Description
HTTP Method	Evaluates inbound HTTP request method based on the methods specified in the criteria. The valid values are HTTP methods, such as GET, POST, PUT, and DELETE. Example: `{ "http-method-criteria": { "description": "description", "http-methods": [ "GET", "POST" ] } }`
Expression	Evaluates request based on the expressions specified in the criteria. For information on the expressions, operators, and keywords that you can use in this criteria, see the Gatekeeper Criteria topic in this guide. Example: `{ "expression-criteria": [ { "expression-string": "target-system pr and target-system eq buying" }, { "expression-string": "OP eq GET" } ] }`

You can group these criteria using the `any-criteria-of` condition. When you group the criteria, the API routes the request only if at least one of the criteria evaluates to true. For example:

```
{
"any-criteria-of": [
{
"http-method-criteria": { "description": "description", "http-methods": [
"GET", "POST"
]
},
"query-param-criteria": { "query-parameters": [
{
"query-parameter-name": "target-system", "query-parameter-value-one-of": [


"siebel", "fa"
],
"parameter-nature": "pass-through"
```

```
}
]
}
},
{
"http-method-criteria": { "http-methods": [
"PUT", "DELETE"
]
}
}
] }
}
],
"criteria-link-key": "fa criteria"
}
```

In this example, the API routes the requests only if one of the following conditions evaluates to true:

- HTTP Method of the input request is either GET or POST and `target-system` is present with either `siebel` or `fa` as its value.
- HTTP Method of the input request is either PUT or DELETE.

> **Note**
>
> When you use the `any-of-criteria` condition, you must specify the `criteria-link-key` in the payload. Ensure that it is unique and readable. You can use this link key to reuse the criteria.

You can reuse an existing criteria and group it with any new criteria by using `ref-criterion`. Here's an example with the reference criterion:

```
{
"any-criteria-of": [
{
"ref-criterion": { "criteria-refs": [ "Buying Query Params",
"Buying Criteria Expression"
]
}
}
],
"criteria-link-key": "Reference Criteria"
}
```

In this example, the API routes the request only if both the criteria `BuyingQuery Params` and `BuyingCriteria Expression` evaluate to true.

Review Gatekeeper Rules

If you want to use this custom API for routing requests or publishing events, you must review and update the target application's gatekeeper rule to include this API.

> **Note**
>
> You can use only APIs that follow TMF Open API structure for publishing events.

If you have added a non-TMF API, you must add the Destination Selection details. This includes API ID, version, resources that belong to this API, and the criteria for publishing events.

Here's an example:

```
"destination-selection": [
{
"api-id": "siebel-200",
"api-version": "v1",
"include-in-sparse-query-results":{
"enabled": true # To specify if the Gatekeeper is interested in Routing
Ambiguity or not , to be defined per api
},
"criteria": [
{
"rank": 14,
"resource-ids": [ "res-0"
]
},
{
"rank": 39,
"resource-ids": [
"res-1",
"res-2",
"res-3",
"res-4",
"res-5"
],
"criterion-link": "UNI1648100823"
}
]
}
]
}
```

If you have only a single gatekeeper and that gatekeeper supports all the resources belonging to the non-TMF API, then you need not define any criteria or specify the resources. You can just add the following:

```
"destination-selection": [ { "api-id": "Siebel-200", "api-version": "v4",
"criteria": [ { "rank": 50 } ] } ]
```

If you have added a TMF API, you can skip the Destination Selection details and update only the gatekeeper rules. To review and update the gatekeeper rule for a custom API, see Update Gatekeeper Rules.

6

Integrate External Applications

How You Integrate External Back-End Applications

Your application may come predefined with integrations to other Oracle cloud services. You can also integrate other applications, such as third-party applications, with the existing applications in your topology.

To integrate with external Back-end applications, use the Industries Framework. For example, to integrate:

- Customer Relationship Management (CRM) software
- Billing software and custom tax engines
- Enterprise Resource Planning (ERP) software

To interact with external front-end applications, set up the front-end applications to call the corresponding TM Forum (TMF) Open APIs per your business needs. For example, you can connect a web store UI to create customer accounts or a Website for customers to manage their own accounts.

You integrate external Back-end applications to route API requests, publish event data, and receive necessary event data from those applications. To integrate an application, you start by adding the system description and provide the necessary details to connect to that application. And, to route requests or publish events to that application, you define routing conditions and events that your external application wants to listen to.

Before you integrate an external Back-end application, do the following:

- If you're using a custom API for this integration, register that API. If you're using TM Forum (TMF) Open APIs, you can skip this step. To add a custom API, see the Set Up Custom APIs chapter.
- Determine the APIs you want to associate with your external application for routing requests and receiving data. You can associate any Open API or your own API with the external application. For example, you can associate the Account Management API with your external customer master to route account-related requests.
- Check which APIs generate the events that you want your external application to listen to. You can use TMF Open APIs or your own API for listening to events. For example, you can register Oracle Communications Billing and Revenue Management to use the Account Management API to listen to account data for managing accounts.
- If the external application doesn't support TMF Open APIs, build an adapter for the TMF Open APIs in the external application. The adapter is responsible for transforming the data shared between the applications into the format that each application understands. This enables your external application to interact with your application. You can build the adapter on a cloud service (for example, Oracle Fusion applications) or an on- premises system (for example, Oracle Siebel applications).

> **Note**
>
> The Oracle cloud services that are preintegrated with your application use TMF adapters to interact. You don't have to build your own adapter for these applications. For example, Care Experience uses the adapter in Oracle Communications Billing and Revenue Management to interact with that application. For integrating Oracle cloud services that support customer, account, or party management, you can use the corresponding TMF adapters.

To integrate an external back-end application, you must do the following:

1. Add the external application that you want to integrate with your application.
2. Set up the workspace connections for your external application.
3. Add the additional routing conditions for context-based routing by using the gatekeeperRules API.

Add the External Application

You use systemDescriptors API to add the external or target application.

> **Note**
>
> Ensure that you don't update default system descriptors to integrate any new external application. To replace a preintegrated target application with an external application, create descriptors by using the POST method of systemDescriptorsAPI. In case, if you lose a default system descriptor, create a service request on My Oracle Support to revert your changes.

Here's how you add your external application:

1. Create a payload with the target application details described in this table.

Field	Description
`target-name`	A unique name of the target application.
`system`	The type of the target application.
`domain`	The domain of the target application.
`apis`	The details of the APIs that connect with the target application to send requests. Ensure that you specify the URL prefix to connect to the APIs. You must specify at least one API and one resource for the target application.

2. Call the systemDescriptors API by running this cURL command or by using a REST API client:

```
curl-H Authorization: Bearer <accessToken> https://<hostName>/admin/
{workspace}/systemDescriptors/ -X POST -H "Content-Type: application/json"
-d @api-data.json
```

Where:

- `<accessToken>` is the OAuth access token for your account.

- <hostName> is the URL for your CX Industries Framework API Gateway.
- <workspace> is the path parameter for the production or test workspace. For example, 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

Here is a sample request payload for defining the external tax engine as the target application by using the systemDescriptors API:

```
{
"target-name":"Tax Engine", "external":{
"offered-apis":[
{


"api-name":"taxCalculation", "api-number":"orcl-200",
"api-version":"v1",
"url-prefix":"cx/industry", "api-resources":[ "CalcTaxes",
"Healthcheck"
]
}
]
},
"system":"Taxation", "domain":"Taxes" "type": "external"
}
```

Now that you have added your external application,add the connections details for each workspace connection.

Set Up Workspace Connection

Here's how you add the workspace connection details for your external application:

> **Note**
>
> Ensure that you don't update default connection descriptors to integrate any new external application. To replace a preintegrated target application with an external application, create connection descriptors by using the POST method of connectionDescriptors API. In case, if you lose a default connection descriptor, create a service request on My Oracle Support to revert your changes.

1. Create a payload with the target application's connection details described in this table.

Field	Description
system-descriptor	The unique identifier of the descriptor defined for the target application.
endpoint-name	The name of the endpoint. This must be unique for each application. The endpoint can be an API or an adapter that acts as a bridge for the integration of the applications.
endpoint-url	The Endpoint URL of the target application.
user-name	The user name for the endpoint. Required for the Basic Auth authentication types.

Field	Description
password	The password for the endpoint. Required for the Basic Auth authentication types.
identity-uri	The URL from which your communications application retrieves an authorization token. Required for OAuth authentication type.
client-id	The client ID your authorization server assigns to your communications application. Required for OAuth authentication type.
client-secret	The client secret assigned to your communications application and used to retrieve an authorization token. Required for OAuth authentication type. The OAuth scope that your communications application requests access to and that's granted with the token. Appears for OAuth authentication type.
scope	The OAuth scope that your communications application requests access to and that's granted with the token. Appears for OAuth authentication type.
code	Specifies the connection type. For integrating external applications, the connection type must be external.
server-audience	Leave this blank if you're using Oracle Identity Cloud Service as the identity provider. If you're using an external identity provider, create a service request on My Oracle Support at https://support.oracle.com for updating this value for your identity provider.

2. Call the connectionDescriptors API by running this cURL command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
{workspace}/connectionDescriptors/
-X POST -H "Content-Type: application/json" -d @api-data.json
```

Where:

- <accessToken> is the OAuth access token for your account.
- <hostName> is the URL for your CX Industries Framework API Gateway.
- <workspace> is the path parameter for the production or test workspace. For example, 02 for the test workspaceand 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

The API returns the response with the application descriptor ID once configured.

Here is a sample request payload for providing the external tax engine's connection details by using the connectionDescriptors API:

```
"routing":
{
"system-descriptor": "SYSTEM_DESCRIPTOR_ID", "endpoint-name": "taxation-
rest",
"endpoint-url": "http://TAXATIONGW_IP:PORT", "fabric-facing-auth": {
"oidc-client-credentials": { "client-id": "OIDC_CLIENT_ID",
```

```
"client-secret": "OIDC_CLIENT_SECRET", "identity-uri": "https://
IDENTITY_URI", "scope": "OIDC_SCOPE"
}
},
"type": "external"
}
```

If you're using OAuth for authorizing access to APIs, provide the OAuth client credentials as specified in the sample payload. Instead, if you're using HTTP basic authentication, provide the basic credentials, such as:

```
"basic":
{
"username": "admin",
"password": "admin"
}
```

Now that you have added the connection details, update the routing map to connect with the external application to send requests.

7

Set Up Listeners for Events

Set Up Event Listeners Using API

Use CX Industries Framework REST APIs to publish data from events to external applications. You publish event data to automatically notify external applications when specific events occur in your communications application. For example, when a customer purchases a product, your application sends the event data to your Customer Relationship Management (CRM) software. You can also use this data in your external analytics and reporting applications.

You must specify which events to publish to each external application. For example, you can configure your communications application to publish events when

- A customer's account is created
- A customer purchases a product
- A customer's invoice or bill is created
- A service is added to a customer's account
- A product catalog is published
- Taxes to be calculated for a product by a custom tax engine.

To do so, set the external applications as event listeners. You can also use the corresponding TMF API's hub instance to set up a listener.

For more information on TM Forum Open APIs and CX Industries Framework APIs, see Rest API for CX Industries Framework.

Before you begin, remember to check the following:

- Check which APIs generate the events you want to listen to. You can use TM Forum (TMF) Open APIs or your own API for listening to events. For example, you can register Oracle Communications Billing and Revenue Management to use the Account Management API to listen to account data for managing accounts.
- Check if the listener supports TMF Open APIs. If the listener doesn't support these APIs, build an adapter for the TMF Open APIs in the listener application. The adapter is responsible for transforming the data shared between the applications into the format that each application understands. This enables the listeners to connect with your communications application, such as Launch Experience and Care Experience. You can build the adapter on a cloud service (for example, Oracle Fusion applications) or an on-premises system (for example, Oracle Siebel applications).

Here's how you can set up the event listeners using API:

1. Create the payload for setting up the event listener with the details described in this table.

Field	Description
`api-name`	The name of the TMF Open API or the custom API that publishes the event data.
`api-number`	API Number The API number.

Field	Description
api-version	API Version The API version number.
query	The criteria used for determining the events that must be published to this listener. For setting the criteria, see the Gatekeeper Criteria topic in the Set Up Gatekeepers chapter.
listener-url	The endpoint URL of the target application.
identity-uri	The URI from which your communications application retrieves an authorization token. Required for the OAuth authentication.
client-id	The client ID your authorization server assigns to your communications application. Required for the OAuth authentication.
client-secret	client Secret The client secret assigned to your communications application and used to retrieve an authorization token. Required for the OAuth authentication.
scope	The OAuth scope that your communications application requests access to and that's granted with the token. Appears for the OAuth authentication.
type	The listener type. For external applications to listen to events, the connection type must be external.

Here is the sample payload for setting up a custom tax engine as an event listener using the listenerRegistration API:

```
{
"affiliated-apis": [
{
"api-name": "taxCalculation",
"api-number": orcl-200,
"api-version": "v1",
"query": "(/eventType eq "/event/TaxEvent3" or /eventType eq "/event/
TaxEvent1" and "/event/TaxEvent3/
familyName" eq \"abc\")"
}
],
"listener-locality": {
"external-listener": {
"listener-url": "http://TAXATIONGW_IP:PORT",
"listener-auth": {
"oidc-client-credentials": { "identity-uri": "https://IDENTITY_URI",
"client-id": "OIDC_CLIENT_ID", "client-secret": "OIDC_CLIENT_SECRET",
"scope":
"OIDC_SCOPE"
}
}
}
},
"type": "external"
}
```

2. Call the listenerRegistrations API by using the cURL command or by using a REST API client:
```
curl -H Authorization: Bearer <accessToken https://<hostName>/admin/
<workspace/listenerRegistrations/ -X POST -H "Content-Type: application/json"
-d @lr-data.json
```
Where:
   - `<accessToken>` is the OAuth access token for your account.
   - `<hostName>` is the URL for your CX Industries Framework API Gateway.
   - `<workspace>` is the path parameter for the production or test workspace. For example, 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

   The API returns the response with the listener ID once added.

Set Up Event Listeners Using Hub

Here's how you can set up the event listeners in the respective TM Forum API's hub:

1. Create a payload for setting up the event listener with the details described in this table.

Table 7-1 Payload for Setting Up Event Listeners

Field	Description
callback	The endpoint URL of the listener.
query	The criteria used for determining the events that must be published to this listener. For setting the criteria, see the Gatekeeper Criteria topic in the Set Up Gatekeepers chapter.

Here is the sample payload for setting up a custom tax engine as an event listener by using a TMF API's Hub:

```
{
"callback": "http://TAXATIONGW_IP:PORT",
"query": "(/eventType eq "/event/TaxEvent3" or /eventType eq "/event/
TaxEvent1" and "/event/TaxEvent3/ familyName" eq \"abc\")"
}
```

Here is the sample response payload with the listener ID:

```
{
"id": "listener-registration9tx8k", "callback": "http://
TAXATIONGW_IP:PORT",
"query": "(/eventType eq "/event/TaxEvent3" or /eventType eq "/event/
TaxEvent1" and "/event/TaxEvent3/ familyName" eq \"abc\")"
}
```

2. Call the corresponding TM Forum API by running this cURL command or by using a REST API client:
```
curl -H Authorization: Bearer <accessToken> https://<hostName>/api/
<pathParameter>/<apiName>/ <apiVersion>/hub -X POST -H "Content-Type:
application/json" -d @lr-data.json
```

where:

- <accessToken> is the OAuth access token for your account or the HTTP basic authentication.
- <hostName> is the URL for your API Gateway.
- <pathParameter> is the path parameter for the production or test workspace, such as, 01 for the test workspace and 02 for the production workspace.
- <apiName> is the name of the TMF Open API or the custom API that publishes the event data.
- <apiVersion> is the version of the API.

The API returns the response with the listener ID if the listener is added.

3. Specify the authentication credentials for authorizing access to the listener endpoint by doing the following:

   a. Create a payload with the OAuth or basic authentication credentials described in this table.

Table 7-2 Creating a Payload with OAuth

Field	Description
User name	The user name for accessing the endpoint. Required for the HTTP basic authentication.
Password	The password for accessing the endpoint. Required for the HTTP basic authentication.
Identity URI	The URI from which your application retrieves an authorization token. Required for the OAuth authentication
Client ID	The client ID your authorization server assigns to your application. Required for the OAuth authentication.
Client Secret	The client secret assigned to your application and used to retrieve an authorization token. Required for the OAuth authentication.
Scope	The OAuth scope that your application requests access to and that's granted with the token. Appears for the OAuth authentication.

   b. Call the listener Registrations API by using the cURL command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
<pathParameter>/listenerRegistrations/<id> -X POST -H "Content-Type:
application/json" -d @lr-data.json
```

   where:

   - <accessToken> is the OAuth access token for your account or the HTTP basic authentication.
   - <hostName> is the URL for your API Gateway.
   - <pathParameter> is the path parameter for the production or test workspace, such as, 01 for the test workspace and 02 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

- `<id>` is the unique identifier of the listener.

Here is the sample payload with the OAuth credentials:

```
[
{
"op": "replace",
"path": "listenerRegistrations/listener-locality/external-listener/
listener-auth", "value": {
"oidc-client-credentials": {
"identity-uri": "https://IDENTITY_URI", "client-id": "OIDC_CLIENT_ID",
"client-secret": "OIDC_CLIENT_SECRET", "scope": "OIDC_SCOPE"
}
}
}
]
```

Here is the sample payload with the HTTP basic authentication credentials:

```
[
{
"op": "replace",
"path": "listenerRegistrations/listener-locality/external-listener/
listener-auth", "value": {
"basic": { "username": "admin", "password": "admin"
}
}
}
]
```

Related Topics

- Update Gatekeeper Rules
- How You Integrate External Back- End Applications

8

Set Up Gatekeepers

Set Up Gatekeeper Rules

Here's why you must set up and how you can set up gatekeeper rules in this application.

You set up gatekeeper rules to route requests as well as to filter and publish events specific to given API resources. Setting up gatekeeper rules also ensures that only the events generated by gatekeepers are published to event listeners. These events aren't republished to the gatekeeper which generated them. This avoids duplication of events in the application. For example, if you use Oracle Customer Data Management as the customer master and an Oracle Siebel application for account management, Customer Data Management is set as the gatekeeper. In this scenario, the TMF FA Adapter publishes the account event generated by Customer Data Management to Care Experience. This can lead to creation of the same account in the Siebel application but that application can't republish the account event to Customer Data Management.

By default, an internal application is set as the gatekeeper. You can also set an external application as a gatekeeper. To set the application as a gatekeeper, you must update the gatekeeper rule generated for that application by using the Gatekeeping Rules API. If your new gatekeeper can publish events only for some resources, the existing gatekeeper coexists with the new gatekeeper and continues to publish events for remaining resources.

You can assign multiple gatekeepers for a specific combination of API name, API version, and API resource. If you use multiple applications for managing account or customer data, you must assign one of those applications as the gatekeeper. For example, for the Customer Management API, you can set the Oracle Fusion application to publish events for resources R1 and R2 and your Buying Experience application to publish events for resources R3 and R4.

When multiple gatekeepers are defined for a specific combination of API name, API version, and API resource, the application applies the gatekeeper rules based on the criteria and rank defined in those rules.

How to View Gatekeeper Rules

An empty gatekeeper rule is automatically generated when you integrate your application. The associated API uses this rule to publish the events to event listeners.

> **Note**
>
> You can't create or delete these rules. You can only update them. These rules are generated based on the details that you provided during integration.

To view the gatekeeper rule for a specific application, call the gatekeeping Rules API by running this cURL command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
gatekeepingRules/{id} -X GET
```

The API returns the gatekeeper rule generated for the specified ID.

To view all the gatekeeper rules, call the gatekeeping Rules API by running this cURL command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
gatekeepingRules/ -X GET
```

The API returns a list of gatekeeper rules generated for all the applications.

Update Gatekeeper Rules

Here's how you can update gatekeeper rules:

1. Create a payload with the details described in this table. You can create this payload by copying the existing gatekeeper rule generated for the application.

> **Note**
>
> - The application automatically populates the Endpoint Name and Rule Name fields with the values from the system and connection details you provided.
> - For internal applications, you can only update the API resources. For external applications, you can update all the fields as applicable. However, if you make any changes in this rule, ensure that you update the corresponding system and connection details. To update system or connection details, see the Integrate External Applications chapter in this guide.
> - If you are using a custom non-TMF API in a gatekeeper rule, you must provide the Destination Selection details.

Table 8-1 Creating Payload

Field	Description
Rule Name	The name of the rule. This must be unique for each application.
Endpoint Name	The name of the endpoint API or adapter. You can't update this field.
External Event Emitter Identification	The client ID your authorization server assigns to your application. Required for OAuth authentication type. Applicable only for external application.
API ID	The unique identifier of the API. You can add new APIs but you can't delete existing APIs.
API Name	The name of the associated APIs.

Table 8-1 (Cont.) Creating Payload

Field	Description
API Version	The API version number.
API Resources	The resources for which you want to publish events; for example, individual, party, or organization. You must add at least one criteria and one rank for a resource.
Criteria	The criteria is a set of conditions used for filtering the events you want to publish to listeners. For more information, see the Gatekeeper Criteria topic in this chapter. You can use * to indicate that there is no criteria for this resource.
Criteria	The order in which you want to apply the criteria. This field is optional. The minimum value is 0. The rank must be different for the same combination of criteria and resource.
Listener Registration Refs	The list event listener IDs. You can update this only for external applications.
Destination Selection	The details of the Open API and the resources, criteria, and rank you want to use with this API. Applicable only for non-TMF APIs. For more information, see the Review Gatekeeper Rules topic in this guide. **Note:** You can use the non-TMF APIs only to route requests.

2. Call the gatekeeping Rules API by running the following curl command or by using a REST API client:

```
curl -H Authorization: Bearer <accessToken> https://<hostName>/admin/
<workspace>/gatekeepingRules/{id} -
X PATCH -H "Content-Type: application/json" -d @gkr-data.json
```

Where:

- `<accessToken>` is the OAuth access token for your account.
- `<hostName>` is the URL for your API Gateway.
- `<workspace>` is the path parameter for the production or test workspace, which is 02 for the test workspace and 01 for the production workspace. If you are calling the API using FA API Gateway, you can skip this parameter.

The API returns the response with an accepted status if the rule is updated.

Here's the sample gatekeeper rule to publish events belonging to individual and organization resources to listeners using a TMF API. The application publishes the events only if the parameters specified in the criteria are present in the event payload:

```
{
"endpoint-name": "tmf632",
```

```
"rule-name": "Generated gatekeeping rule for endpoint tmf632", "gatekeeping-
apis": [
{
"api-id": "tmf-632",
"api-name": "Party",
"api-version": "v4", "api-resources": [
{
"resource-name": "Individual", "gatekeeping-criteria": {
"criteria": "(ID pr or /event/individual/familyName pr or /event/
individual/id pr or /event/organization/ tradingName pr or /event/
organization/id pr)",
"rank": 1
}
},
{
"resource-name": "Organization", "gatekeeping-criteria": { "criteria": "*",
"rank": 1
}
}
]
}
]
}
```

Gatekeeper Criteria

You specify criteria in the gatekeeper rules to filter the events for publishing. Your criteria can include:

- Event type: The events you want to publish to the event listeners, for example, CustomerCreateEvent.
- Attribute: The JSON path to the parameters in the request payload and HTTP header of the request or reserved keywords used as identifiers. The application uses the value of this attribute for comparing the conditions. You can use only the parameters and keywords listed in the Operators, Parameters, and Keywords topic to define the criteria.
- Operator: The operator for comparing the conditions. You can use only the operators listed in the Operators, Parameters, and Keywords topic to define the criteria.
- Value: The actual value to be compared. This condition isn't applicable if the operator is pr.

A valid criteria must contain at least one condition. If you have multiple conditions, use these logical operators:

- and: Indicates that both the conditions are applicable.
- or: Indicates only one of the conditions is applicable.
- not: Indicates that the condition isn't applicable.

You can also use parentheses () to indicate the condition that takes precedence over other conditions.

Your criteria can be a combination of the following:

- Event types and conditions: Use this to publish specified events when the specified conditions are met. For example, to publish customer creation and deletion-specific events if the customer ID is greater than 10000, set this criteria:

  ```
  ((/eventType eq "CustomerCreateEvent" or /eventType eq
  "CustomerDeleteEvent") and "/event/customer/id" gt 10000)
  ```

- Only conditions: Use this to publish all the generated events when the specified conditions are met. For example, to publish all the relevant events if the customer ID is greater than 10000, set this criteria:

  ```
  ""/event/customer/id" gt 10000"
  ```

- Only event types: Use this to publish only a subset of events. For example, to publish only customer creation and deletion-specific events, set this criteria:

  ```
  "(/eventType eq "CustomerCreateEvent" or /eventType eq
  "CustomerDeleteEvent")"
  ```

Operators, Parameters, Reserved Keywords

Here are the operators that you can use in your criteria.

Table 8-2 Operators

Operator	Description	Applicable to
eq	Equal to	All conditions
ne	Not equal to	All conditions
co	Contains	Strings only
sw	Start with	Strings only
ew	End with	Strings only
pr	Present(represents actual value)	All conditions
gt	Greater than	All conditions
ge	Greater than or equal to	All conditions
lt	Less than	All conditions
le	Less than or equal to	All conditions

Here are the request payload parameters that you can use in your criteria.

Table 8-3 Request Payload Parameters

Parameters	Description
id	The unique identifier of the event. For example, CDRM_1.
href	The reference to the event. For example,"/cx/industry/party/v4/individual/CDRM_123"
familyName	The family name of the event. For example, "abc".
fullName	The full name of the event. For example, "New Party".

Table 8-3 (Cont.) Request Payload Parameters

Parameters	Description
givenName	The given name of the event. For example, "New'
status	The status of the event. For example, "active".
type	The type of the event. For example, "individual".

Here are the reserved keywords that you can use in your criteria.

Table 8-4 Reserved Keywords

Keywords	Description
ID	The URL path parameter IDs used in the HTTP Header. For example, to use only URL path parameter IDs that start with "NEW-" in your criteria, specify the condition as `ID sw "NEW- "`.
OP	The HTTP method used for an operation. For example, to use all the GET operations irrespective of the ID in your criteria, specify the condition as `OP eq "GET" and not (ID pr)`. **Note:** With notification and delivery APIs, you can use only the POST operation method.
MS	The endpoint types, such as: • RMS: for API-specific endpoints • NMS: for event-specific endpoints For example, to use notification APIs and specific values of an incoming event type and an account type in your criteria, specify the condition as: **`MS eq "NMS" and /eventType eq "financialAccountCreateEvent" and /payload/financialAccount/accountType eq "business"`**

Related Topics

- How You Integrate External Back-End Applications
- Review Gatekeeper Rules