Oracle® Banking Microservices Architecture

Routing Hub Configuration User Guide

Release 14.6.0.0.0
F59277-01
May 2022

Oracle Banking Microservices Architecture Routing Hub Configuration User Guide, Release 14.6.0.0.0

F59277-01

Contents

Preface

Purpose

This guide enables the user to integrate Oracle Products with External Product Processor through Oracle Banking Routing Hub Platform.

Audience

This guide is intended for the customers and partners.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
JSON	Java Script Object Notation
XML	Extensible Markup Language
WSDL	Web Services Description Language

List of Topics

This guide is organized as follows:

Table 2 List of Topics

Topics	Description
Introduction	This topic provides the general information about the guide.
Configuration	This topic provides the information about the Configuration.
Service Consumer	This topic provides the information about the Service Consumer.

Table 2 (Cont.) List of Topics

Topics	Description
Service Providers	This topic provides the information about the Service Providers.
Implementation	This topic provides the information about the Implementation.
Consumer Services	This topic provides the information about the Consumer Services.
Transformation	This topic provides the information about the Transformation.
Routing	This topic provides the information about the Routing.
Request Audit	This topic provides the information about the Audit logs.
Chaining	This topic provides the information about the chaining.
Extensibility	This topic provides the information about the extensibility in Routing Hub.
Audit Purging / Archiving	This topic provides the information about the process for audit purging and archiving.
Multipart Request	This topic provides the information about the multipart request template.
Dashboard	This topic provides the information about the dashboard.
Transformation Type	This topic provides the information about the transformation type.
Oracle Banking Routing Hub Integration Specification	This topic provides the information about the Oracle Banking Routing Hub Integration Specification.
Oracle Banking Routing Hub VM Arguments	This topic provides the information about the Oracle Banking Routing Hub VM Arguments.

Symbols and Icons

This guide has the following list of symbols and icons.

Table 3 Symbols and Icons - Common

Symbol/Icon	Function
	Minimize
	Maximize
	Close
	Perform Search
	Open a list

Table 3 (Cont.) Symbols and Icons - Common

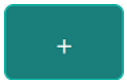
Symbol/Icon	Function
	Add a new record
	Navigate to the first record
	Navigate to the last record
	Navigate to the previous record
	Navigate to the next record
	Grid view
	List view
	Refresh
	Click this icon to add a new row.
	Click this icon to delete a row, which is already added.
	Calendar
	Alerts

Table 4 Symbols and Icons – Audit Details

Symbol/Icon	Function
	A user

Table 4 (Cont.) Symbols and Icons – Audit Details

Symbol/Icon	Function
	Date and time
	Unauthorized or Closed status
	Authorized or Open status

Table 5 Symbols and Icons - Widget

Symbol/Icon	Function
	Open status
	Unauthorized status
	Closed status
	Authorized status

1 Introduction

Oracle Banking Routing Hub is a routing hub that enables seamless & standardized integrations between FSGBU Banking Product using configurations provided as part of the product infrastructure.

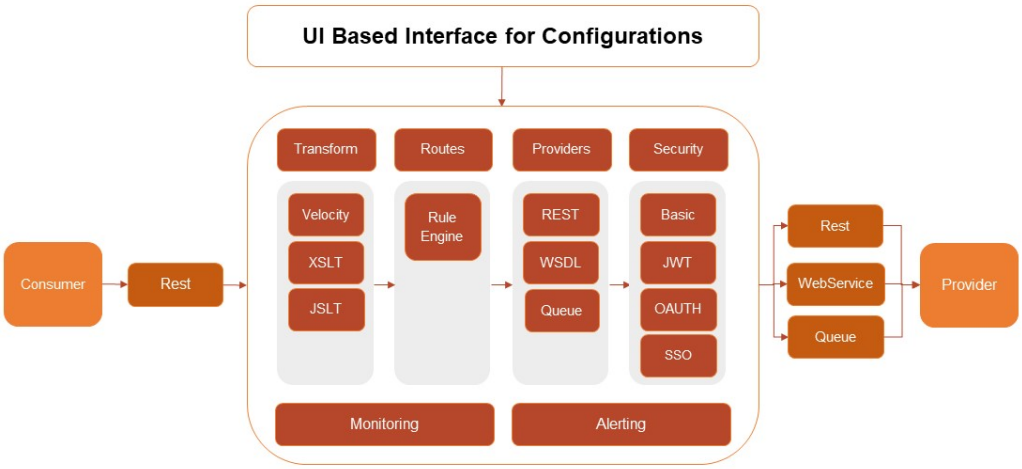
Consumer application does not need to know the points given below.

- Servicing Providers or Product Processors: Product processor to which the integration is required.
- Name of the Service: logical name of the service. For example, Funds Transfer and Letter of Credit Initiation.
- Messaging structure of Service: structure of the message. For example, JSON and XML files.
- Communication Protocol: Web services, Rest API and Queue.
- Can be integrated with different versions of a Product processors.

This guide shows the maintenance of two product as given below.

- Oracle Service Consumer as Service Consumer
- External Product Processor as Service Provider

Figure 1-1 UI Based Interface for Configurations



2 Configuration

This topic describes the systematic instructions to perform the configuration.

The end-user can configure the properties with respect to monitoring and alerting. The end-user can configure the same at system level and granular levels such as consumer, consumer service, and routing.
The **Configuration** screen contains the following sections.

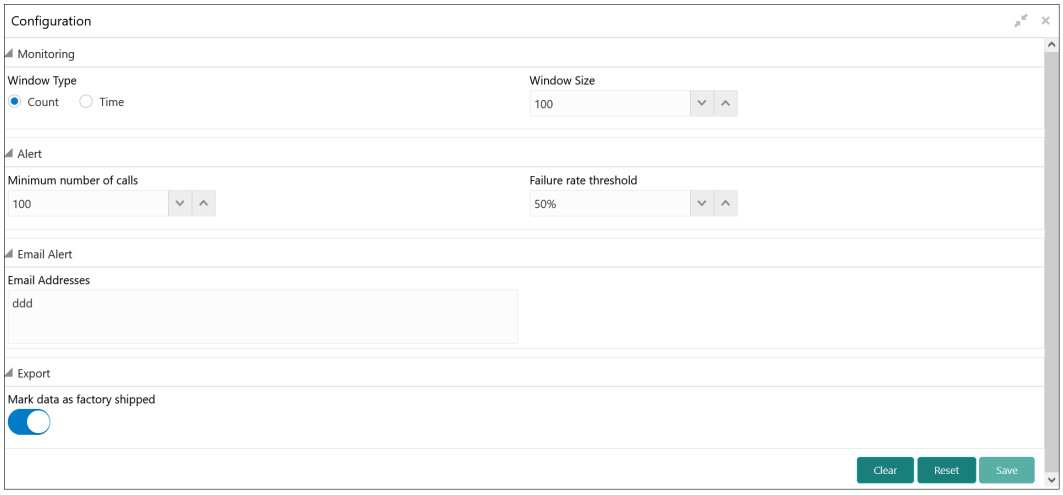
- **Monitoring** - It has the features required by the breaker to store and aggregate the result of calls.
- **Alert** - It has the features required for transitioning circuit breaker.
- **Email Alert** - It has the feature required for mail notification.
- **Export** - It has the properties that are required for exporting the configuration JSON and will be visible at system level only.

Specify **User ID** and **Password**, and login to **Home** screen.

1. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing Hub**.
2. Under **Routing Hub**, click **Configuration**.

   The **Configuration** screen displays.

Figure 2-1 Configuration



3. Specify the fields on **Configuration** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 2-1 Configuration - Field Description

Field	Description
Window Type	Select the type of the window. The available options are: • **Count**: The count-based sliding window aggregates the outcome of the last N calls (**Window Size**). • **Time**: The time-based sliding window aggregates the outcome of the calls of the last N seconds (**Window Size**).
Window Size	Specify the window size to record the outcome of the calls when the circuit breaker is closed. • For **Count** window type, The window size is N calls. • For **Time** window type, The window size has N seconds.
Minimum number of calls	Specify the minimum number of calls. For example: If the minimum number of calls are 10, then at least 10 calls must be recorded before calculating the failure rate. If only nine calls are recorded, the circuit breaker is not transitioned to open even if all nine calls are failed.
Failure rate threshold	Specify the failure rate threshold in percentage. When the failure rate is equal or greater than the threshold, the circuit breaker transitions to open and starts short-circuiting calls.
Email Addresses	Specify the E-mail address. The user can use semi-colon to add more email addresses. Once the failure rate crosses the **Failure rate threshold**, a mail is sent to the end-user about the event.
Mark data as factory shipped	Select the toggle to mark the exported configuration JSON as factory shipped JSON. The end-user will not be able to modify or delete the certain data once imported. By default, the toggle is OFF.

Example:

Table 2-2 Configuration - Field Entry Values

Field	Entry Values
Window Type	Count
Window Size	20
Minimum number of calls	10
Failure rate Threshold	50%

Configured properties will result as below:

After 10 (minimum number of calls) calls, routing would get shutdown if 50% (failure rate) of almost last 20 (window size) calls have failed. If the email address property is configured, then the end-user is notified as well.

4. Click **Clear** to clear all the specified details.

5. Click **Reset** to reset the details.
6. Click **Save** to save all the details.

3
Service Consumers

This topic describes the systematic instructions to configure the service consumers.

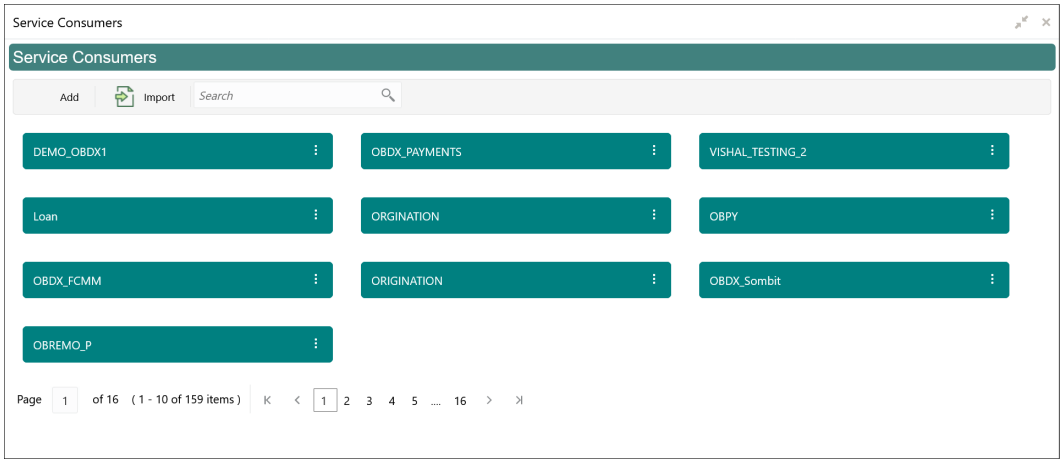
The **Service Consumer** is an Oracle product that invokes Oracle Banking Routing Hub API. Oracle Banking Routing Hub analyses, evaluates the destination product processor, and transforms the data into a format of the same. It comprises the source and destination integration details.

Specify **User ID** and **Password**, and login to **Home** screen.

1. On **Home** screen, click**Core Maintenance**. Under **Core Maintenance**, click **Routing Hub**.
2. Under **Routing Hub**, click **Service Consumers**.

   The **Service Consumers** screen displays.

Figure 3-1 Service Consumers

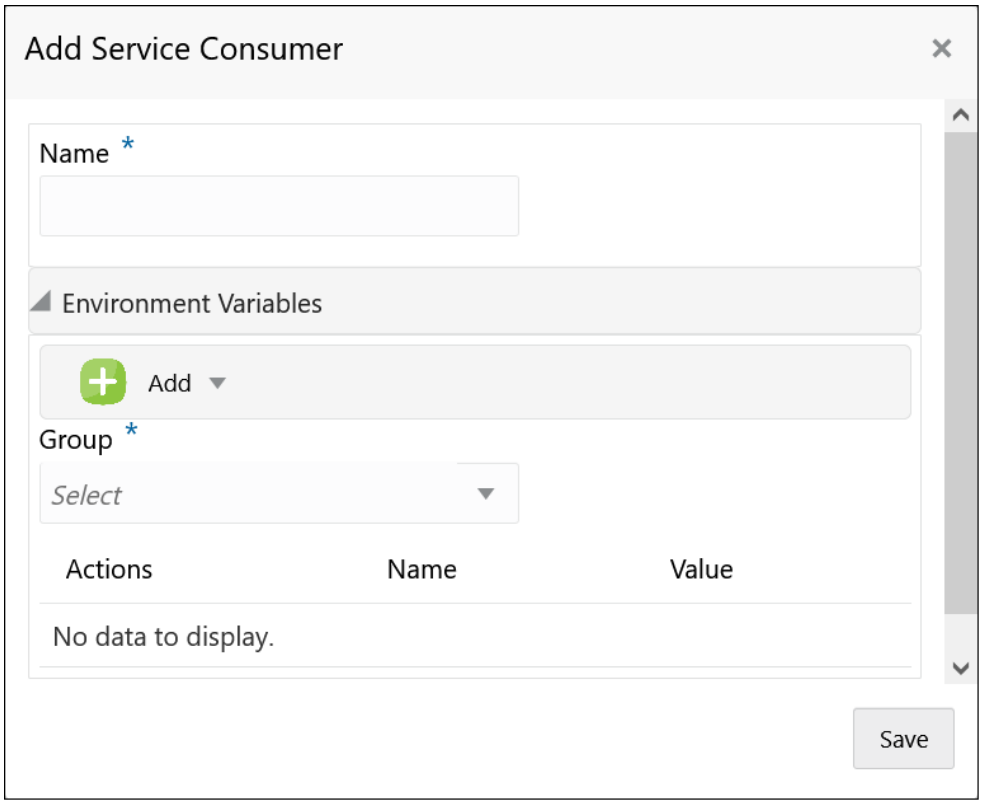


Add Service Consumer
The user can create new Service Consumer manually.

3. Click **Add**.

   The **Add Service Consumer** screen displays.

Figure 3-2 Add Service Consumer



4. Specify the fields on **Add Service Consumer** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 3-1 Add Service Consumer - Field Description

Field	Description
Name	Specify the unique service consumer name. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.

Table 3-1 (Cont.) Add Service Consumer - Field Description

Field	Description
Add	To add, refer to step 5. Select the group from the drop-down list. The available options are: • **Group** • **Variable**
Group	Select the group from the drop-down list.
Action	Displays the action. The user can edit or delete the header.
Name	Displays the name of the header.
Value	Displays the value of the header.

Environment Variables

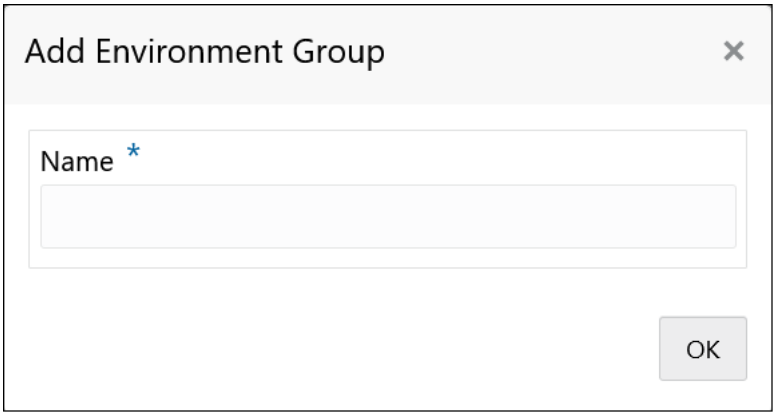
The user must define the group of variables which can be accessed throughout the specific consumer's configuration. The syntax for accessing environment variables is below: $env.Environment_Group_Name.Environment_Variable_Name

For example, $env.COMMON.BRANCH_CODE

5. To add **Environment Variables**, follow the below steps.
   a. Click **Add** on the **Add Service Consumers** screen, and select **Group** from drop-down list to add the group.

      The **Add Environment Group** screen displays.

Figure 3-3 Add Environment Group



   b. Specify the fields on **Add Environment Group** screen and click **OK**.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

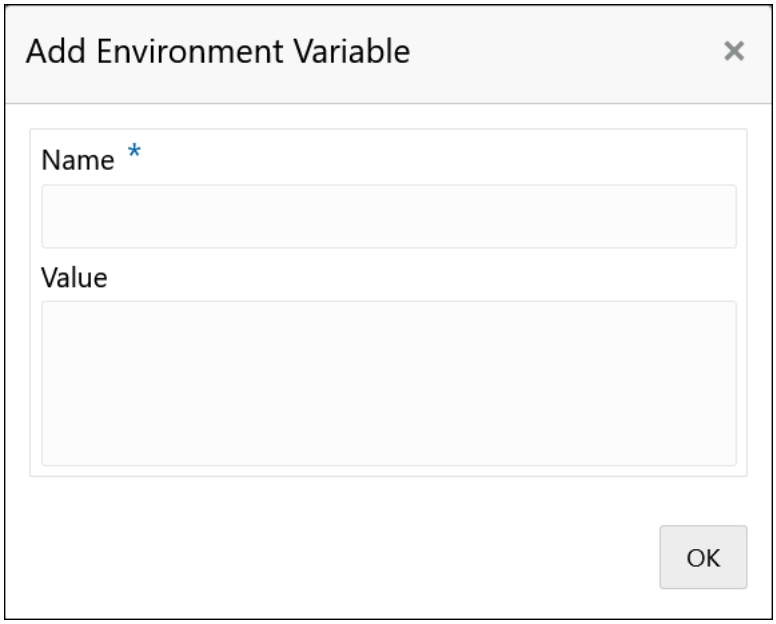
Table 3-2 Add Environment Group - Field Description

Field	Description
Name	Specify the name of the environment group. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.

c. Click **Add** on **Add Service Consumer** screen and select **Variable** from drop-down list to add the variable.

The **Add Environment Variable** screen displays.

Figure 3-4 Add Environment Variable



d. Specify the fields on **Add Environment Variable** screen and click **OK**.

Note:

The fields which are marked with asterisk are mandatory.

For more information on fields, refer to the field description table below.

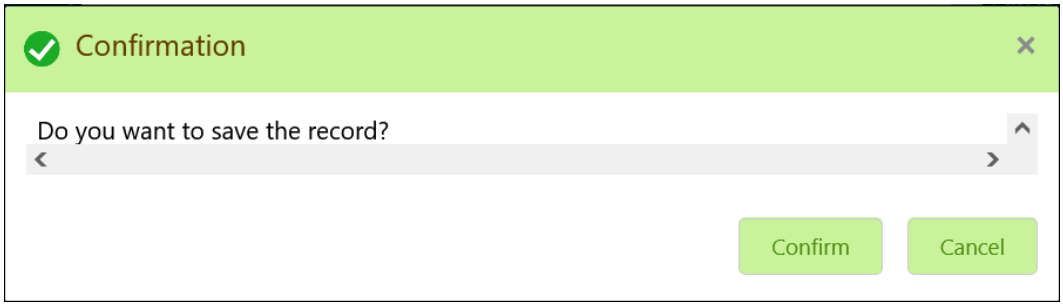
Table 3-3 Add Environment Variable - Field Description

Field	Description
Name	Specify the name of the environment variable. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Value	Specify the value of the environment variable. The value can either be hardcoded or Velocity mapping.

6. Click **Save** to save the details.

   The **Confirmation** screen displays.

Figure 3-5 Confirmation - Add Service Consumers



7. Click **Confirm** to save the record.

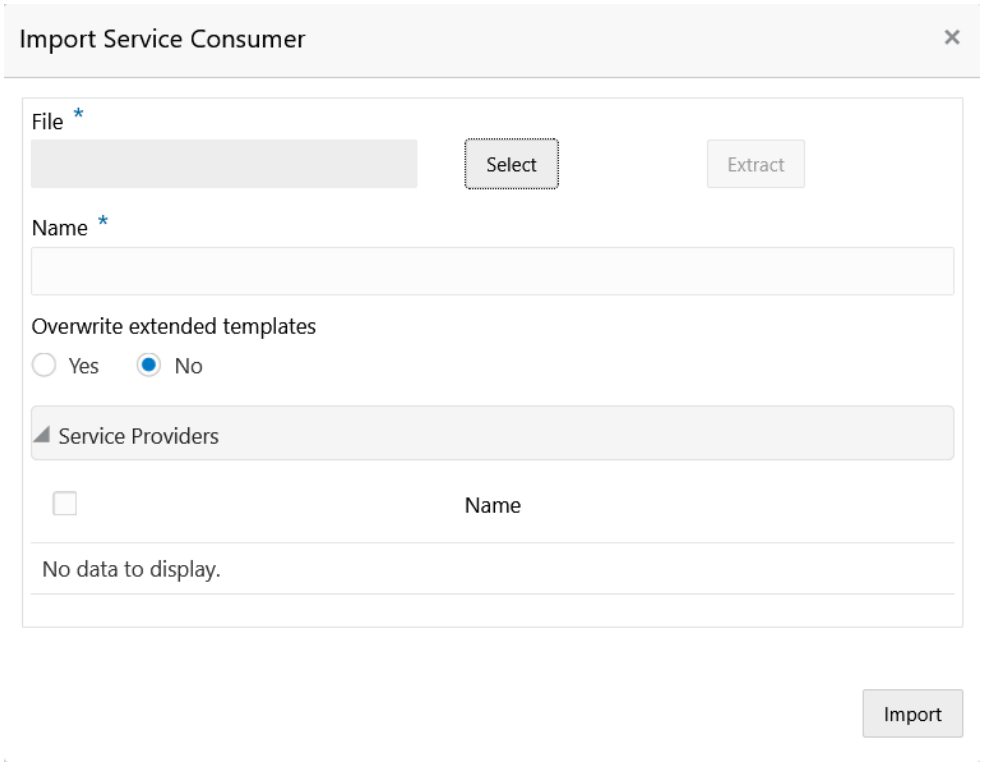
Import Service Consumer

The user can create a service consumer by importing the JSON file and manually selecting the service providers or select all providers that needs to be imported. The user can also import zip file in order to import all the configuration JSON files together.

8. Click **Import**.

   The **Import Service Consumer** screen displays.

Figure 3-6 Import Service Consumer



9. Specify the fields on **Import Service Consumer** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 3-4 Import Service Consumer - Field Description

Field	Description
File	Select the file using **Select**. **Note:** Allows only to select one file and accepts JSON and ZIP file.
Extract	Extracts the consumer name and service provider list from JSON file only and displays it in the respective elements.

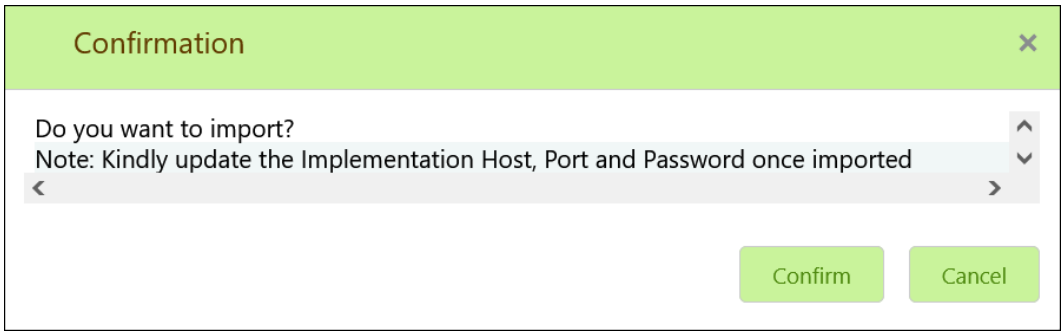
Table 3-4 (Cont.) Import Service Consumer - Field Description

Field	Description
Name	Specify the name of the service provider. **Note:** • Name cannot be blank and required only for JSON file. • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Overwrite extended templates	Select the respective radio button to overwrite the extended templates. The available options are: • **Yes** - This option overwrites the extended templates. • **No** - This option retains the existing extended templates.
Service Providers	Displays the service provider details.
Name	Displays the list of service providers names that are present in JSON file only.

10. Click **Import** to import the selected service consumer file.

    The **Confirmation** screen displays.

Figure 3-7 Confirmation - Import Service Consumer



11. Click **Confirm** to import the file.

> **Note:**
>
> Below data needs to be changed after importing provider configuration file:
>
> • Implementation Host and Port
>
> • Implementation Authentication Password

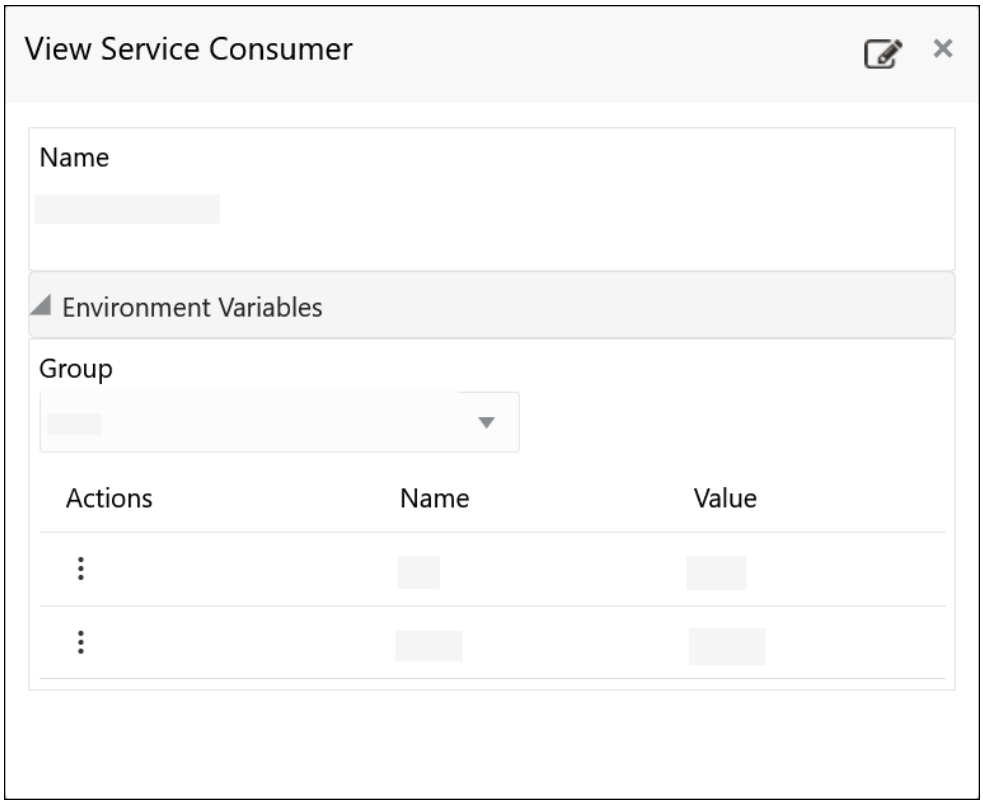
View Service Consumer

The user can view consumer details and can also switch to edit form by clicking on edit icon.

12. On the **Service Consumer** tile, click **Operation Menu** (3 dot icon), and select **View** from the dropdown list.

    The **View Service Consumer** screen displays.

Figure 3-8 View Service Consumer



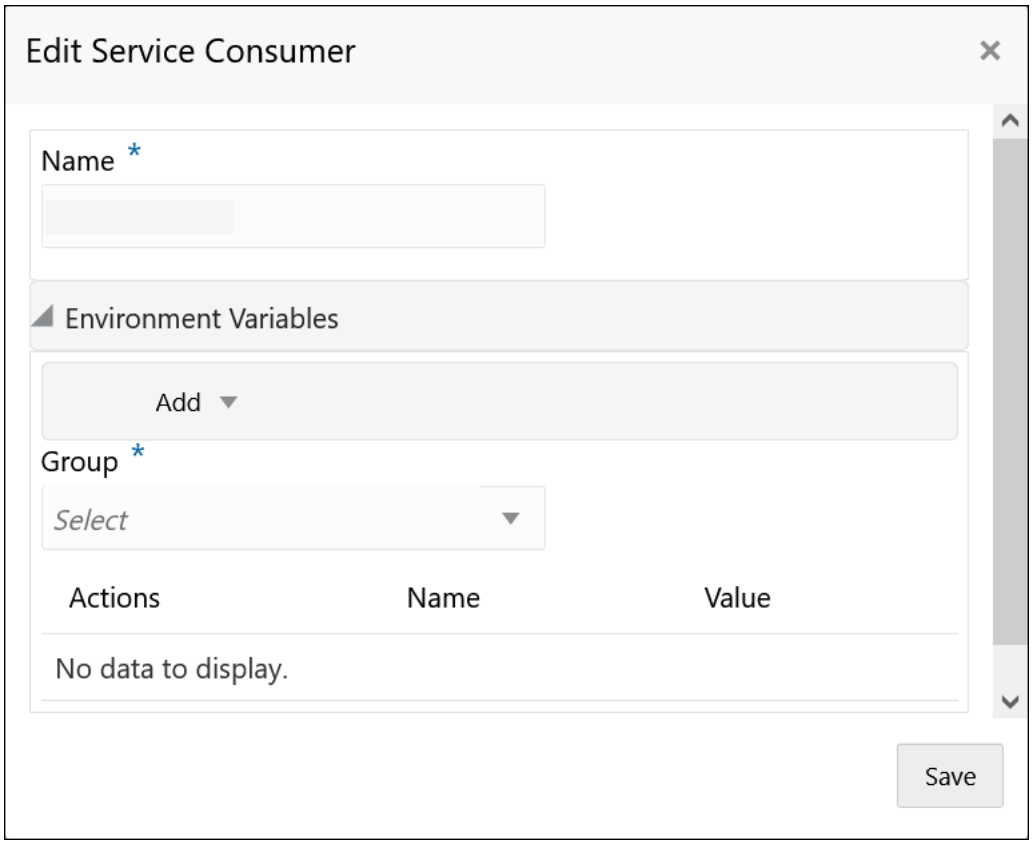
13. Click **Edit** button to edit the Service Consumer.

Edit Service Consumer
The user can modify the consumer details.

14. On the **Service Consumer** tile, click **Operation Menu** (3 dot icon), and select **Edit** from the dropdown list.

    The **Edit Service Consumer** screen displays.

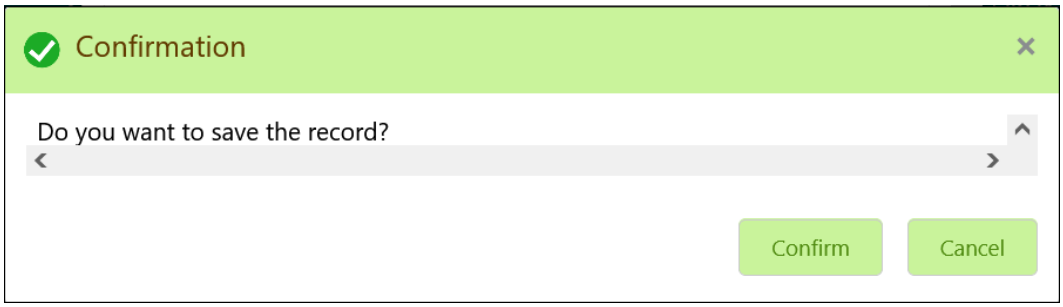
Figure 3-9 Edit Service Consumer



15. Click **Save** to save the modified consumer details.

    The **Confirmation** screen displays.

Figure 3-10 Confirmation - Edit Service Consumer



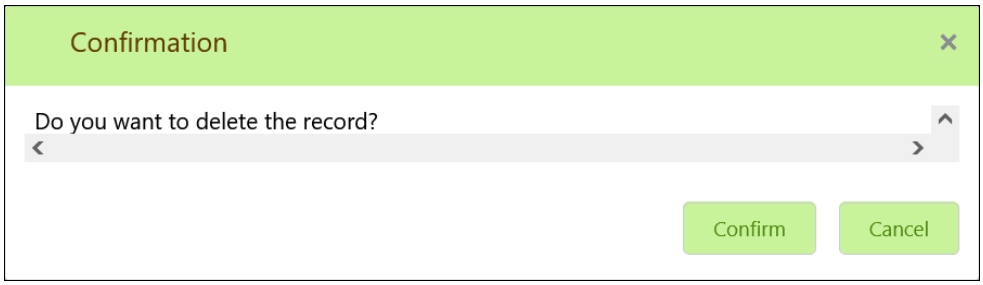
16. Click **Confirm**.

Delete Service Consumer
The user can delete the Service Consumer.

17. On the **Service Consumer** tile, click **Operation Menu** (3 dot icon), and select **Delete** from the dropdown list.

    The **Confirmation** screen displays.

Figure 3-11 Confirmation - Delete



18. Click **Confirm** to delete the service consumer.

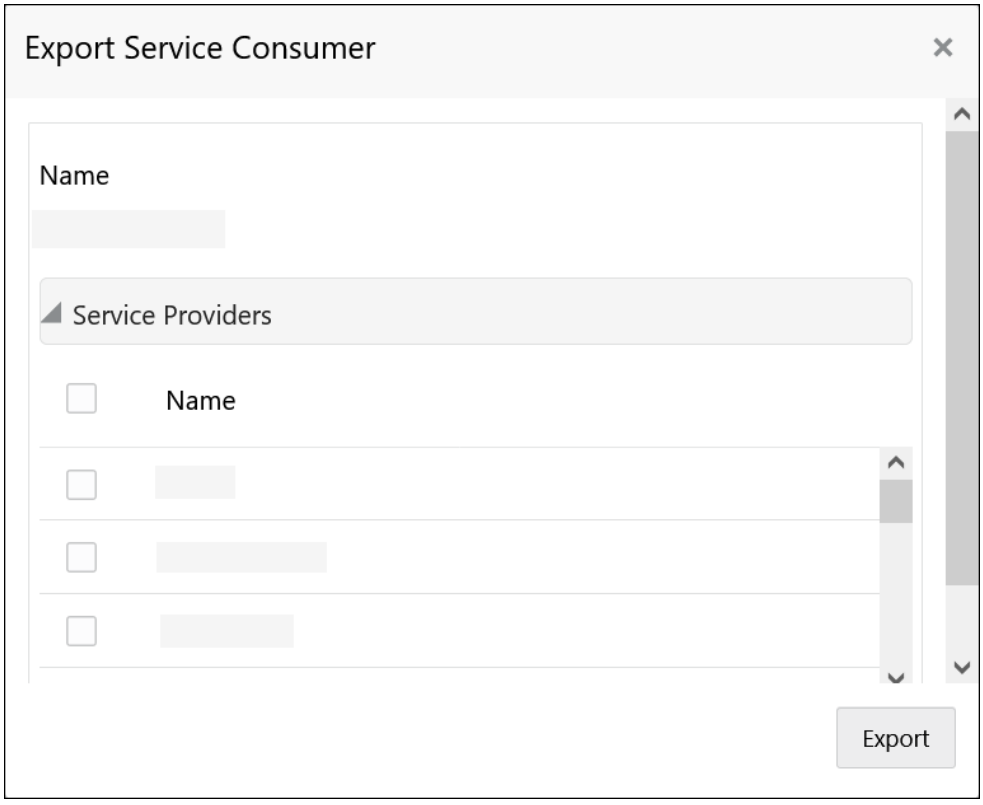
JSON Export

The user can export the consumer configuration as JSON file.

19. On **Service Consumer** tile, click **Operation Menu** (3 dot icon).
20. On **Export** option, select **JSON** from the list.

    The **Export Service Consumer** screen displays.

Figure 3-12 Export Service Consumer



> **Note:**
>
> - The user has an option to select the service providers from the list which needs to be exported or can click on select all for all service providers.
> - The JSON Export feature exports below data:
>   - Selected service consumer
>   - All consumer services
>   - Selected service providers with services
>   - All implementations of selected service providers with services (without Host, Port and Authentication Password)
>   - All transformations
>   - All routes

21. Select the required service providers and click **Export**.

    The **Confirmation** screen displays.

Figure 3-13 Confirmation - JSON Export

22. Click **Confirm** to export the service consumer in JSON file.

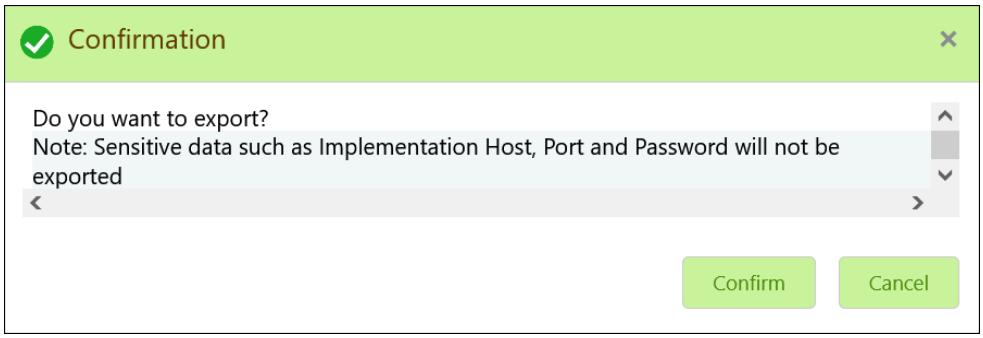
SQL Export

The user can export the consumer configuration as SQL file.

23. On **Service Consumer** tile, click **Operation Menu** (3 dot icon).
24. On **Export** option, select **SQL** from the list.

    The **Confirmation** screen displays.

Figure 3-14 Confirmation



> **Note:**
>
> The SQL Export feature exports entire configuration without Host, Port, and Authentication Password details.

25. Click **Confirm** to export the consumer configuration as SQL file.

Configuration

26. On **Service Consumer** tile, click **Operation Menu** (3 dot icon), and click **Configuration**.

    The **Configuration** screen displays.

> **Note:**
>
> Refer to Configuration topic for the screen and field description.

Request Audit

27. On **Service Consumer** tile, click **Operation Menu** (3 dot icon), and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for the screen and field description.

4
Service Providers

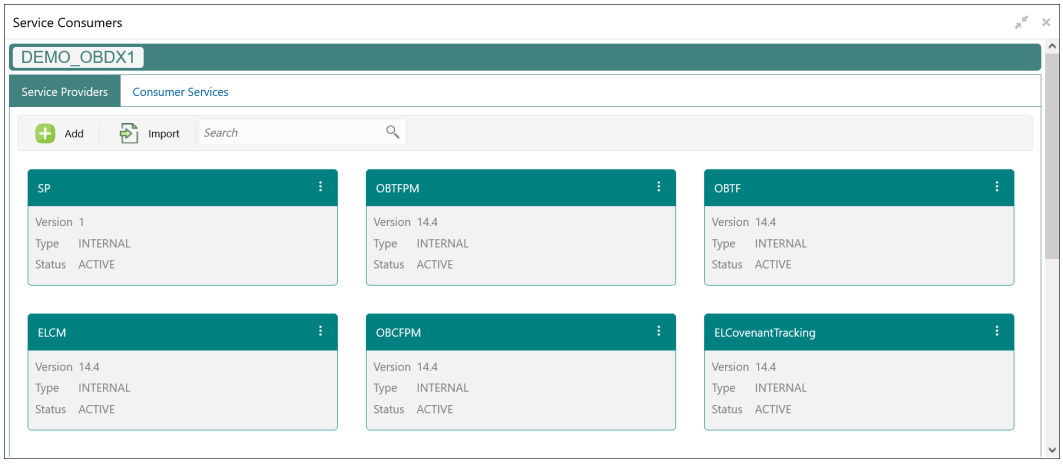
This topic describes the systematic instructions to configure the service providers.

The **Service Providers** are the product processors configure to process request send by Oracle Banking Routing Hub on behalf of service consumers. It comprises destination integration details.

1. On **Service Consumers** screen, click the required service consumer.

   The **Service Providers** screen displays.

Figure 4-1 Service Providers

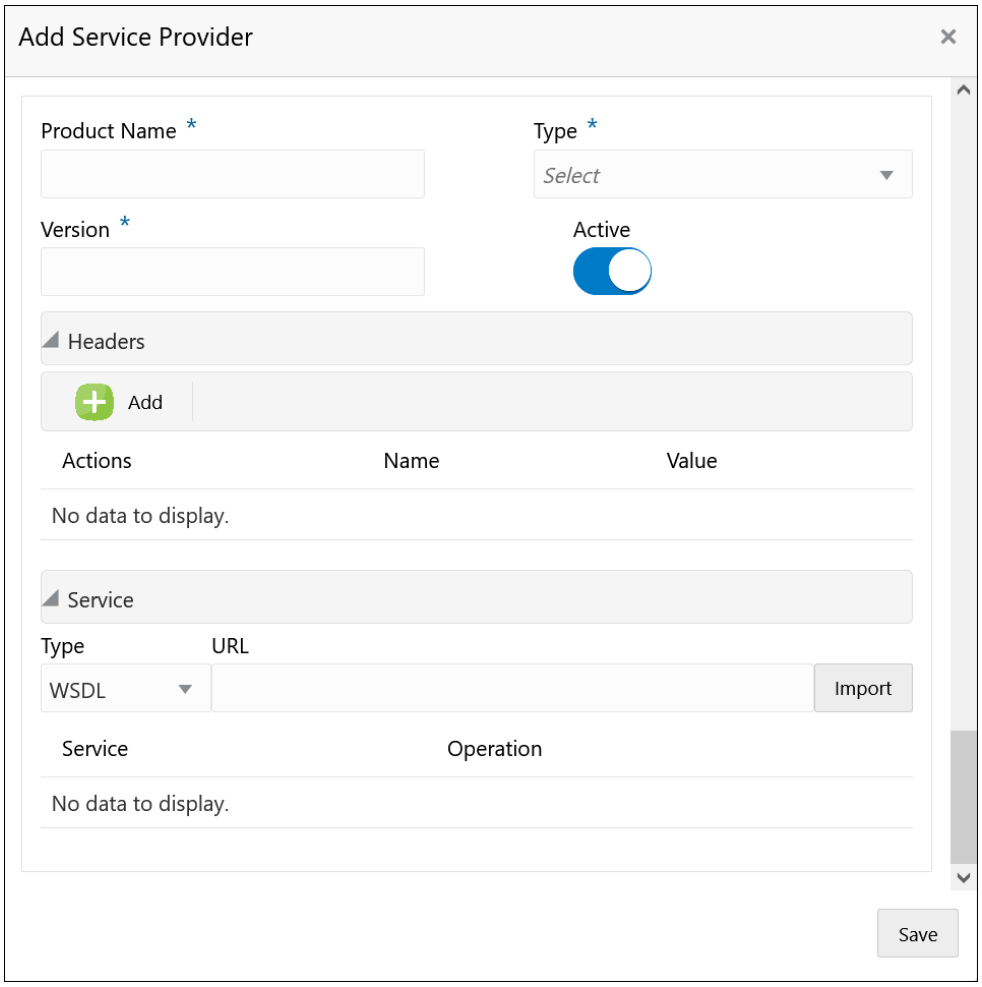


Add Service Provider
The user can create Service Provider manually.

2. Click **Add**.

   The **Add Service Provider** screen displays.

Figure 4-2 Add Service Provider



3. Specify the fields on **Add Service Provider** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 4-1 Add Service Provider - Field Description

Field	Description
Product Name	Specify the product name of the service provider. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Type	Select the type of service provider from drop-down list The available options are: • **INTERNAL**: Used for Oracle products • **EXTERNAL**: Used for non-Oracle products
Version	Specify the provider version. **Note:** • Enter 0 to maximum of 255 characters. • Only numeric or decimal values are allowed.
Active	Predefined values are Active / Inactive If provider is marked as inactive, then all related routes will be stopped.
Add	To add, refer to the below steps.
Actions	Displays the action. The user can edit or delete the header.
Name	Displays the name of the header.
Value	Displays the value of the header.
Type	Select the type of service from drop-down list. The available options are: • **WSDL** • **SWAGGER** • **OTHERS**
Name	Specify the name of the operation. **Note:** This field appears only if the **Type** is selected as **OTHERS**.

Table 4-1 (Cont.) Add Service Provider - Field Description

Field	Description
Http Method	Select the HTTP method. The available options are: • **GET** • **POST** • **PUT** • **PATCH** • **DELETE** **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Endpoint	Specify the endpoint URL for the operation. **Note:** This field appears only if the **Type** is selected as **OTHERS**.
URL	Specify the service URL of the file location. **Note:** This field appears only if the **Type** is selected as **WSDL** and **SWAGGER**.
Import	Click **Import** to extract the service information from URL. **Note:** This field appears only if the **Type** is selected as **WSDL** and **SWAGGER**.
Gateway Prefix	Gateway Prefix is context path of below formatted URL http://host:port/gateway-prefix/endpoint

Table 4-1 (Cont.) Add Service Provider - Field Description

Field	Description
Service Headers	Specify the Endpoint specific headers. Value can either be hardcoded or can be Velocity mapping. **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Service Query Params	Specify the Endpoint specific query parameters. Value can either be hardcoded or can be Velocity mapping. **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Service	Displays the extracted service from the selected URL.
Operation	Displays the extracted operation from the selected URL.

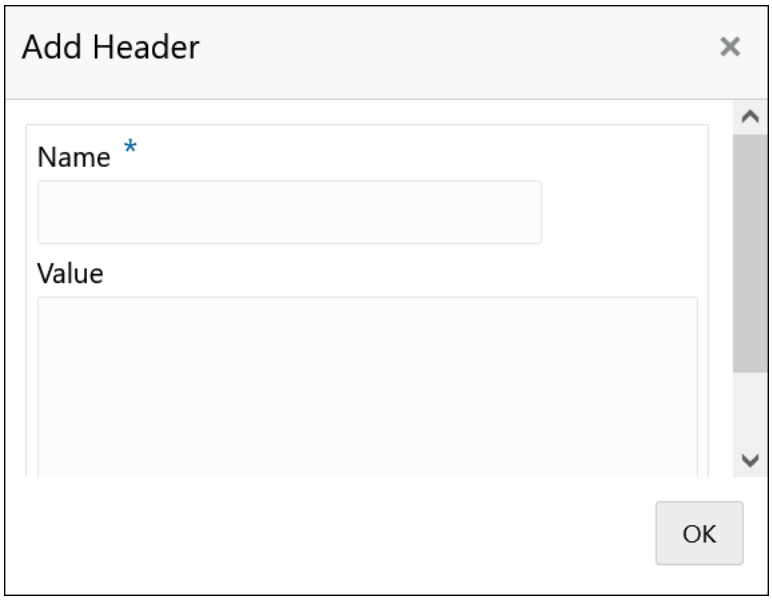
Headers

External product processor might require some standard headers to be passed along with the request. The user can specify the headers which are required by service endpoints for its all implementations but not present in swagger file.

4. To add **Headers**, follow the below steps.
   a. Under **Headers** section, click **Add**.

      The **Add Header** screen displays.

Figure 4-3 Add Header



b. Specify the fields on **Add Header** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 4-2 Add Header - Field Description

Field	Description
Name	Specify the name of the header.
Value	Specify the value of the header.

5. Click **OK** to save the details.

   The **Confirmation** screen displays.

6. Click **Confirm**.

Service

- **WSDL**:
  The Web Services Description Language (WSDL) is an XML-based interface description language that is used for describing the functionality offered by a web service.

  Both SSL and non-SSL WSDL URL are supported.

> **Note:**
>
> If there is a change in wsdl file, then same wsdl file need to be imported again to update the provided service information in Routing Hub.

- **SWAGGER**:
  Swagger is an Interface Description Language for describing RESTful APIs expressed using JSON.

  Currently, Swagger 2.0 & OpenAPI 3.0 both are supported.

  > **Note:**
  >
  > If there is a change in swagger file, then same swagger file need to be imported again in order to update the provided service information in Routing Hub.

- **OTHERS**:
  OTHERS option is selected for adding REST API details manually when provider does not have swagger file.

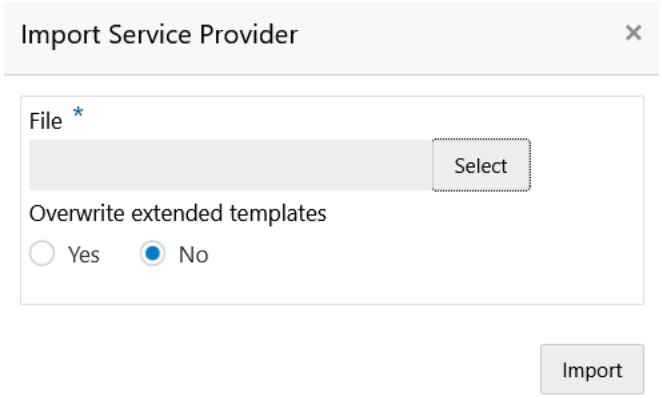
  > **Note:**
  >
  > If there is a change in existing endpoint, then the same endpoint details need to be entered again with the new changes in order to update the existing provided service information in Routing Hub.

Import Service Provider
The user can create a service provider by importing the JSON file and also can import zip file in order to import all the configuration JSON files together (except parent level configuration JSON files).

7. Click **Import**.

   The **Import Service Provider** screen displays.

Figure 4-4 Import Service Provider



For more information on fields, refer to the field description table below.

Table 4-3 Import Service Provider - Field Description

Field	Description
File	Select the file using **Select** button. **Note:** Allows only to select one file and accepts JSON and ZIP file.
Overwrite extended templates	Select the respective radio button to overwrite extended templates. The options are: • **Yes** - This option overwrites the extended templates in configuration. • **No** - This option retains the existing extended templates in configuration. **Note:** This field appears only if the ZIP File is selected.

8. Click **Import** to import the selected file.

   The **Confirmation** screen displays.

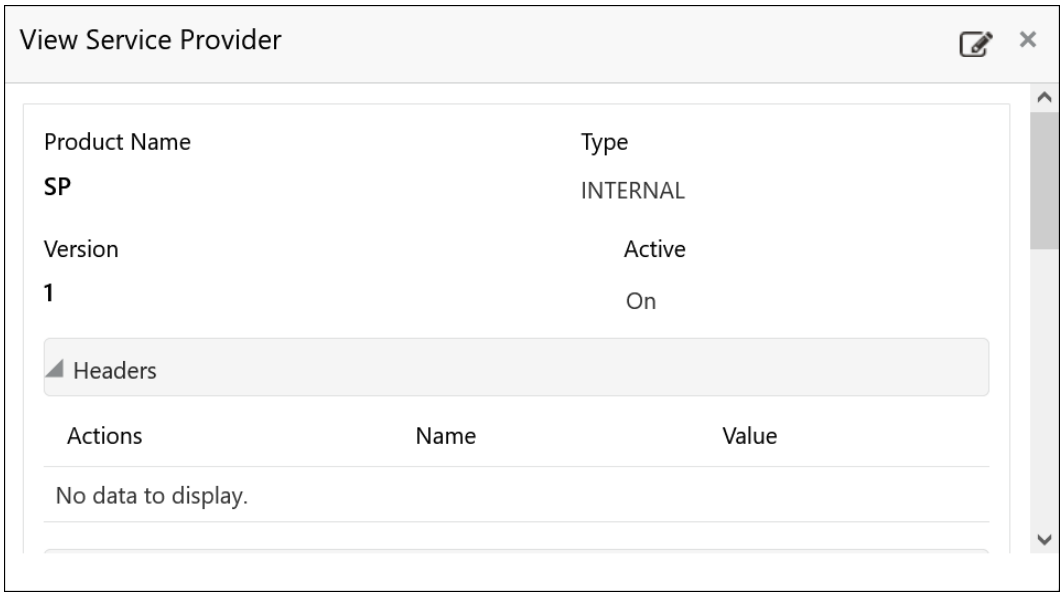
> **Note:**
>
> The following data needs to be changed after importing provider configuration file:
>
> - Implementation Host and Port
> - Implementation Authentication Password

View Service Provider

9. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **View**.

   The **View Service Provider** screen displays.

Figure 4-5 View Service Provider



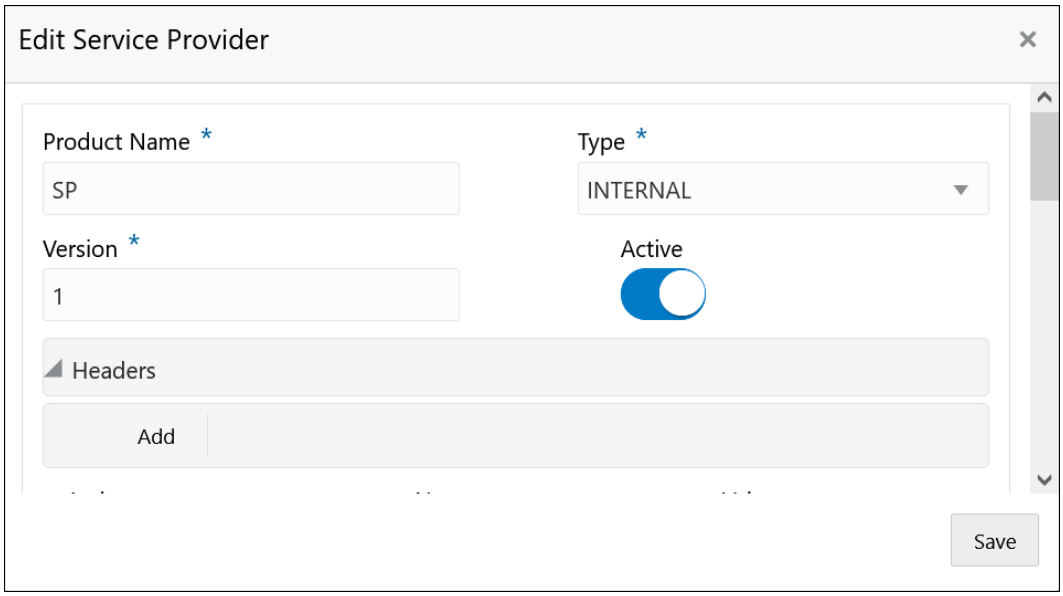
10. Click **Edit** button to edit the Service Provider.

Edit Service Provider
The user can modify the provider details.

11. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **Edit**.

    The **Edit Service Provider** screen displays.

Figure 4-6 Edit Service Provider



12. Click **Save** once the edit is done.

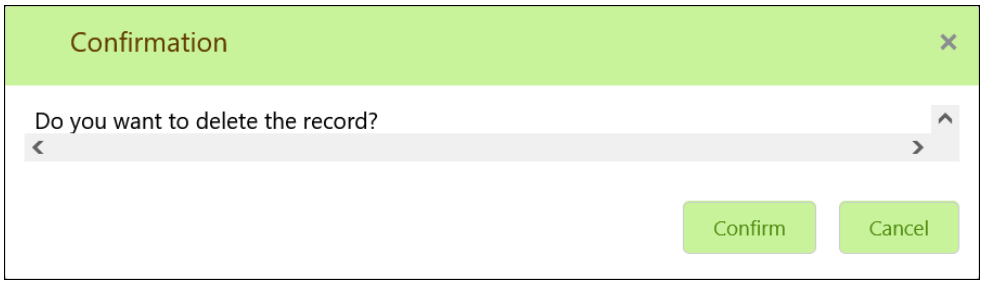
    The **Confirmation** screen displays.

Delete Service Provider
The user can delete the provider.

13. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **Delete**.

    The **Confirmation** screen displays.

Figure 4-7 Confirmation - Delete



14. Click **Confirm** to delete the selected Service Provider.

Export Service Provider

The user can export the provider configuration as JSON file.

15. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **Export**.

    The **Confirmation** screen displays.

Figure 4-8 Confirmation - Export

> **Note:**
>
> The below data cannot be exported:
>
> - Implementation Host
> - Implementation Port
> - Implementation Authentication Password
>
> The above data needs to be configured manually after importing the configuration file. Same has been mentioned in Import section.

16. Click **Confirm** to export the selected Service Provider.

Request Audit

17. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for the screen and field description.

Clear Cache
The user can clear the SOAP client cache for the service providers.

18. On **Service Provider** tile, click **Operation menu** (3 dots button), and click **Clear Cache**.

5
Implementation

This topic provides the systematic instructions to configure the implementation.

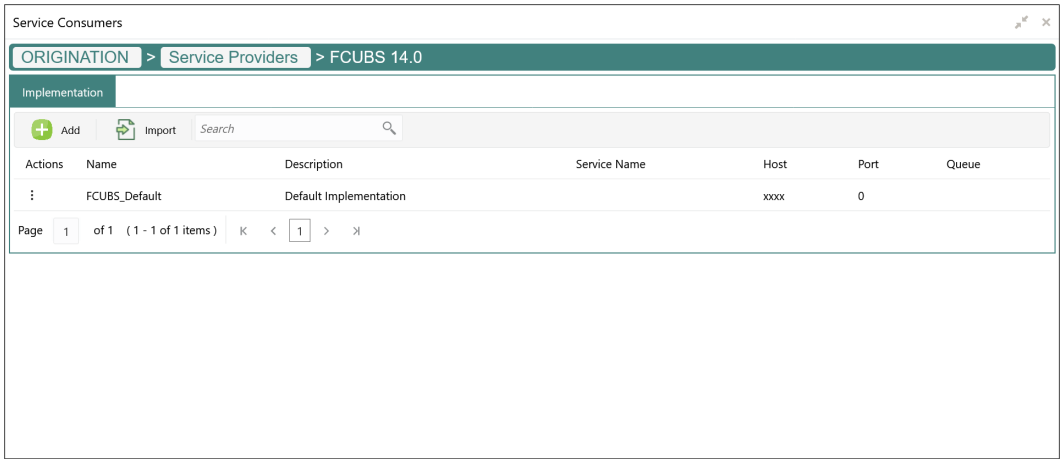
The implementation comprises of Eureka client instance, Host, Port, authentication, and implementation specific service details. Oracle Banking Routing Hub supports web services and Rest API.

> **Note:**
>
> Default implementation is created whenever a new service provider is added.

1. On **Service Provider** screen, click on the required service provider tile.

   The **Implementation** screen displays.

Figure 5-1 Implementation



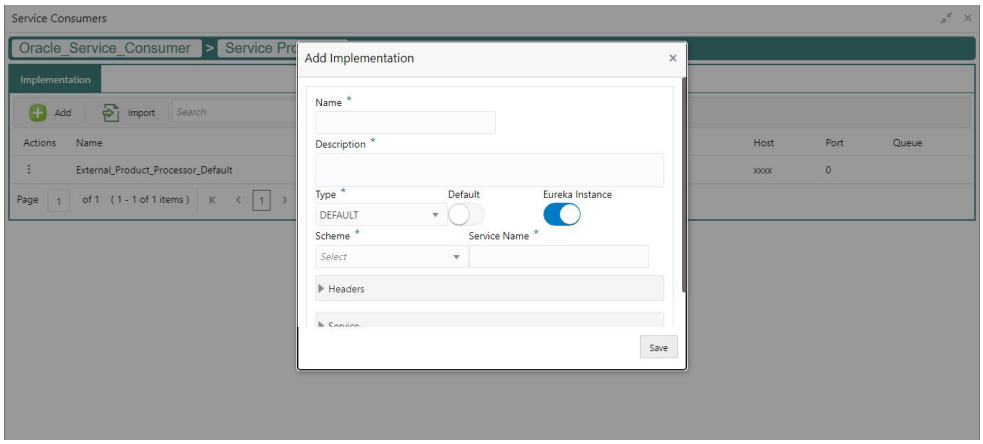
Add Implementation

The user can create the implementation manually.

2. Click **Add**.

   The **Add Implementation** screen displays.

Figure 5-2 Add Implementation



3. Specify the fields on **Add Implementation** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 5-1 Add Implementation - Field Description

Field	Description
Name	Specify the name of the implementation. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Description	Specify the description of the implementation. **Note:** • Enter 0 to 1000 characters. • No space allowed at beginning or ending of the characters.

Table 5-1 (Cont.) Add Implementation - Field Description

Field	Description
Type	Select the type of implementation from drop-down list The available options are: • **DEFAULT** • **QUEUE** **DEFAULT** type is for REST and SOAP API calls.
Default	Toggle the button if user wants to default. Each type can have one default implementation.
Eureka Instance	**Eureka Instance** is available only for internal providers and default type. By default, **Eureka Instance** will be toggled ON for internal providers and OFF for external providers.
Scheme	Select the scheme from drop-down list The available options are: • **http** • **https** **Scheme** option is available only for default type.
Service Name	If **Eureka Instance** is toggled ON and type is default, then only service name is required.
Host	Specify the host. **Note:** • Host cannot be blank. • Enter 0 to 255 characters. • Space is not allowed. If **Eureka Instance** is toggled OFF and type is default, then only host and port is required.
Port	Specify the port number. **Note:** • Port cannot be blank • Enter 0 to 6 characters. • Enter only numeric value. If **Eureka Instance** is toggled OFF and type is default, then only host and port is required.
Authentication	The below fields appear only if **Eureka Instance** is toggled **OFF** and Implementation **Type** is selected as **Default**.

Table 5-1 (Cont.) Add Implementation - Field Description

Field	Description
Type	Select the type of authentication from drop-down list. The available options are: • **BASIC** • **JWT_TOKEN** • **OAUTH_TOKEN** • **SSO** • **OAUTH_TOKEN_OIC**
Username	Specify the name of the user. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Password	Specify the password.
Headers	The below fields appear only if the Implementation **Type** is selected as **Default**.
Add	Click this button to add header.
Actions	Displays the action. The user can edit or delete the header.
Name	Displays the name of the header.
Value	Displays the value of the header.
Service	The below fields appear only if the Implementation **Type** is selected as **Default**.
Type	Select the type of service from drop-down list. The available options are: • **WSDL** • **SWAGGER** • **OTHERS**
URL	Specify the service URL of the file location. **Note:** This field appears only if the **Type** is selected as **WSDL** and **SWAGGER**.
Name	Specify the name of the operation. **Note:** This field appears only if the **Type** is selected as **OTHERS**.

Table 5-1 (Cont.) Add Implementation - Field Description

Field	Description
Http Method	Select the HTTP method. The available options are: • **GET** • **POST** • **PUT** • **PATCH** • **DELETE** **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Endpoint	Specify the endpoint URL for the operation. **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Gateway Prefix	Gateway Prefix is context path of below formatted URL. http://host:port/gateway-prefix/endpoint
Import	Click **Import** to extract the service information from URL and displays it in the Service list. **Note:** This field appears only if the **Type** is selected as **WSDL** and **SWAGGER**.
Service Headers	Specify the Endpoint specific headers. Value can either be hardcoded or can be Velocity mapping. **Note:** This field appears only if the **Type** is selected as **OTHERS**.

Table 5-1 (Cont.) Add Implementation - Field Description

Field	Description
Service Query Params	Specify the Endpoint specific query parameters. Value can either be hardcoded or can be Velocity mapping. **Note:** This field appears only if the **Type** is selected as **OTHERS**.
Add	Click this button to add the endpoint details in the Service list.
Service	Displays the extracted service from the selected URL.
Operation	Displays the extracted operation from the selected URL.

Authentication:
If External Product processor require authentication to connect to it, Oracle Banking Routing Hub provides standard authentication mechanism schemes like BASIC, JWT, OAUTH_TOKEN, SSO, OAUTH_TOKEN_OIC.

Note:

In case of no authentication, NONE needs to be set as Authentication Type.
In case of identity propagation, SSO needs to be set as Authentication Type.
In case of OIC integration, OAUTH_TOKEN_OIC needs to be set as Authentication Type

Services

- **WSDL**:
  The Web Services Description Language (WSDL) is an XML-based interface description language that is used for describing the functionality offered by a web service.

  Both SSL and non-SSL WSDL URL are supported.

  Note:

  If there is a change in wsdl file, then same wsdl file need to be imported again to update the provided service information in Routing Hub.

- **SWAGGER**:
  Swagger is an Interface Description Language for describing RESTful APIs expressed using JSON.

  Currently, Swagger 2.0 & OpenAPI 3.0 both are supported.

> **Note:**
>
> If there is a change in swagger file, then same swagger file need to be imported again in order to update the provided service information in Routing Hub.

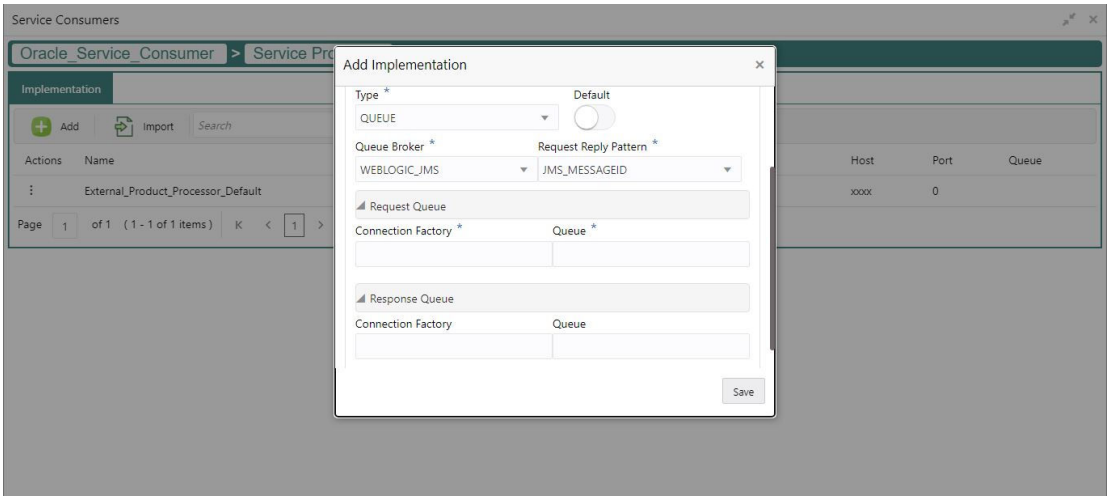
- **OTHERS**:
  OTHERS option is selected for adding REST API details manually when provider does not have swagger file.

> **Note:**
>
> If there is a change in existing endpoint, then the same endpoint details need to be entered again with the new changes in order to update the existing provided service information in Routing Hub.

Queue

If the Implementation **Type** is selected as **Queue**,

Figure 5-3 Add Implementation - Queue



For **QUEUE** type, refer to the field description table below.

Table 5-2 Add Implementation - Queue - Field Description

Field	Description
Type	Select the type of implementation from drop-down list The available options are: • **DEFAULT** • **QUEUE** **DEFAULT** type is for REST and SOAP API calls.
Default	Select the toggle if the user wants to default.

Table 5-2 (Cont.) Add Implementation - Queue - Field Description

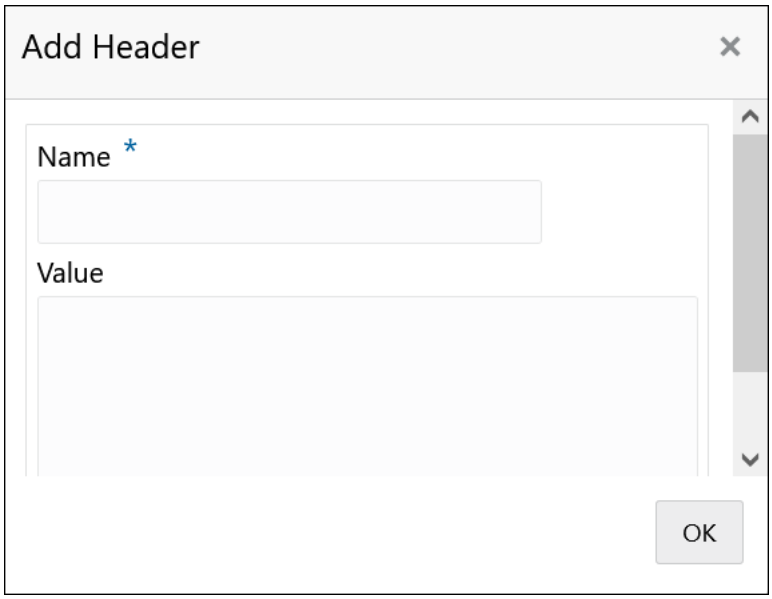
Field	Description
Queue Broker	Select the queue broker from drop-down list. The available options are: • **WEBLOGIC_JMS**
Request Reply Platform	Select the queue broker from drop-down list. The available options are: • **JMS_MESSAGEID** • **JMS_CORRELATIONID** **JMS_MESSAGEID** is default request-reply pattern.
Connection Factory	Specify the connection factory. Connection Factory is JNDI based connection factory name which is used to create connection for JMS client.
Queue	Specify the queue. Queue Name is JNDI based destination name.
Connection Factory	Specify the connection factory. Response Connection Factory is needed when destination is going to respond back after processing the request.
Queue	Specify the queue. Response Queue Name is needed when destination is going to respond back after processing the request.

Headers:
External product processor might require some standard headers to be passed along with the request. User can specify the headers which are required by service endpoints for its all implementations but not present in swagger file.

4. Follow the below steps to add **Headers**.
   a. Click **Add** under **Header** section.

      The **Add Header** screen displays.

Figure 5-4 Add Header



b. Specify the fields on **Add Header** screen and click **OK**.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 5-3 Add Header - Field Description

Field	Description
Name	Specify the name for the header.
Value	Specify the value for the header.

5. Click **OK** to save the details.

   The **Confirmation** screen displays.

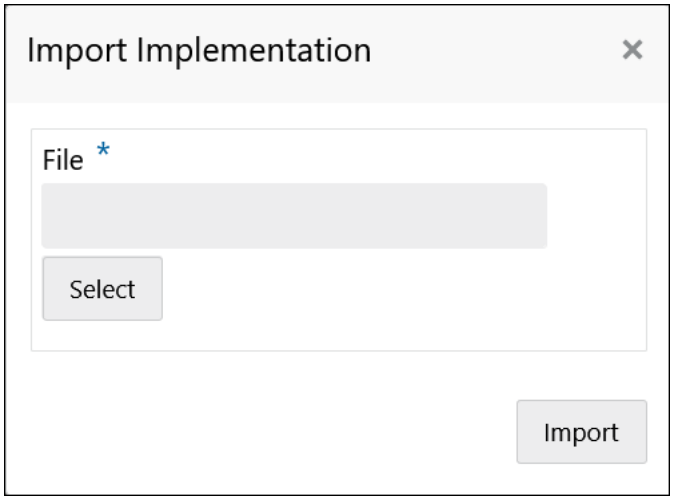
Import Implementation

The user can create an implementation by importing the JSON file. The user can also import zip file in order to import all the configuration JSON files together (except parent level configuration JSON files).

6. On **Implementation** screen, click **Import**.

   The **Import Implementation** screen displays.

Figure 5-5 Import Implementation



For more information on fields, refer to the field description table.

Table 5-4 Import Implementation - Field Description

Field	Description
File	Click **Select** to select the file. **Note:** Allows only to select one file and accepts JSON and ZIP file.

7. Click **Import** to import the selected file.

   The **Confirmation** screen displays.

> **Note:**
>
> The below data needs to be changed after importing provider configuration file:
>
> - Implementation Host and Port
> - Implementation Authentication Password

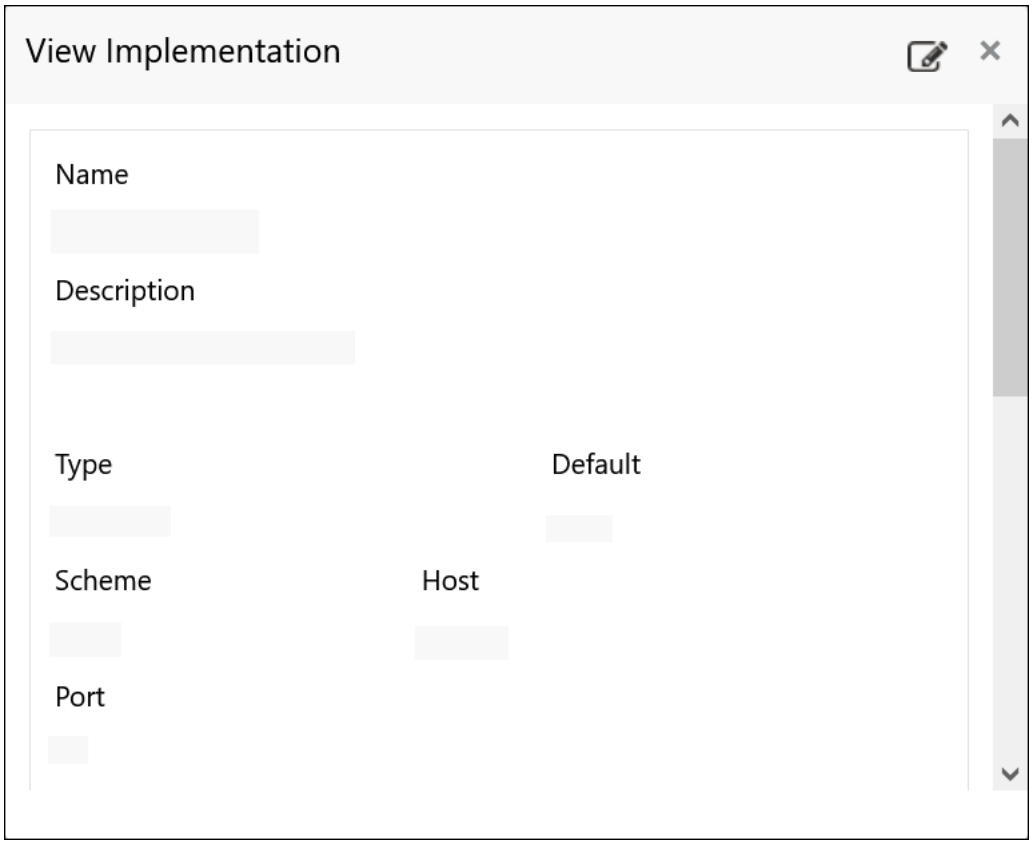
View Implementation

The user can view implementation details and can also switch to edit form by clicking on edit icon.

8. On **Implementation** screen, click **Operation menu** (3 dots button) and click **View**.

   The **View Implementation** screen displays.

Figure 5-6 View Implementation



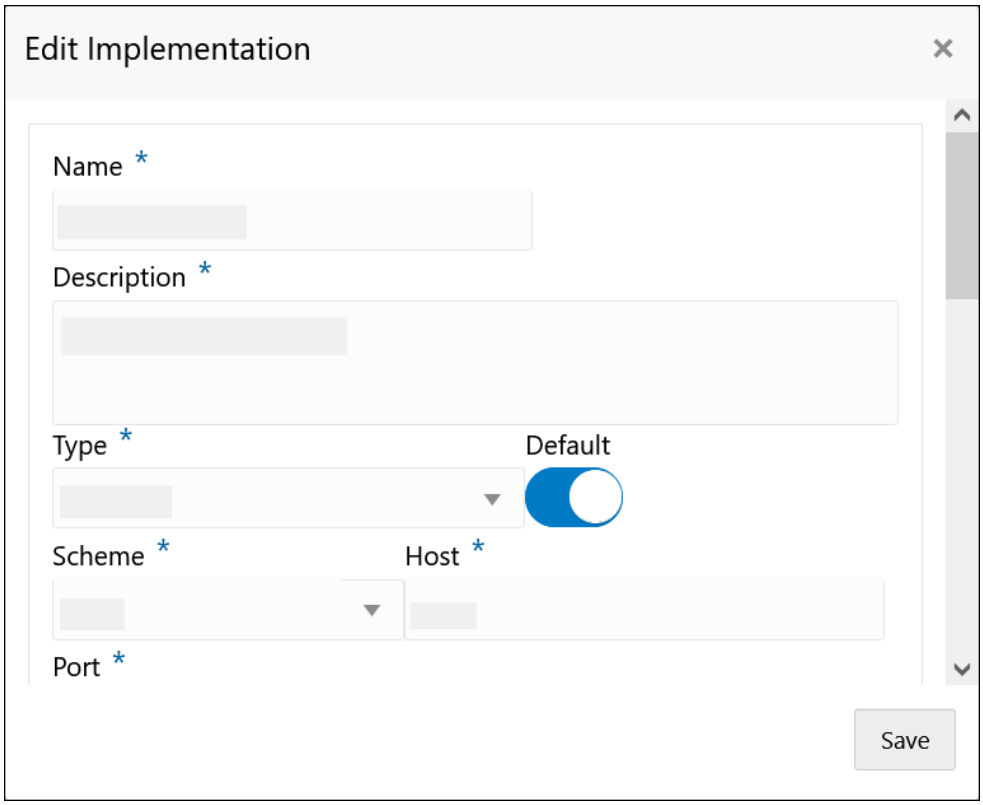
Edit Implementation

The user can modify the implementation details.

9. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Edit**.

   The **Edit Implementation** screen displays.

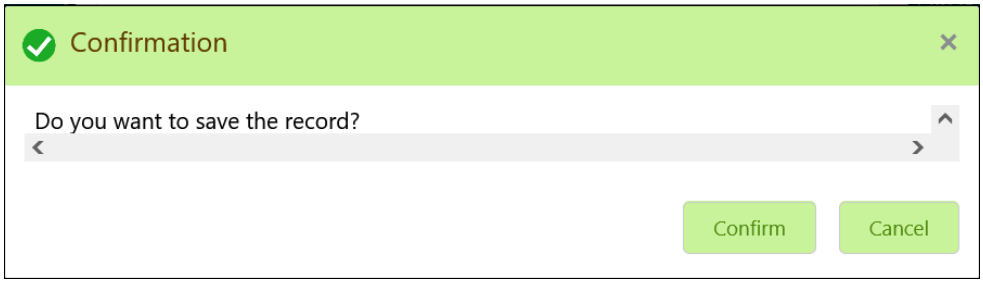
Figure 5-7 Edit Implementation



10. Click **Save** once the edit is done.

    The **Confirmation** screen displays.

Figure 5-8 Confirmation

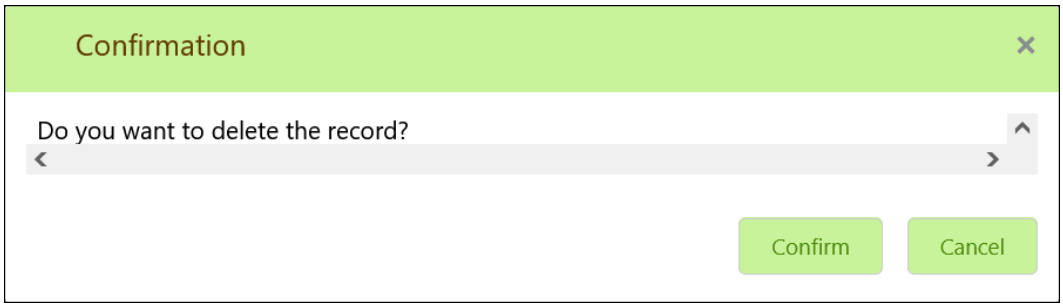


Delete Implementation
The user can delete the implementation details.

11. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Delete**.

    The **Confirmation** screen displays.

Figure 5-9 Confirmation - Delete



Export Implementation

The user can export the implementation configuration as JSON file.

12. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Export**.

    The **Confirmation** screen displays.

Figure 5-10 Confirmation - Export Implementation

> **Note:**
>
> Below data cannot be exported:
>
> - Implementation Host
> - Implementation Port
> - Implementation Authentication Password
>
> The above data needs to be configured manually after importing the configuration file. Same has been mentioned in Import section.

Request Audit

13. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for screen and field description.

Clear Cache

The user can clear the SOAP client cache.

14. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Clear Cache**.

6
Consumer Services

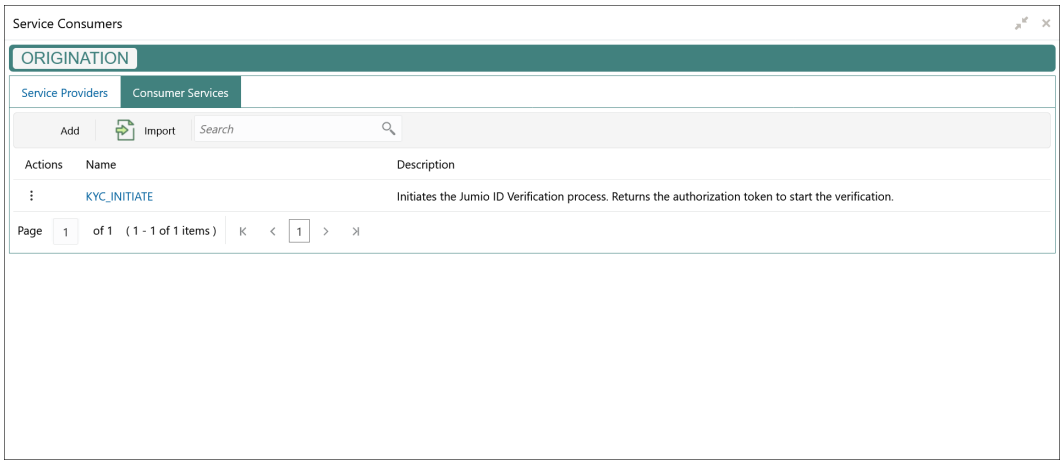
This topic describes the systematic instructions to configure the consumer services.

The **Consumer Services** defines the service ID, which sends from the service consumer. It also caters the transition and route definition. It comprises of source integration details.

1. On **Service Consumers** screen, click **Consumer Services**.

   The **Consumer Services** screen displays.

Figure 6-1 Consumer Services



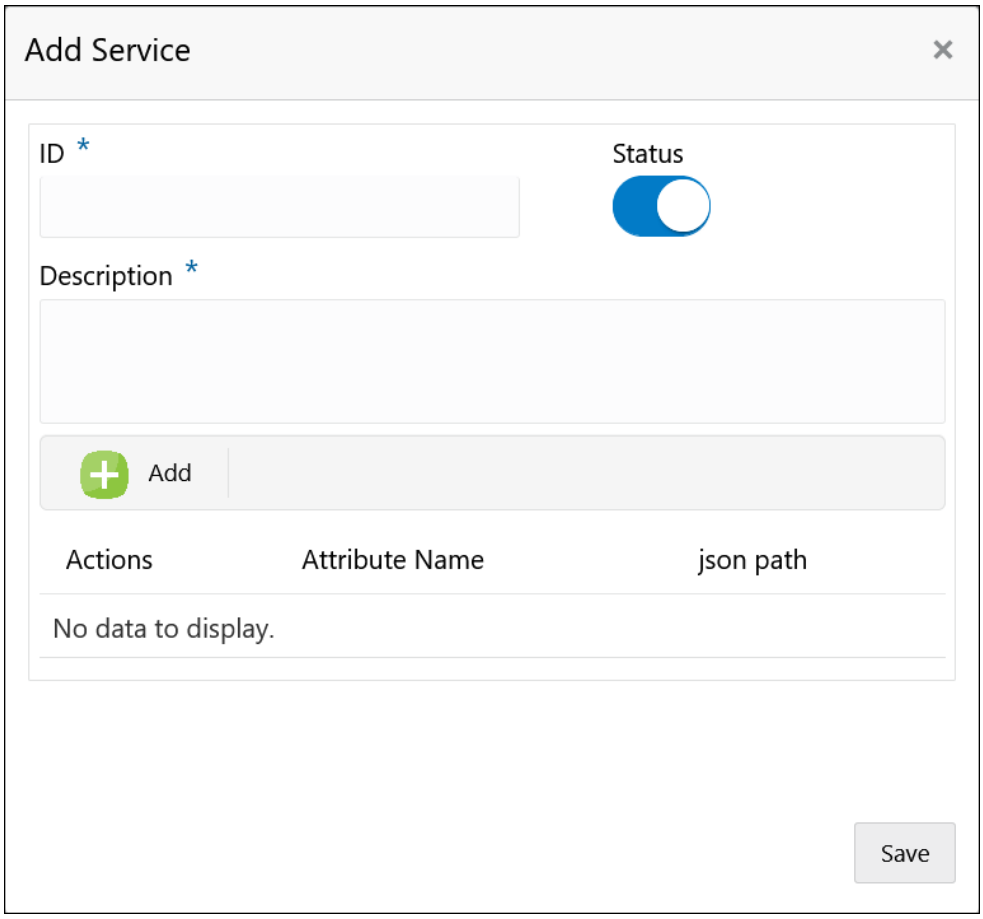
Add Consumer Service

The user can create Consumer Service manually.

2. On **Consumer Services** screen, click **Add**.

   The **Add Service** screen displays.

Figure 6-2 Add Service



3. Specify the fields on **Add Service** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 6-1 Add Service - Field Description

Field	Description
ID	Specify the ID of the consumer service. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.

Table 6-1 (Cont.) Add Service - Field Description

Field	Description
Status	Active / Inactive If consumer service is marked as inactive, then all related routes will be stopped.
Description	Specify the description of the consumer service. **Note:** • Enter 0 to 1000 characters. • No space allowed at beginning or ending of the characters.
Add	To add, refer to the below step.
Actions	Displays the action. The user can edit or delete the header.
Attribute Name	Displays the name of the attribute.
json path	Displays the JSON path.

4. To add **Attributes**, follow the below steps.
   a. Click **Add**.

      The **Add Attribute** screen displays.

Figure 6-3 Add Attribute

   b. Specify the fields on **Add Attribute** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 6-2 Add Header - Field Description

Field	Description
Name	Specify the name of the attribute.
josn path	Specify the json path.

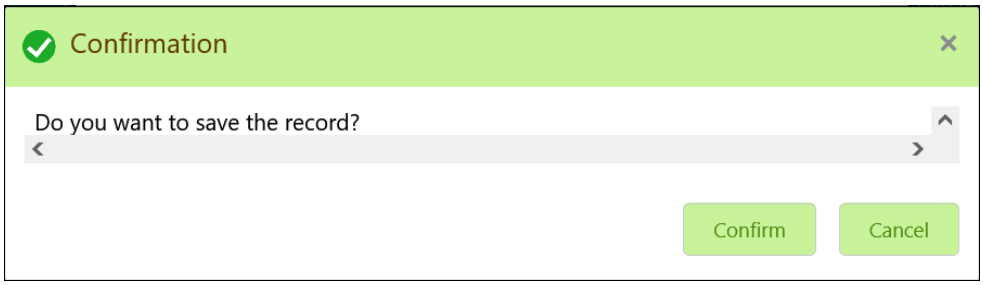
> **Note:**
>
> - Using $.body, the user can access the request body.
> Syntax: $.body.fieldName
>
> Example: $.body.branchCode
> - Using $.headers, the user can access the request headers.
> Syntax: $.headers["fieldName"][0]
>
> Example: $.headers["branchCode"][0]
> - Using $.env, the user can access the environment variables.
> Syntax: $.env.group.variable

c. Click **OK** to save the attributes.

5. Click **Save** to save the details.

   The **Confirmation** screen displays.

Figure 6-4 Confirmation



6. On **Confirmation** screen, click **Confirm** to add the service.

Import Consumer Service

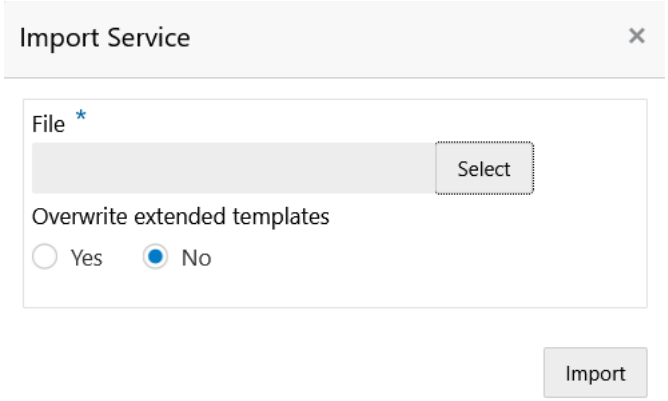
The user can create a consumer service by importing the JSON file.

The user can also import zip file in order to import all the configuration JSON files together (except parent level configuration JSON files).

7. On **Consumer Services** screen, click **Import**.

The **Import Service** screen displays.

Figure 6-5 Import Service



For more information on fields, refer to the field description table.

Table 6-3 Import Service - Field Description

Field	Description
File	Select the file using **Select** button. **Note:** Allows only to select one file and accepts only JSON file.
Overwrite extended templates	Select the respective radio button to overwrite the extended templates. The available options are: • Yes - This option overwrites the extended templates. • No - This option retains the existing extended templates.

8. Click **Import** to import the selected file.

   The **Confirmation** screen displays.

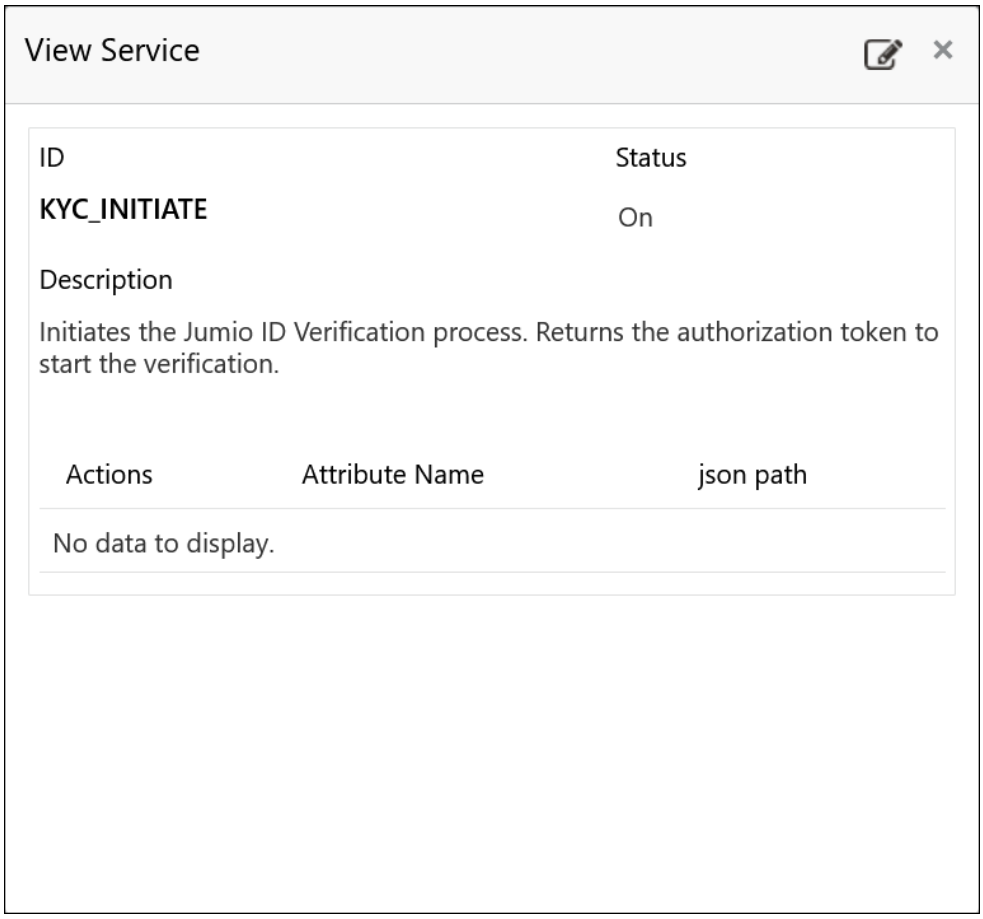
View Consumer Service

The user can view consumer service details and can also switch to edit form by clicking on edit icon.

9. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **View**.

   The **View Service** screen displays.

Figure 6-6 View Service



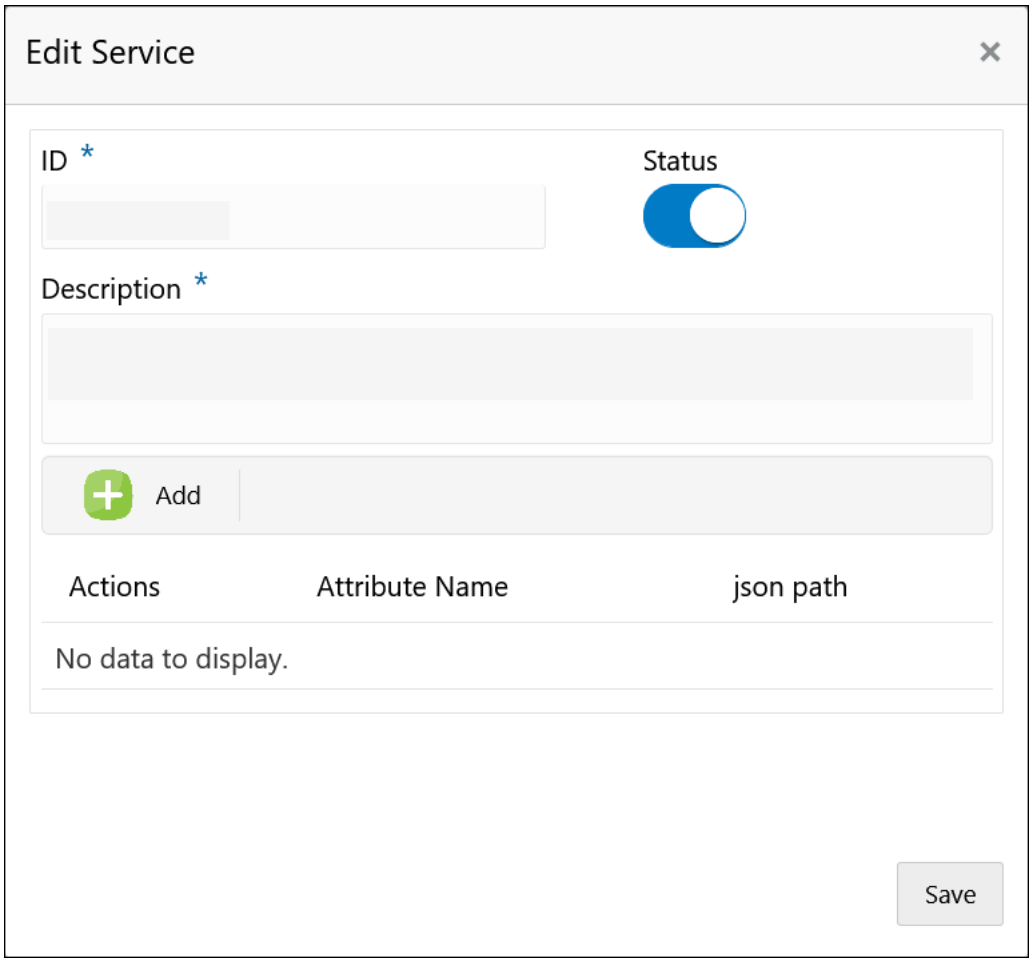
Edit Consumer Service

The user can modify the consumer service details.

10. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **Edit**.

    The **Edit Service** screen displays.

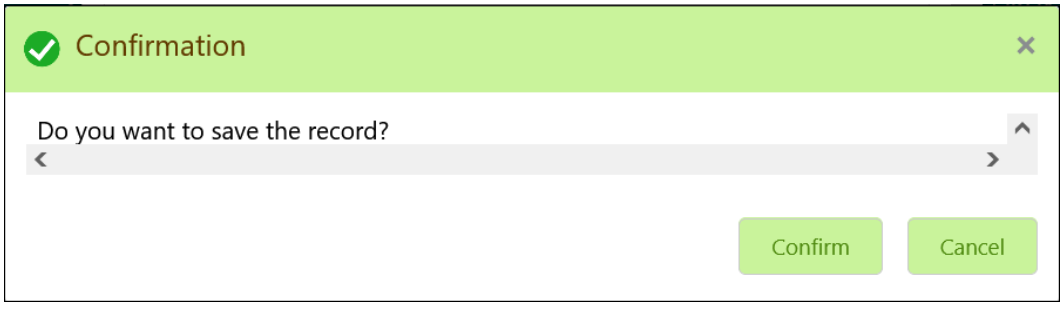
Figure 6-7 Edit Service



11. Click **Save** once the edit is done.

    The **Confirmation** screen displays.

Figure 6-8 Confirmation - Edit



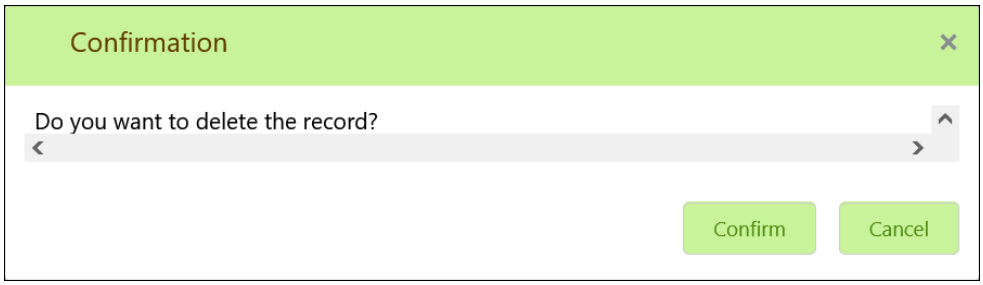
Delete Consumer Service

The user can delete the consumer service.

12. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **Delete**.

    The **Confirmation** screen displays.

Figure 6-9 Confirmation



Export Consumer Service

The user can export the consumer service configuration as JSON file.

13. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **Export**.

    The **Confirmation** screen displays.

Figure 6-10 Confirmation - Export

Consumer Service - Configuration

14. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **Configuration**.

    The **Configuration** screen displays.

> **Note:**
>
> Refer to Configuration topic for the screen and field description.

Consumer Service - Request Audit

15. On **Consumer Service** tile, click **Operation menu** (3 dots button), and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for the screen and field description.

7
Transformation

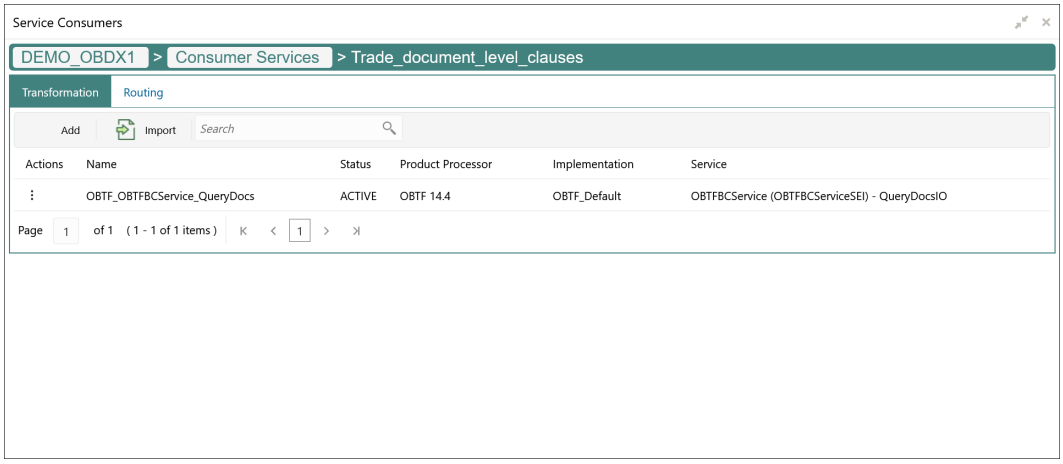
This topic describes the systematic instructions to configure the transformation.

The **Transformation** acts as assembling and transforming data from source to destination and vice-versa. This takes place under consumer service. This converts the data of service consumer into service provider.

1. On **Consumer Services** screen, click the required consumer service tile.

   The **Transformation** screen displays.

Figure 7-1 Transformation

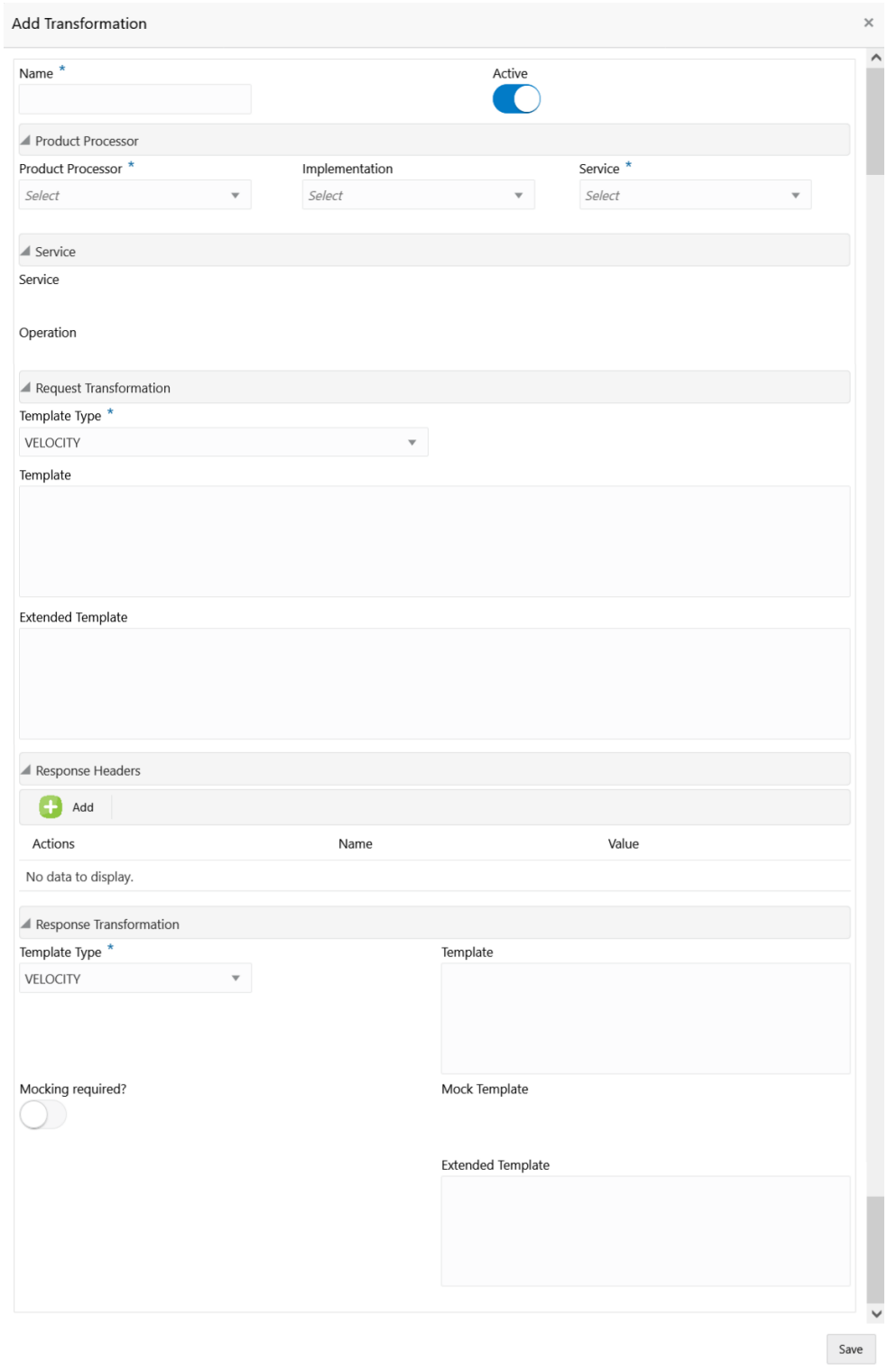


Add Transformation
The user can create transformation manually.

2. On **Transformation** screen, click **Add**.

   The **Add Transformation** screen displays.

Figure 7-2 Add Transformation



3. Specify the fields on **Add Transformation** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 7-1 Add Transformation - Field Description

Field	Description
Name	Specify the name for the transformation. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
Active	Active / Inactive If transformation is marked as inactive, then the user will not be able to select transformation in routing.
Product Processor	Displays the Product Processor details.
Product Processor	Select the product processor from the drop-down list.
Implementation	Select the implementation from the drop-down list.
Service	Select the service from the drop-down list.
Service	Displays the service details of the selected service.
Headers	Displays the header list relevant to the selected provider, implementation and service. User can change the header values. The value can either be hardcoded or can be Velocity mapping.
Path Params	Displays the path param list relevant to the selected service. User can change the param values. Value can either be hardcoded or can be Velocity mapping.
Query Params	Displays the query param list relevant to the selected service. User can change the param values. Value can either be hardcoded or can be Velocity mapping.
Request Transformation	Displays the Request Transformation details.
Body Type	Select the body type for the Request Transformation from the drop-down list. The available options are: • **RAW** • **FORM DATA** **Note:** This field appears only if the selected service is REST service.

Table 7-1 (Cont.) Add Transformation - Field Description

Field	Description
Template Type	Select the template type for the Request Transformation from the drop-down list. The available options are: • **VELOCITY** • **JSLT** • **XSLT**
Template	Specify the template for the Request Transformation in which provider accepts. Refer to Transformation Type for syntax.
Extended Template	Specify the custom template in order to extend the kernel template. Refer to Extensibility and Transformation Type for syntax. **Note:** This field appears only if the **Body Type** is selected as **FORM DATA**.
Response Header	Specify the additional headers required to be part of Routing Hub response headers. Value can either be hardcoded or can be Velocity mapping.
Response Transformation	Displays the response transformaton details.
Template Type	Select the template type for the Response Transformation from drop-down list. The available options are: • **VELOCITY** • **JSLT** • **XSLT**
Template	Specify the kernel template in which consumer accepts. Refer to Transformation Type for syntax.
Mocking required?	Select the toggle if the mocking is required for the Response Transformation or not. If the toggle is **ON**, the Routing Hub will return the mocked template output (with extended template output if mentioned) to consumer without invoking provider API.
Mock Template	Specify the kernel template for the Response Transformation in which the consumer accepts. Refer Transformation Type for syntax.

4. Click **Save** to save the details.

   The **Confirmation** screen displays.

5. Click **Confirm** to add the transformation.

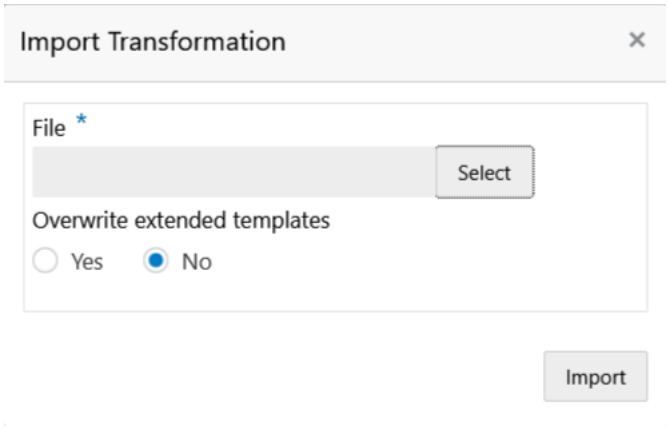
Import Transformation

The user can create a transformation by importing the JSON file. The user can also import zip file in order to import all the configuration JSON files together (except parent level configuration JSON files).

6. On **Transformation** screen, click **Import**.

The **Import Transformation** screen displays.

Figure 7-3 Import Transformation



For more information on fields, refer to the field description table.

Table 7-2 Import Transformation - Field Description

Field	Description
File	Select the file using **Select** button. **Note:** Allows only to select one file and accepts JSON and ZIP file.
Overwrite extended templates	Select the respective radio button to overwrite the extended templates. The available options are: • Yes - This option overwrites the extended templates. • No - This option retains the existing extended templates.

7. Click **Import** to import the selected file.

   The **Confirmation** screen displays.

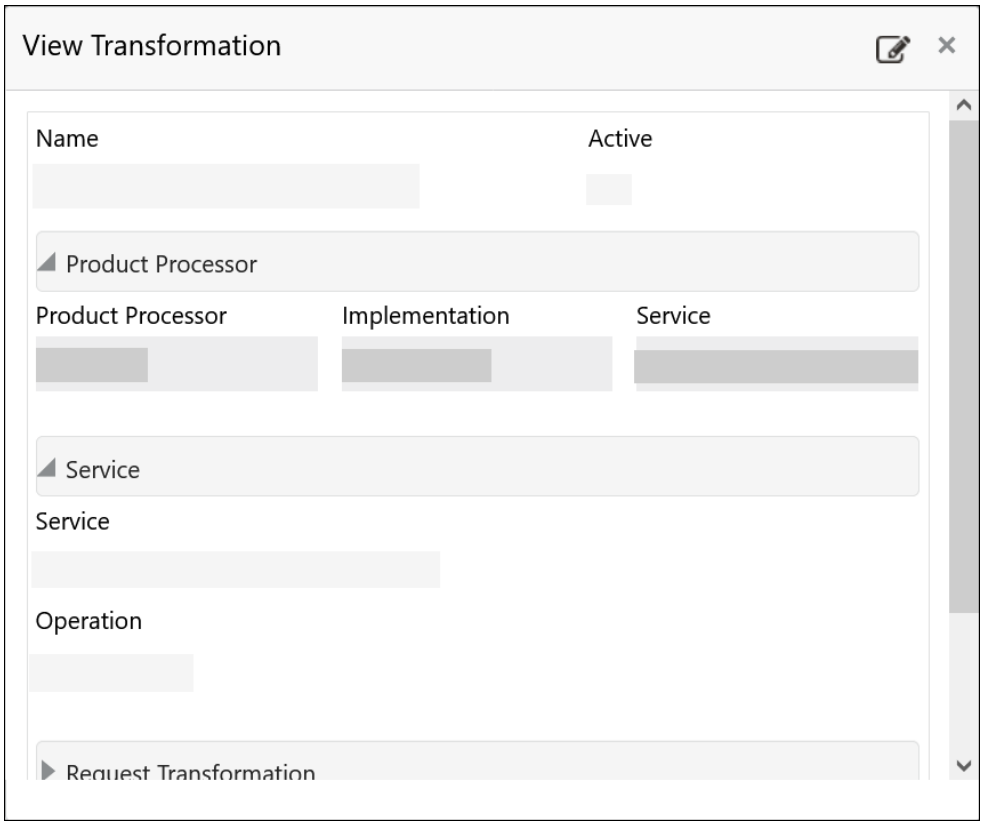
View Transformation

The user can view transformation details and can also switch to edit form by clicking on edit icon.

8. On **Transformation** list, click **Operation menu** (3 dots button), and click **View**.

   The **View Transformation** screen displays.

Figure 7-4 View Transformation



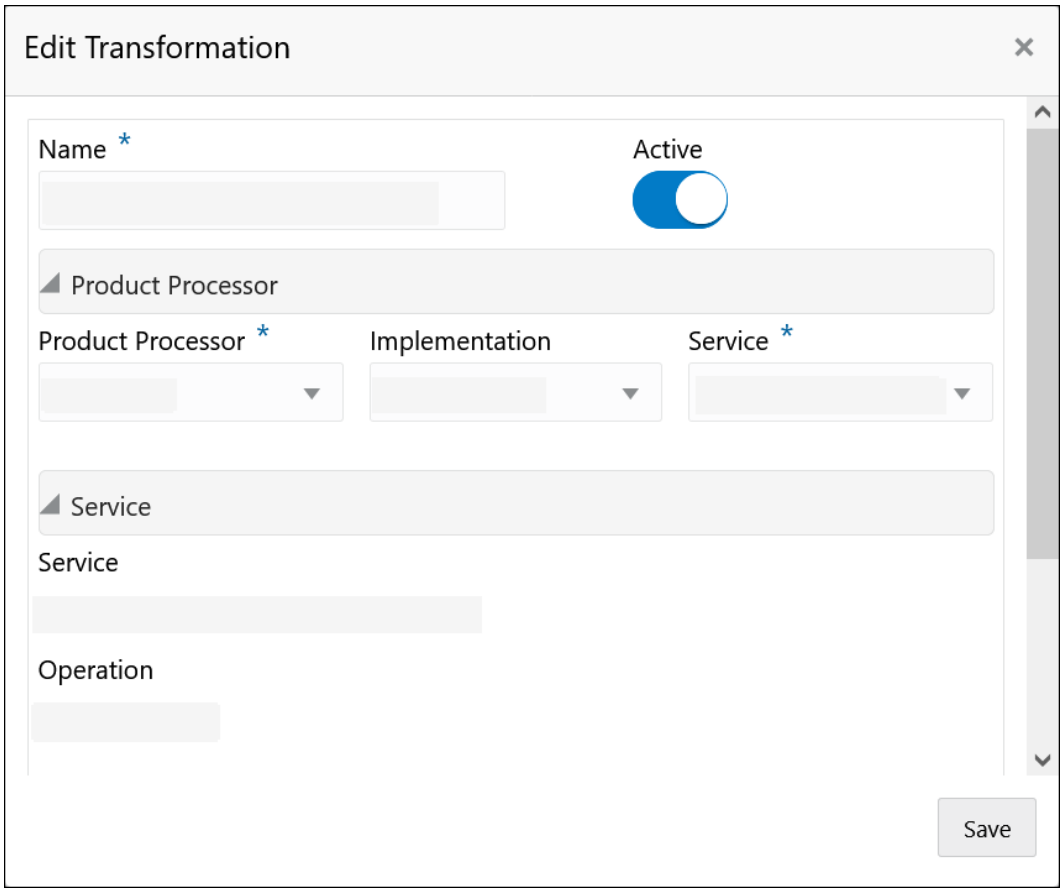
Click **Edit** icon to edit the **Transformation**.

Edit Transformation
The user can modify the transformation details.

9. On **Transformation** list, click **Operation menu** (3 dots button), and click **Edit**.

   The **Edit Transformation** screen displays.

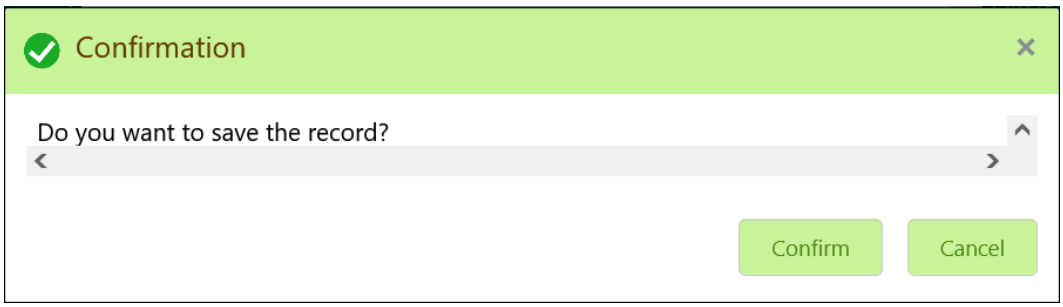
Figure 7-5 Edit Transformation



10. Click **Save** once the edit is done.

    The **Confirmation** screen displays.

Figure 7-6 Confirmation



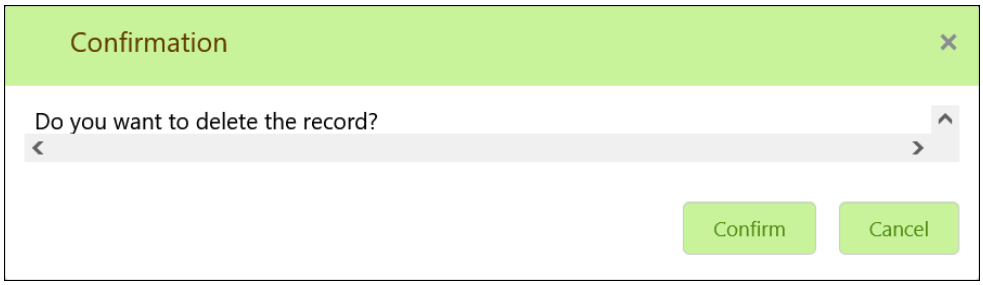
Delete Transformation

The user can delete the transformation.

11. On **Transformation** list, click **Operation menu** (3 dots button), and click **Delete**.

    The **Confirmation - Delete** screen displays.

Figure 7-7 Confirmation - Delete



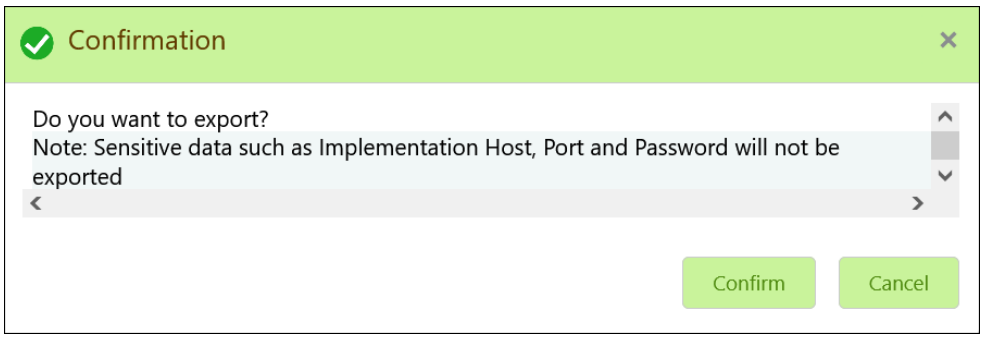
Export Transformation

The user can export the transformation configuration as JSON file.

12. On **Transformation** list, click **Operation menu** (3 dots button), and click **Export**.

    The **Confirmation** screen displays.

Figure 7-8 Confirmation - Export



Request Audit

13. On **Transformation** list, click **Operation menu** (3 dots button), and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for screen and field description.

8
Routing

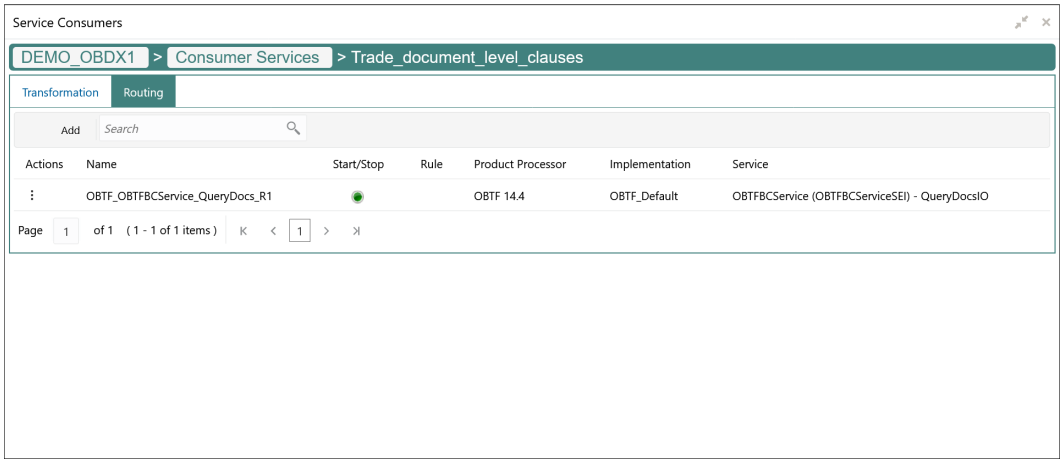
This topic describes the systematic instructions to configure the routing.

Routing defines no rule or rule-based route configuration. Route decide the actual request to be send to which service provider based on maintenance and evaluation.

1. On **Consumer Services** screen, click **Routing**.

   The **Routing** screen displays.

Figure 8-1 Routing

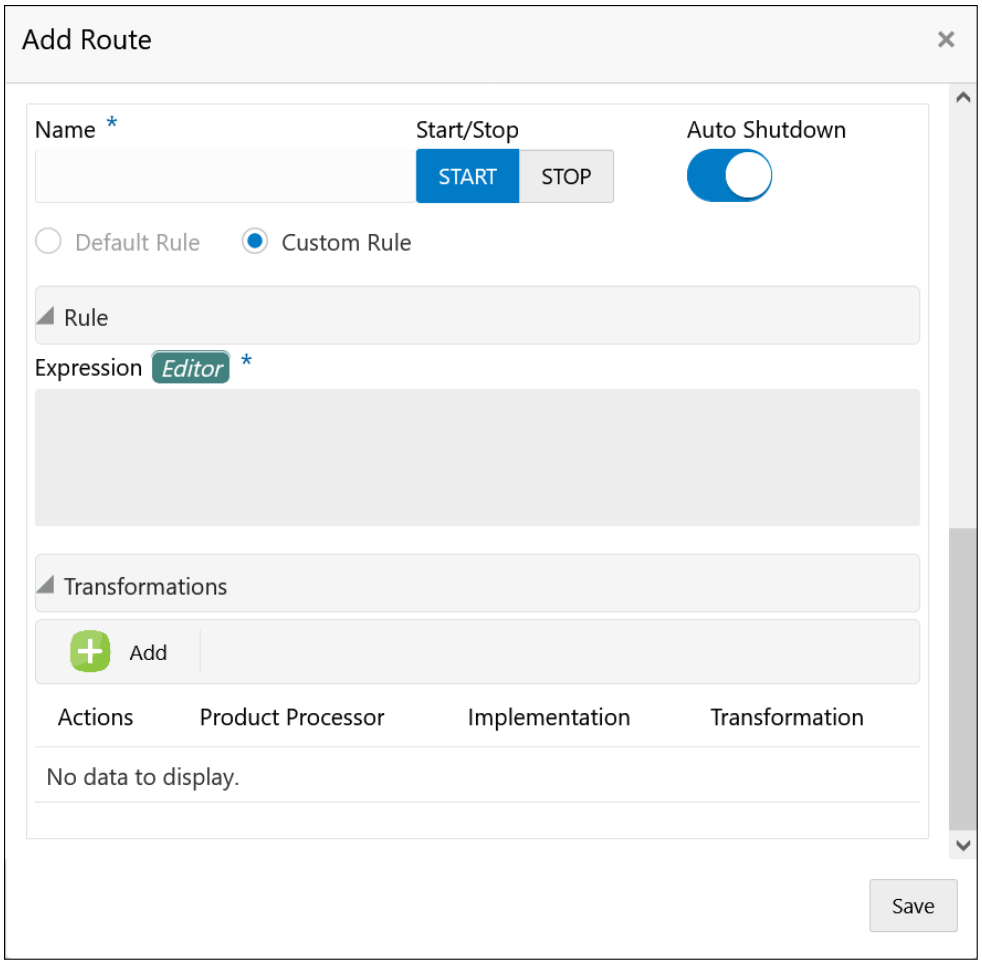


Add Route
The user can create routing manually.

2. On **Routing** screen, click **Add**.

   The **Add Route** screen displays.

Figure 8-2 Add Route



3. Specify the fields on **Add Route** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 8-1 Add Route - Field Description

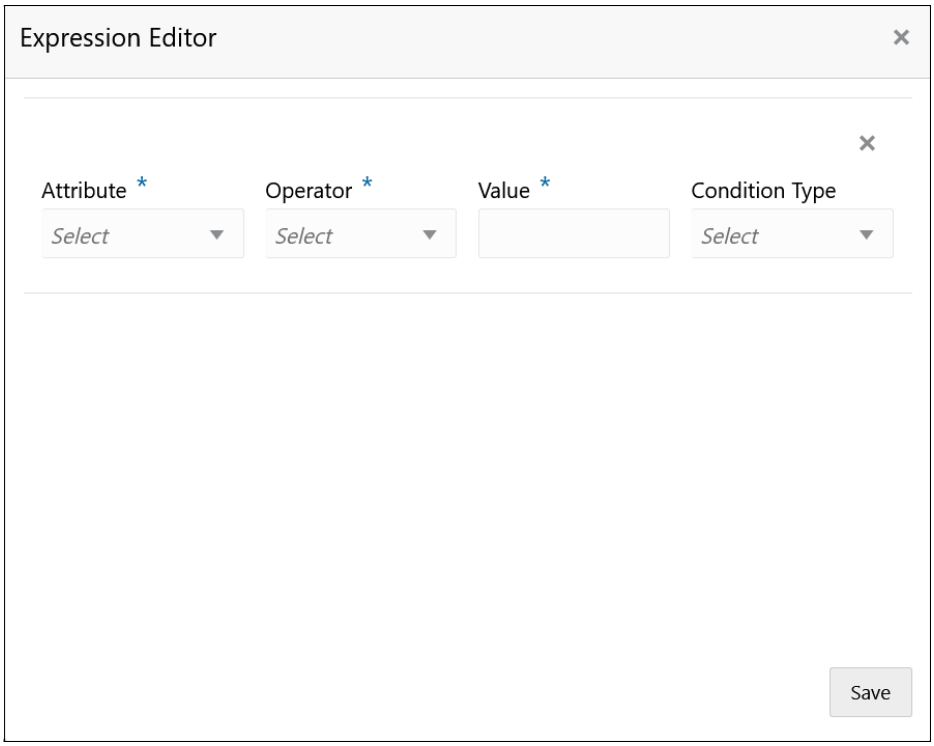
Field	Description
Name	Specify the name for the route. **Note:** • Enter 0 to maximum of 255 characters. • No numeric value at beginning and no space allowed.
START / STOP	START / STOP If routing is marked as STOP, then consumer request fails at routing hub level only.
Auto Shutdown	ON / OFF If AutoShutdown flag is ON, then route state will be changed to STOP if route failure goes beyond the threshold failure limit based on the monitoring and alert configuration.
Rule Type	Select the rule type. The available options are: • **Default Rule** • **Custom Rule**
Expression Editor	Displays the expression that is formed through expression editor.
Add	To add, refer to the below steps.
Actions	Displays the action. The user can edit or delete the header.
Product Processor	Displays the product processor.
Implementation	Displays the implementation.
Transformation	Displays the transformation.

Add Custom Rule using Expression Editor

4. To add **Editor**, follow the below steps.
   a. On **Add Route** screen, click **Editor** button.

      The **Expression Editor** screen displays.

Figure 8-3 Expression Editor



b. Specify the fields on **Expression Editor** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 8-2 Expression Editor - Field Description

Field	Description
Attribute	Select attribute relevant to consumer service from drop-down list.
Operator	Select the logical operators to form an expression from drop-down list.
Value	Specify the value. **Note:** Enter 0 to 255 characters.
Condition Type	Select the condition type from drop-down list.

c. Click **Save** to save the details.

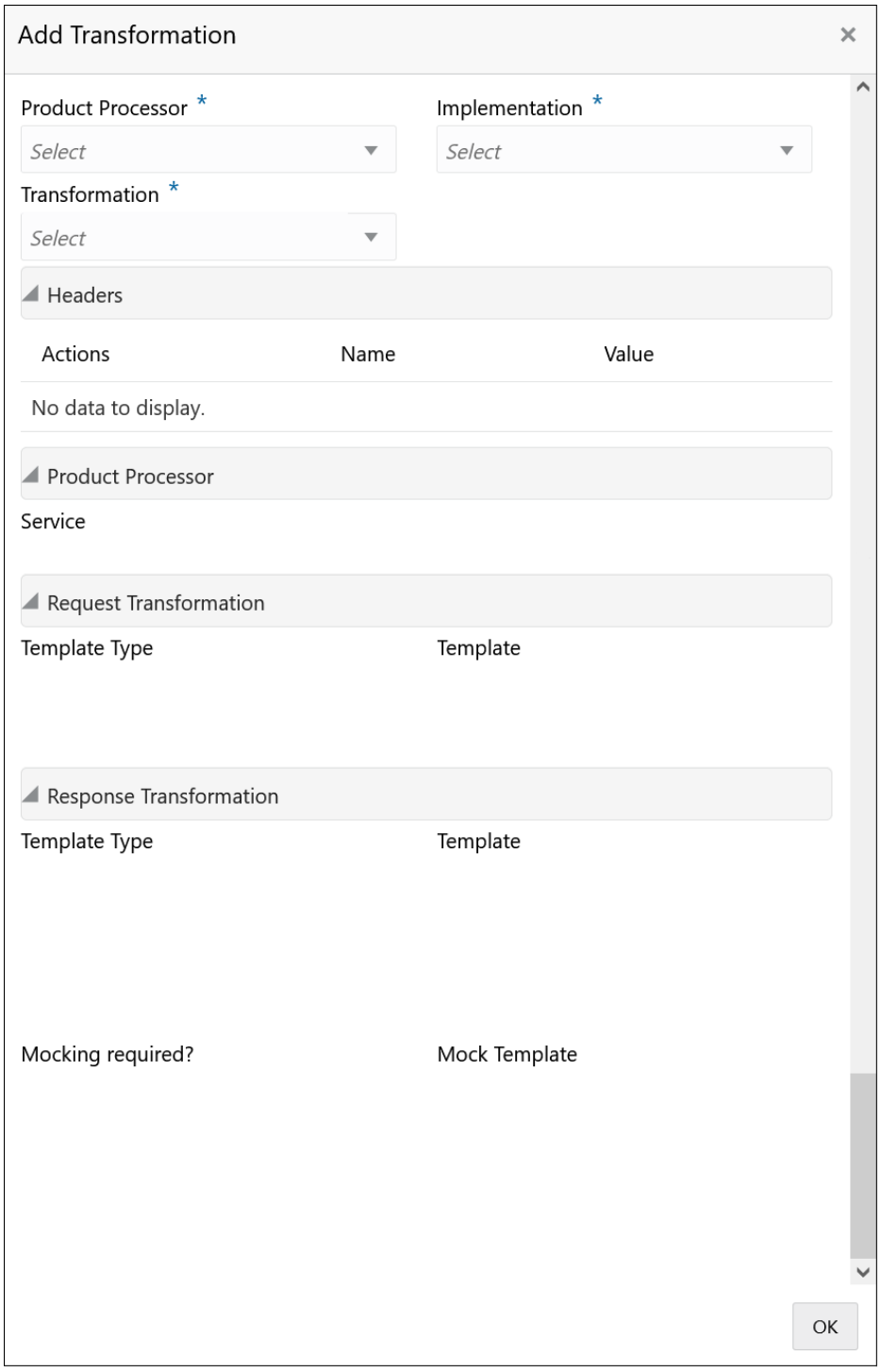
> **Note:**
>
> String value should be quoted using single quotes (') Example: 'abc'
> List value should be comma separated values and quoted using single quotes (') Example: 'abc,xyz,1.23,true'

Transformations
The user can define the sequence of transformations for each routing in which request should be processed. Sequence of transformations in list can be changed by using drag-n-drop feature.

5. To add **Transformations**, follow the below steps.
   a. On **Add Route** screen, click **Add**.

      The **Add Transformation** screen displays.

Figure 8-4 Add Transformation



b. Specify the fields on **Add Transformation** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 8-3 Add Transformation - Field Description

Field	Description
Product Processor	Select the product processor from the drop-down list.
Implementation	Select the implementation from the drop-down list.
Transformation	Select the transformation from the drop-down list.
Action	Displays the action. The user can edit or delete the header.
Name	Displays the name of the header.
Value	Displays the value of the header.
Service	Displays the service of the product processor.
Template Type	Displays the template type for the request transformation.
Template	Displays the template for the request transformation.
Template Type	Displays the template type for the response transformation.
Template	Displays the template for the response transformation.
Mocking required?	Displays whether the mocking required for the response transformation or not.
Mock Template	Displays the mock template for the response transformation.

c. Click **OK**.

6. Click **Save** to save the details.

   The **Confirmation** screen displays.

7. Click **Confirm** to add the routing details.

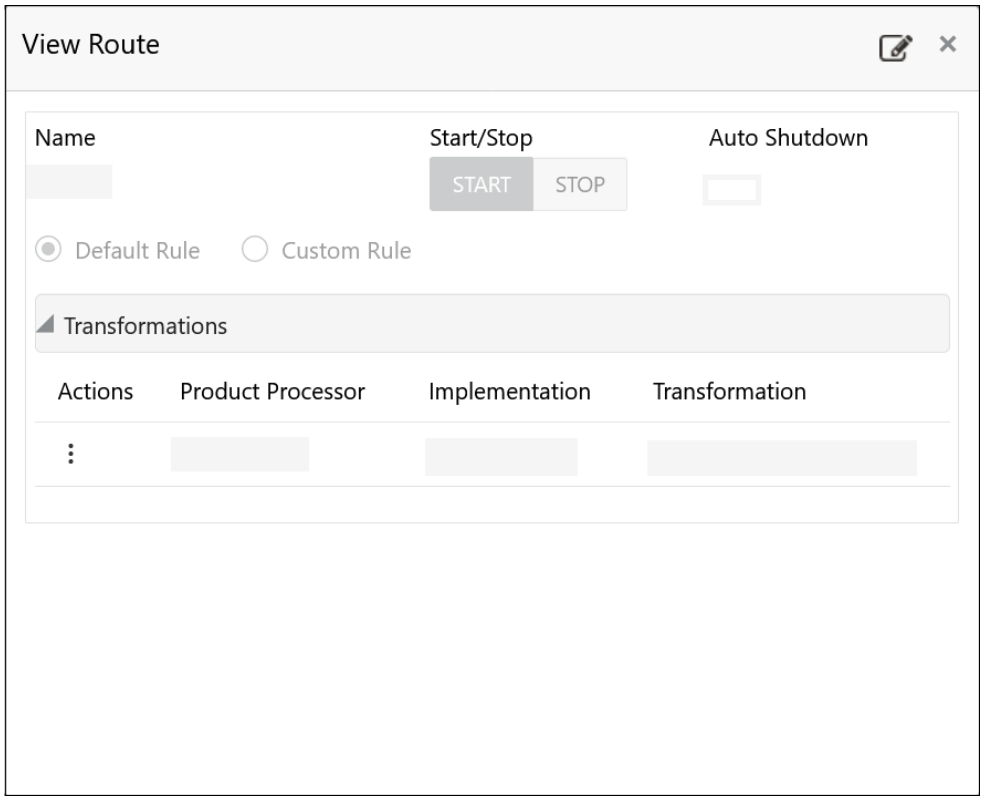
View Route

The user can view the routing details and can also switch to edit form by clicking on edit icon.

8. On **Routing** screen, click **Operation menu** (3 dots button), and click **View**.

   The **View Route** screen displays.

Figure 8-5 View Route



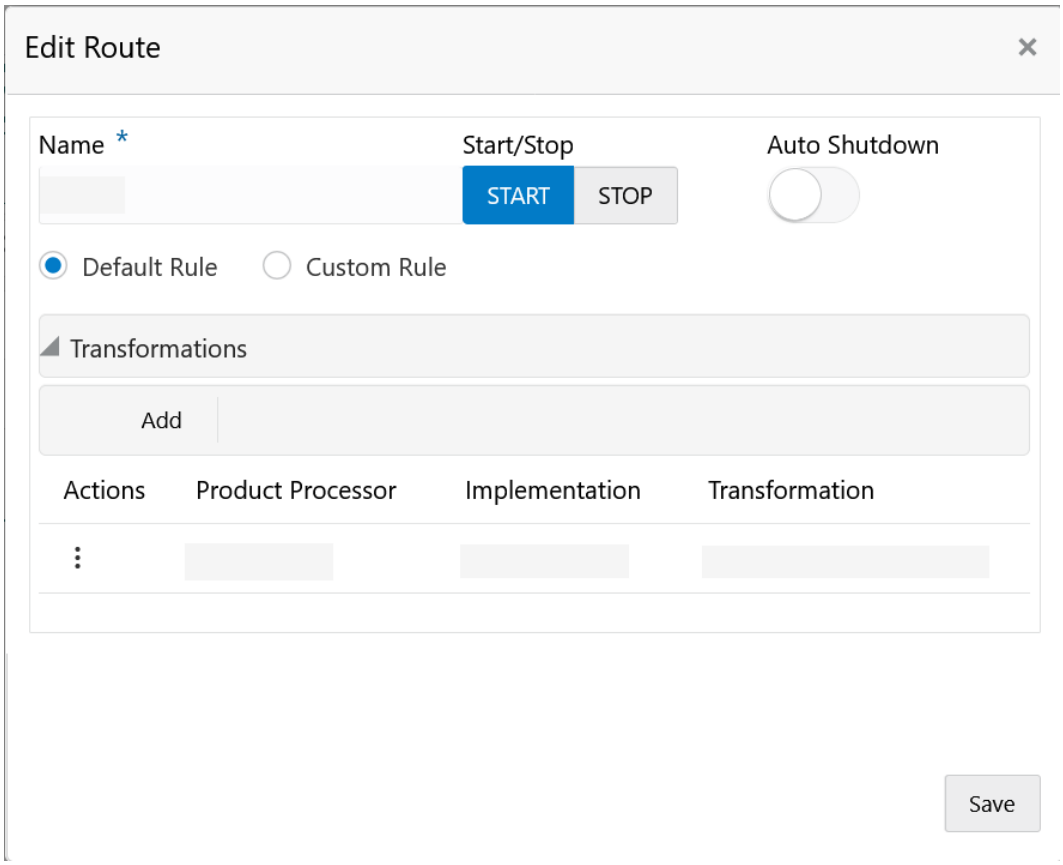
Edit Route

The user can modify the routing details.

9. On **Routing** screen, click **Operation menu** (3 dots button), and click **Edit**.

   The **Edit Route** screen displays.

Figure 8-6 Edit Route



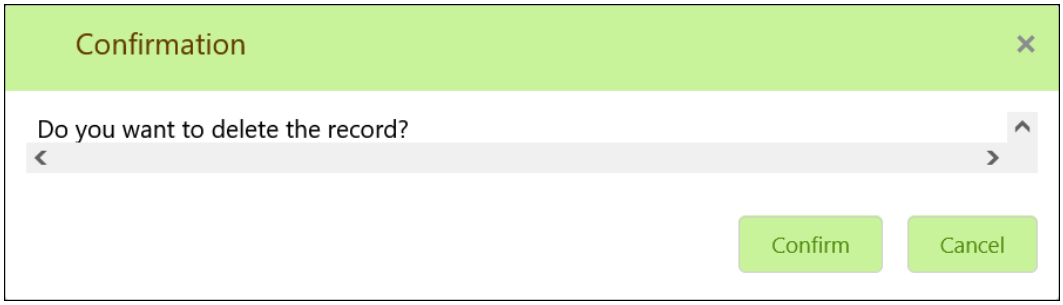
Delete Route

The user can delete the routing details.

10. On **Routing** screen, click **Operation menu** (3 dots button), and click **Delete**.

    The **Confirmation** screen displays.

Figure 8-7 Confirmation - Delete



11. Click **Confirm** to delete the selected routing.

Routing - Configuration

12. On **Routing** screen, click **Operation menu** (3 dots button), and click **Configuration**.

    The **Configuration** screen displays.

> **Note:**
>
> Refer to Configuration topic for screen and field description.

Routing - Request Audit

13. On **Routing** screen, click **Operation menu** (3 dots button), and click **Request Audit**.

    The **Request Audit** screen displays.

> **Note:**
>
> Refer to Request Audit topic for screen and field description.

9
Request Audit

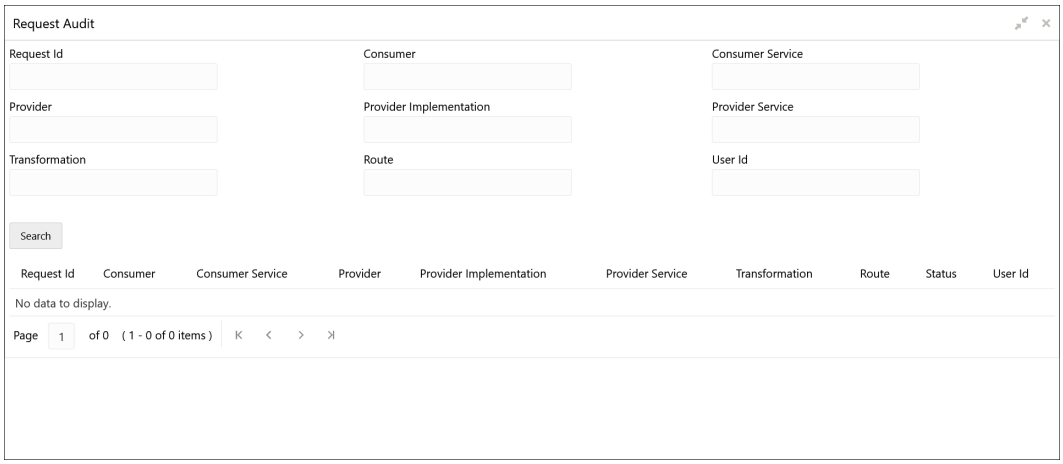
This topic describes the systematic instructions to check the audit log in Oracle Banking Routing Hub.

Specify **User ID** and **Password**, and login to **Home** screen.

1. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing Hub**.
2. Under **Routing Hub**, click **Request Audit**.

   The **Request Audit** screen displays.

Figure 9-1 Request Audit



3. Specify the fields on **Request Audit** screen.

> **Note:**
>
> The fields, which are marked with an asterisk, are mandatory.

For more information on fields, refer to the field description table.

Table 9-1 Request Audit - Field Description

Field	Description
Request ID	Specify the request ID.
Consumer	Specify the consumer.
Consumer Service	Specify the consumer service.
Provider	Specify the provider.

Table 9-1 (Cont.) Request Audit - Field Description

Field	Description
Provider Implementation	Specify the provider implementation.
Provider Service	Specify the provider service.
Transformation	Specify the transformation name.
Route	Specify the route.
User ID	Specify the user ID.

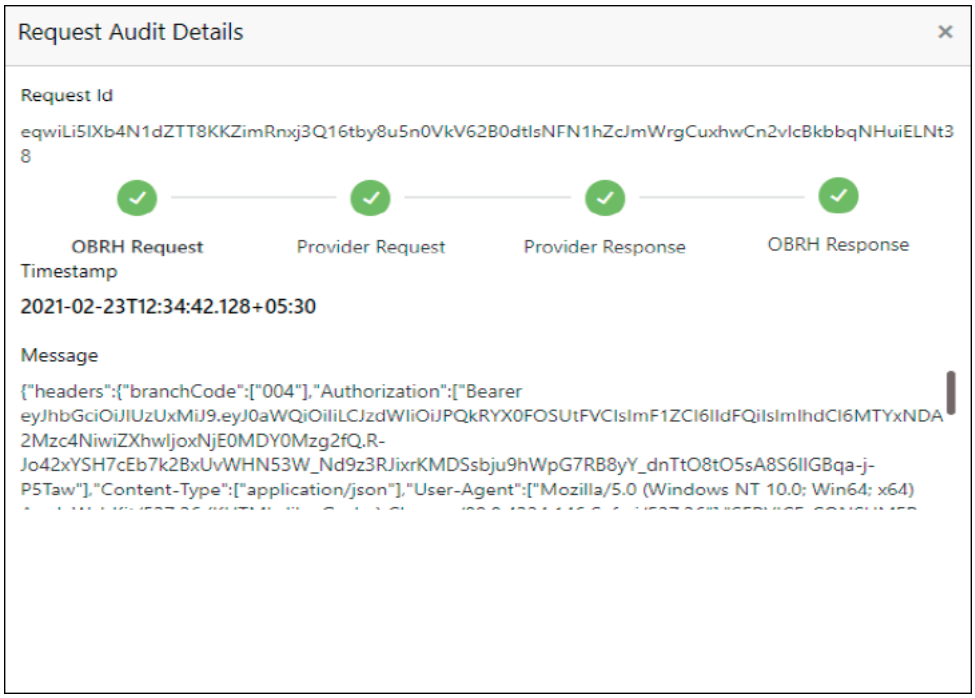
4. Click **Search**.

   The list of request ID's displays with relevant details.

5. Click on the **Request ID** to view the step by step execution of request audit details.

   The **Request Audit Details** screen displays.

Figure 9-2 Request Audit Details



For more information on fields, refer to the field description table.

Table 9-2 Request Audit Details - Field Description

Field	Description
Request ID	Displays the selected request ID.
OBRH Request	Displays the status of Routing Hub request.
Provider Request	Displays the status of provider request.
Provider Response	Displays the status of provider response.

Table 9-2 (Cont.) Request Audit Details - Field Description

Field	Description
OBRH Response	Displays the status of Routing Hub response.
Timestamp	Displays the date and time.
Message	Displays the message.

10 Chaining

This topic provides the information about chaining of the transformation.

The end-user can define the sequence of transformations for each routing in which the request needs to be processed.

Chaining can be achieved by using the snapshot list. The snapshot list stores the response body and response headers whenever the transformation is processed. Therefore, the end-user can access the response body or headers of all processed transformations at any stage.

Syntax: $snapshot.get(index).body or $snapshot.get(index).headers

> **Note:**
>
> $body and $headers refers to the response body and headers of previous step.

Figure 10-1 Chaining

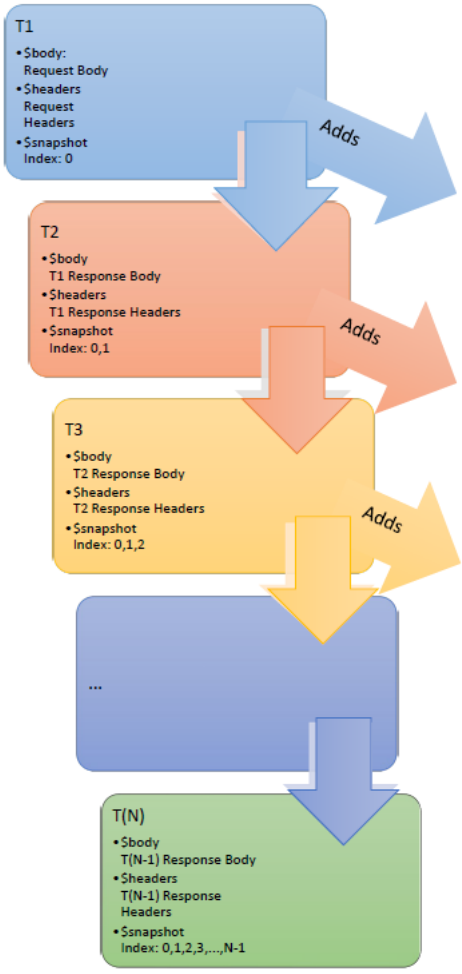


Table 10-1 Snapshot List

Index	Body	Headers
1	Request Body	Request Headers
2	T1 Response Body	T1 Response Headers
3	T2 Response Body	T2 Response Headers
4	T3 Response Body	T3 Response Headers
…	...	...
N	T(N-1) Response Body	T(N-1) Response Headers

11

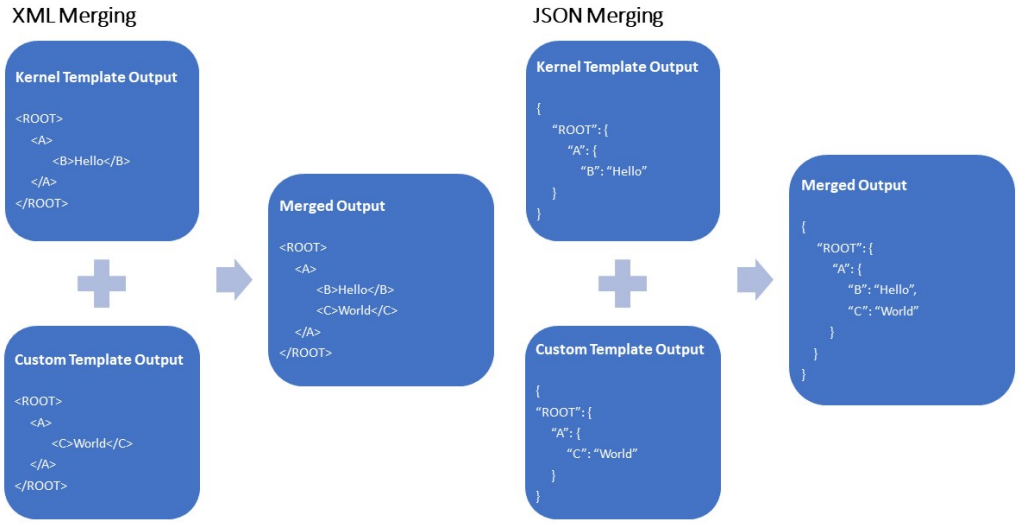
Extensibility

Extensibility in Routing Hub refers to template extensibility and is achieved by specifying the extended templates for request and response kernel transformation templates. And as part of extensibility, Routing Hub merges the output of kernel template and custom template in terms of JSON / XML merging.

In case of request, Routing Hub will send the merged output as request payload to provider.

In case of response, Routing Hub will return the merged output as response back to consumer

Figure 11-1 Extensibility - Example



12

Audit Purging / Archiving

Purging/Archiving of audit data is done on the basis of retention policy.

This process uses plato-batch-server for Job execution.

The following steps are required to schedule purging/archiving job (routingHubAuditRetentionJob) once cmc-obrh-services and plato-batch-server is UP and RUNNING:

1. On **Home** screen, click **Task Management**. Under **Task Management** menu, click **Configure Tasks**.
2. Select **Schedule** option.
3. Select **Task Name** as routingHubAuditRetentionJob and **Task Trigger Name** will be generated automatically.
4. Specify the CRON expression to daily EOD.

In order to resolve table space issue of Audit table (CMC_RH_AUDIT_EVENT_LOG), Database Management Team has to configure database job which should be triggered after routingHubAuditRetentionJob. This database job can be redefining the table (DBMS_REDEFINITION) after purging/archiving is done or other approach. So, the unused LOB segment space can be released. And in order to resolve table space issue of Audit history table (CMC_RH_AUDIT_EVENT_LOG_HISTORY), Database Management Team has to configure database job to truncate table periodically basis.

13
Multipart Request

This topic provides the sample template for the multipart request

Example 13-1 Multipart Request

```
[
    {
      "key": "file",
      "type": "FILE",
      "value": "$body.files.get(0).file"
    },
    {
      "key": "name",
      "type": "TEXT",
      "content": "$body.name"
    }
]
```

14
Dashboard

This topic provides information about dashboard widgets.

Routing Health Indicator Widget

The user can view the metric information Successful calls vs Failed calls ratio of each routing.

> **Note:**
>
> For failed calls, refer to the calls that are failed due to timeout issue.

Figure 14-1 Routing Health Indicator Widget

15
Transformation Type

This topic provides the information about the transformation types.

Velocity

Velocity is a Java-based template engine. It is used to generate XML files, SQL, PostScript, and most other text-based formats.

> **Note:**
>
> In Routing Hub, velocity is used to generate JSON and XML files.

- Using **$body**, user can access request/response body.
  Syntax: $body.fieldName

  Example: $body.branchCode
- Using **$headers**, user can access request/response headers.
  Syntax: $headers["fieldName"][0]

  Example: $headers["branchCode"][0]
- Using **$bodyAsString**, user can access response body as string.
  Syntax: $bodyAsString
- Below are some available extension methods:
  - Date Conversion
    Syntax: $dateUtil.convert(inputDate, fromPattern, toPattern)

    Parameters:

    * inputDate - String
    * fromPattern - String
    * toPattern - String

    Returns: String

    Refer to https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html for different patterns
  - Default Value
    Syntax: $custom.defaultValue(inputValue, defaultValue)

    Parameters:

    * inputValue - Object
    * defaultValue - String

    Returns: Object
  - Null Check

Syntax: $custom.isNull(inputValue)

Parameters:

* inputValue - Object

Returns: Boolean

- Random Number
  Syntax: $mathUtil.getRandom()

  Returns: Object of Random class (java.util.Random)

- Xml Tool
  Syntax: $xml.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/XmlTool.html

- Date Tool
  Syntax: $date.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/DateTool.html

- Json Tool
  Syntax: $json.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/JsonTool.html

- Math Tool
  Syntax: $math.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/MathTool.html

- Number Tool
  Syntax: $number.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/NumberTool.html

- Escape Tool
  Syntax: $esc.methodName()

  Refer to https://velocity.apache.org/tools/3.1/apidocs/org/apache/velocity/tools/generic/EscapeTool.html

- Serialization of object into its equivalent Json representation
  Syntax: $custom.toJson(src)

  Parameters:

  * src - Object

  Returns: String

- Get additional field's value based on fieldname
  Syntax: $custom.getFieldValueById(jsonString, fieldname)

  Parameters:

  * jsonString – String
  * fieldname - String

  Returns: String

- Get list of additional fields based on fieldname prefix
  Syntax: $custom.getAdditionalFieldSetByType(jsonString,prefixval)

  Parameters:

  * jsonString - String
  * prefixval - String

  Returns: String

- If issue occurred with hyphen in velocity template of Request or Response Transformation, then use get method.
  Example:

```
<FCUBS_BODY>
   <Customer-IO>
     <CUSTNO>003942</CUSTNO>
   </Customer-IO>
</FCUBS_BODY>
```

  If `$in.FCUBS_BODY.Customer-IO.CUSTNO` does not work ,
  use `$in.FCUBS_BODY.get("Customer-IO").CUSTNO` to get customer number.

XSLT

XSLT is a language for transforming XML documents into other XML documents, or other formats such as HTML for web pages, plain text or XSL formatting objects, which may subsequently be converted to other formats, such as PDF, PostScript and PNG.

> **Note:**
>
> In Routing Hub, XSLT is used to transform arbitrary XML to JSON.

JSLT

JSLT is a complete query and transformation language for JSON.

16

Oracle Banking Routing Hub Integration Specification

This topic provides information about Oracle Banking Routing Hub Integration Specification.

- Token Generation
  This topic provide information about the Token Generation.
- Synchronous Dispatch API Specification
  This topic provide information about the Synchronous Dispatch API Specification.
- Asynchronous Dispatch API Specification
  This topic provide information about the Asynchronous Dispatch API Specification.
- Asynchronous Dispatch Response API Specification
  This topic provide information about the Asynchronous Dispatch Response API Specification.

16.1 Token Generation

This topic provide information about the Token Generation.

PlatoJWTAuth endpoint Signature -

- **Path**: /platojwtauth
- **Headers**:
  - appId : SECSRV001
  - Content-Type : application/json
- **Request Body**:

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

  - Username and password is base64 encoding of plaintext.
- **Response Body**:

```
{
  "token": "",
  "userAlreadyLoggedIn": "Y",
  "expires_in": 3180,
  "home_entity_id": "DEFAULTENTITY",
  "multi_entity_admin": "N",
  "multi_entity_admin_locale": ""
}
```

Refer the below sample screenshots:

Figure 16-1 Headers

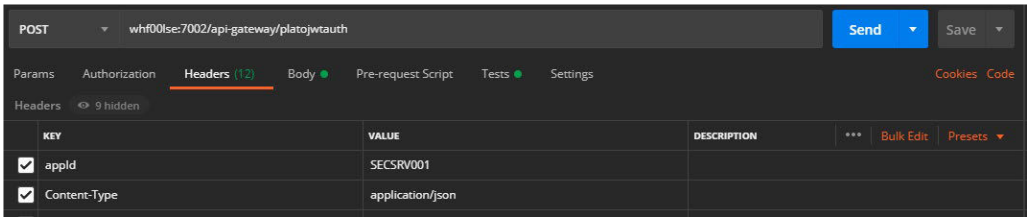


Figure 16-2 Request Payload

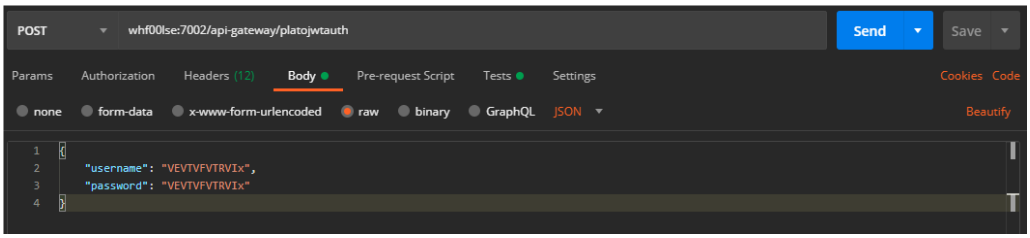


Figure 16-3 Response Payload

16.2 Synchronous Dispatch API Specification

This topic provide information about the Synchronous Dispatch API Specification.

Dispatch endpoint is the single entry-point for invoking the routes configured in Oracle Banking Routing Hub for services of a Service Consumer.

Dispatch Endpoint Signature -

- **Method**: POST / GET
- **Path**: /route/dispatch
- **Headers**:
  - appId : CMNCORE
  - entityId : DEFAULTENTITY
  - userId : <user id>
  - branchCode : <branch code>

  - Authorization : Bearer <Token>
  - SERVICE-CONSUMER : <name of service consumer>
  - SERVICE-CONSUMER-SERVICE : <name of service consumer service>
- **Request Body**:
  - Any valid JSON payload which shall act as input to the transformation template in request transformer.
- **Response Body**:

```
{
 "data": {}
 "messages": {}
}
```

  - If the route invocation succeeds, data JSON member would contain the transformed (optional) response of the provided service. If it is a provided web service and no response transformation template is supplied, XML Soap Body of response would be converted into JSON object and sent in data JSON member.
  - If the route invocation fails due to misconfigured route or connection time out to ServiceProviderImpl or some other reason, relevant error messages would be sent in messages JSON member. In that case, data JSON member would be null or empty.

Examples: Refer the below screenshots of route dispatch for Service-Consumer **Oracle_Service_Consumer** and Service-Consumer Service **List_Facilities**.

Figure 16-4 Headers

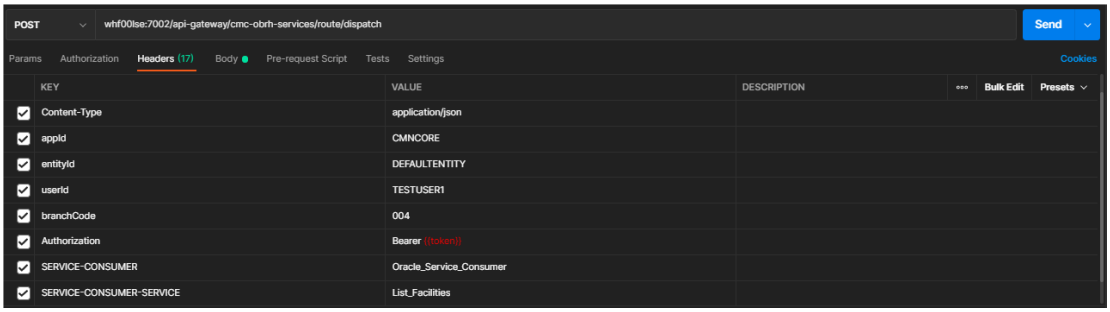


Figure 16-5 Request Payload

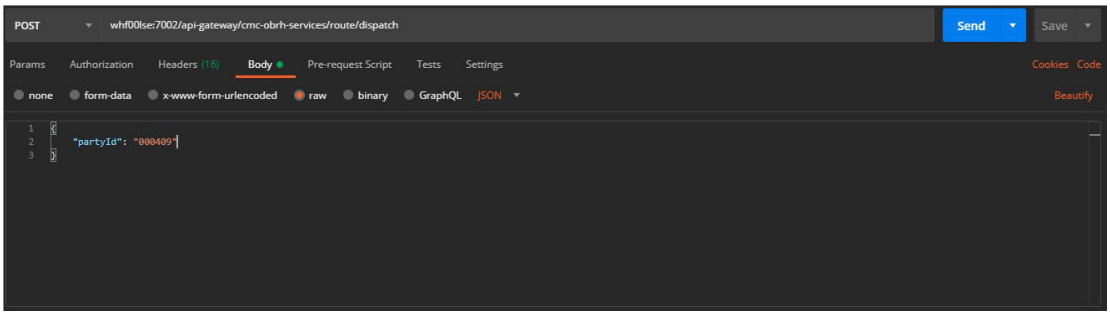


Figure 16-6 Response Payload on Successful Dispatch

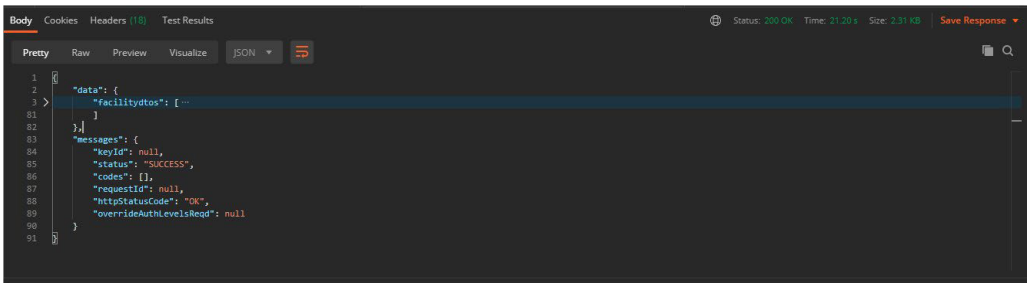
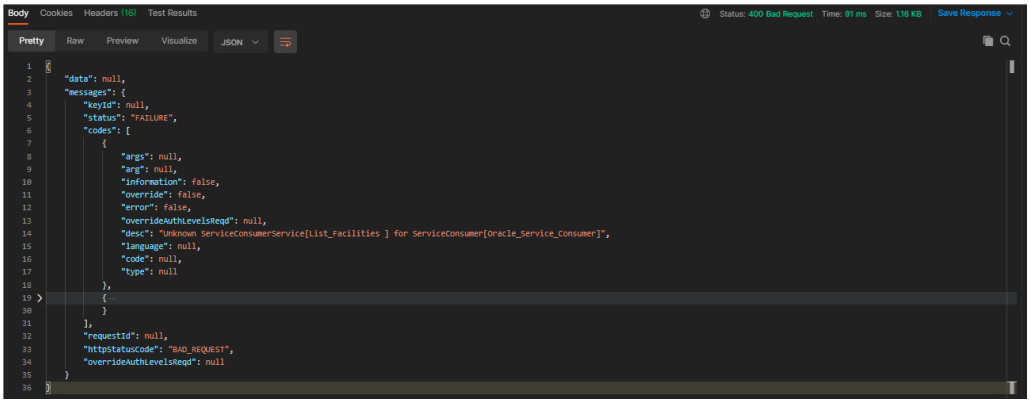


Figure 16-7 Response Payload on Failed Dispatch



16.3 Asynchronous Dispatch API Specification

This topic provide information about the Asynchronous Dispatch API Specification.

Dispatch endpoint is the single entry-point for invoking the routes configured in Oracle Banking Routing Hub for services of a Service Consumer.

Dispatch endpoint Signature -

- **Path**: /route/dispatch
- **Query Params**:
  - isAsync : true
- **Headers**:
  - appId : CMNCORE
  - entityId : DEFAULTENTITY
  - userId : <user id>
  - branchCode : <branch code>
  - Authorization : Bearer <Token>
  - SERVICE-CONSUMER : <name of service consumer>
  - SERVICE-CONSUMER-SERVICE : <name of service consumer service>

- **Request Body**:
  - Any valid JSON payload which shall act as input to the transformation template in request transformer.
- **Response Body**:

```
{
 "data": { "correlationId" : "" },
 "messages": {}
}
```

Example: Refer the below screenshots of route dispatch for Service-Consumer **Oracle_Service_Consumer** and Service-Consumer Service **List_Facilities**.

Figure 16-8 Query Params

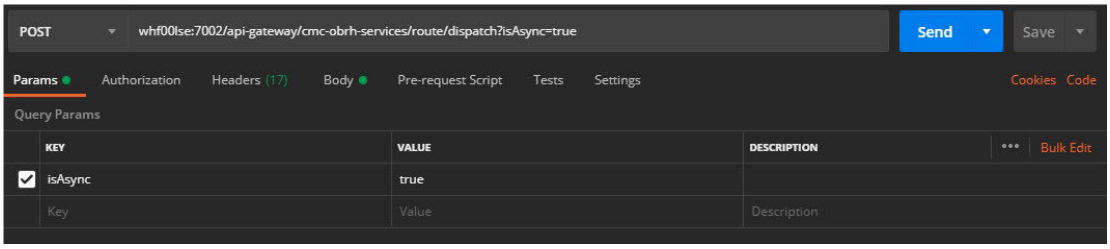


Figure 16-9 Headers

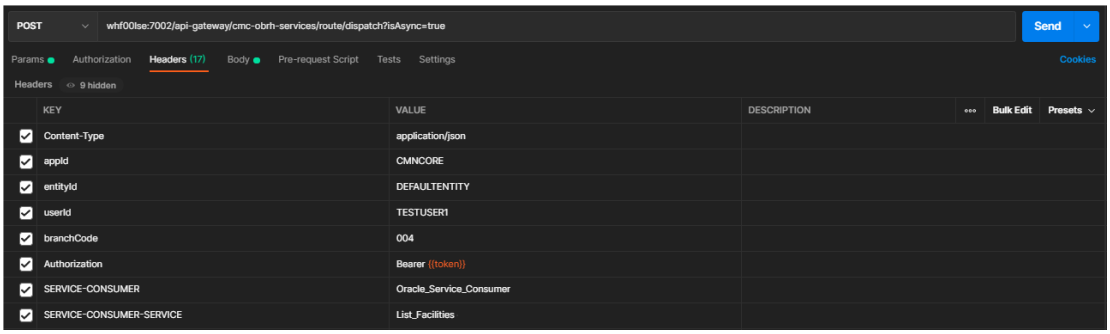


Figure 16-10 Request Payload

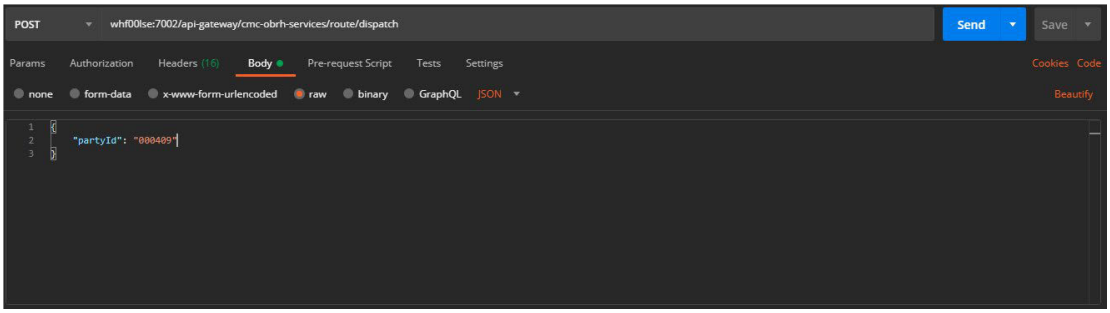
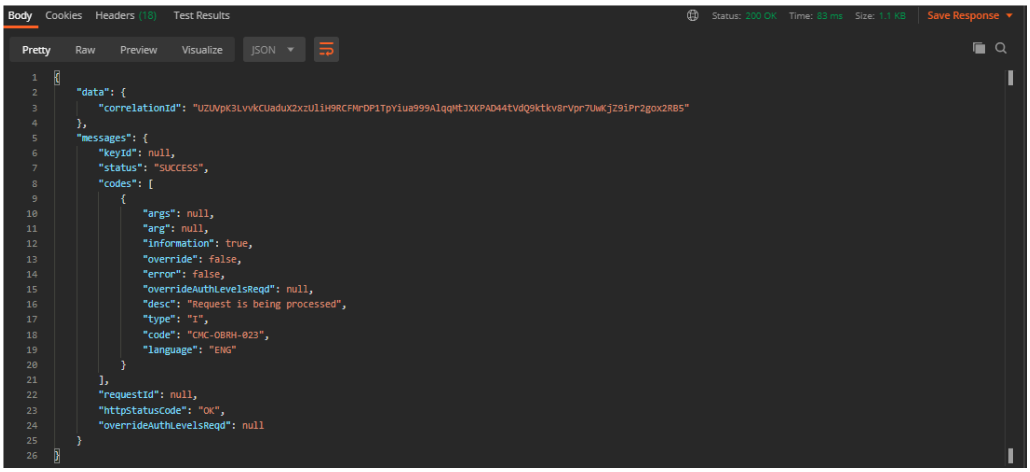


Figure 16-11 Response Payload



16.4 Asynchronous Dispatch Response API Specification

This topic provide information about the Asynchronous Dispatch Response API Specification.

Response endpoint Signature -

- **Path**: /route/dispatchResponse/{Correlation-Id}
  - Correlation-Id will be coming from the response of dispatch endpoint.
- **Headers**:
  - appId : CMNCORE
  - entityId : DEFAULTENTITY
  - userId : <user id>
  - branchCode : <branch code>
  - Authorization : Bearer <Token>
- **Response Body**:

```
{
 "data": {}
 "messages": {}
}
```

  - If the route invocation succeeds, data JSON member would contain the transformed (optional) response of the provided service. If it's a provided web service and no response transformation template is supplied, XML Soap Body of response would be converted into JSON object and sent in data JSON member.
  - If the route invocation fails due to misconfigured route or connection time out to ServiceProviderImpl or some other reason, relevant error messages would be sent in messages JSON member. In that case, data JSON member would be null or empty.

Example: Refer the below screenshots of route dispatch for Service-Consumer **Oracle_Service_Consumer** and Service-Consumer Service **List_Facilities**.

Figure 16-12 Header

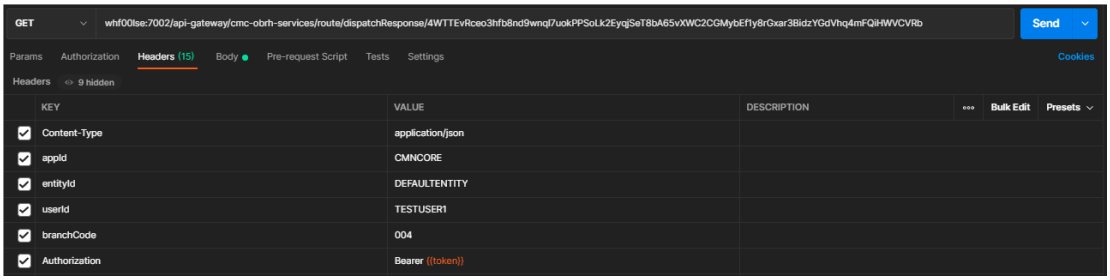


Figure 16-13 Response Payload when request is still processing

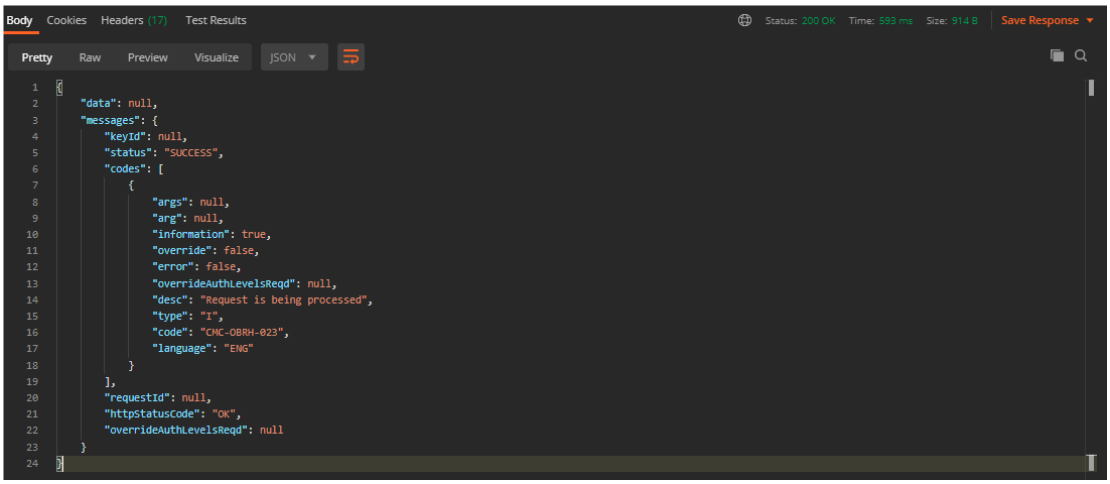


Figure 16-14 Response Payload when request is processed (on Successful Dispatch)

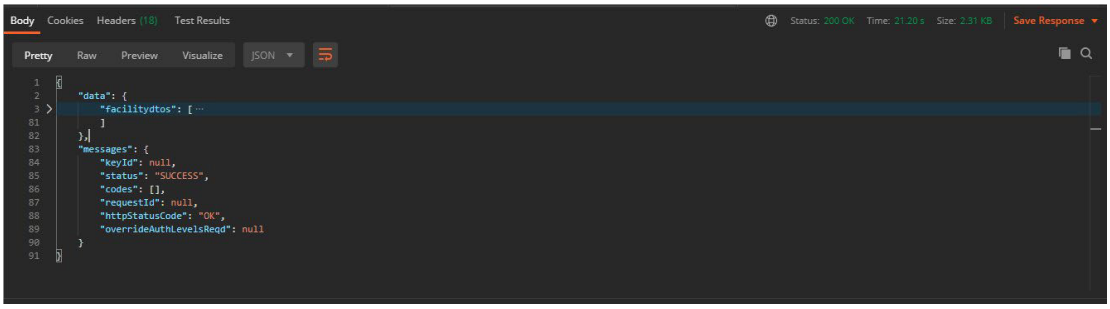
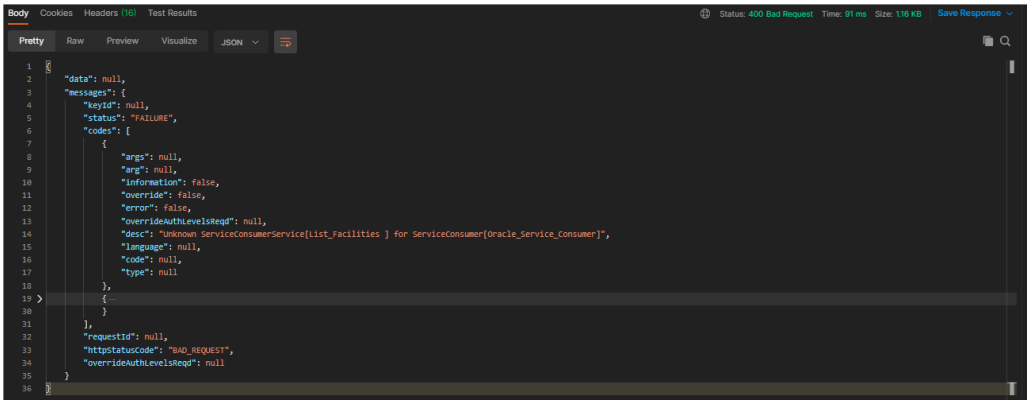


Figure 16-15 Response Payload when request is processed (on Failed Dispatch)



17

Oracle Banking Routing Hub VM Arguments

This topic provides information about Oracle Banking Routing Hub VM arguments.

Common Core Managed Server

-Dcmc-obrh-services.server.port=<SERVER_PORT>
-Dobrh.db.jndi=<CMNCORE_JNDI>
-Dcmc-obrh-services.oic.oauth.scope=<OIC_OAUTH_SCOPE>
-Dcmc-obrh-services.oic.secretStore.url=<OIC_SECRET_STORE_URL>
-Dcmc-obrh-services.oic.idcs.url=<OIC_IDCS_URL>
-Dcmc-obrh-services.audit.retention.days=<AUDIT_RETENTION_POLICY_DAYS>
-Dcmc-obrh-services.audit.retention.archival=<AUDIT_RETENTION_POLICY_APPROACH> (Y for archiving and N for purging)

To receive routing failure mail notification via plato-alerts-management-service, then set the following property to true, (Default value is false)

-Dobrh.alerts.enabled=<ALERTS_ENABLED>

In order to change the behavior of auditing, then set the below property from predefined values (DEFAULT / KAFKA / LOG / OFF), (Default value is DEFAULT)

-Dobrh.audit.type=<AUDIT_TYPE>

> **Note:**
>
> For KAFKA option, cmc-obrh-kafka-consumer service needs to be deployed.

In order to fail Oracle Banking Routing Hub request if provider request fails with specific status code, then set the below property to comma-separated status codes

-Dobrh.provider.exception.statuscodes=<STATUS_CODES>

In order to overwrite the customization that is not part of configuration JSON,

-Dobrh.import.overwrite=<IMPORT_OVERWRITE> (Default value is false)

To use Custom Keystore and Truststore for HTTPS scheme,

-Dobrh.keystore.password.encoded=<IS_PASSWORD_ENCODED> (true, if password is base 64 encoded)
-Dobrh.truststore.path=<TRUSTSTORE_PATH>
-Dobrh.truststore.password=<TRUSTSTORE_PASSWORD>
-Dobrh.usekeystore=<USE_KEYSTORE> (true, if keystore is required along with truststore)
-Dobrh.keystore.path=<KEYSTORE_PATH> -
Dobrh.keystore.password=<KEYSTORE_PASSWORD>
-Dobrh.keystore.alias=<KEYSTORE_ALIAS>
-Dobrh.keystore.aliaspassword=<KEYSTORE_ALIAS_PASSWORD>
-Dobrh.ssl.protocol=<SSL_PROTOCOL> (Default value is TLS)

To perform the tomcat deployment,

`-Dobrh.server.isJavaEE=false (mandatory)`
`-Dobrh.taskexecutor.corepoolsize=<CORE_POOLSIZE>` (default is 50) (optional)
`-Dobrh.taskexecutor.maxpoolsize=<MAX_POOLSIZE>` (default is 50) (optional)
`-Dobrh.taskexecutor.queuecapacity=<QUEUE_CAPACITY>` (default is 100) (optional)

To set Proxy settings for HTTPS,

```
-Dhttps.proxyHost=<PROXY_HOST_NAME>
-Dhttps.proxyPort=<PROXY_PORT>
-Dhttps.nonProxyHosts=<NON_PROXY_HOST_LIST>
-Dhttp.nonProxyHosts=<NON_PROXY_HOST_LIST>
```

> **Note:**
>
> As per the Java Networking documentation, HTTPS protocol handler uses the same as the http handler (http.nonProxyHosts). in case of Weblogic, http.nonProxyHosts do not work for some reason. So, use https non proxy host argument (https.nonProxyHosts).

To set logger level,

`- Dplato.service.logging.level=`

To do SSL based SOAP provider calls and if RoutingHub is deployed on weblogic environment,

`-DUseSunHttpHandler=true`

> **Note:**
>
> This property will enforce WebLogic Server to use SUN SSL implementation (javax package) rather than the WebLogic one.

Plato Core Managed Server

Oracle Banking Routing Hub is using Multipart for Import feature. By default, spring supports max 1MB file size and 10MB request size for Multipart.

To import bigger files,

`plato-api-gateway.multipart.max-file-size=<MAX_FILE_SIZE>` (default is 1MB)
`plato-api-gateway.multipart.max-request-size=<MAX_REQUEST_SIZE>` (default is 10MB)

> **Note:**
>
> -1 for no size constraint
> **Example**,
>
> plato-api-gateway.multipart.max-file-size=-1
> plato-api-gateway.multipart.max-request-size=-1

CMC-OBRH-KAFKA-CONSUMER

```
-Dcmc-obrh-kafka-consumer.server.port=<SERVER_PORT>
```

Index