

# Oracle® Banking Microservices Architecture

## Routing Hub Configuration User Guide



Release 14.6.1.0.0

F61119-01

August 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2018, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Purpose	v
Audience	v
Acronyms and Abbreviations	v
List of Topics	v
Symbols and Icons	vi

## 1 Introduction

---

## 2 Configuration

---

## 3 Service Consumers

---

## 4 Service Providers

---

## 5 Implementation

---

## 6 Consumer Services

---

## 7 Transformation

---

## 8 Routing

---

## 9 Request Audit

---

10	Chaining	
11	Extensibility	
12	Audit Purging / Archiving	
13	Multipart Request	
14	Dashboard	
15	Transformation Type	
16	Oracle Banking Routing Hub Integration Specification	
16.1	Token Generation	16-1
16.2	Synchronous Dispatch API Specification	16-2
16.3	Asynchronous Dispatch API Specification	16-4
16.4	Asynchronous Dispatch Response API Specification	16-6
17	Oracle Banking Routing Hub VM Arguments	
	Index	

# Preface

- [Purpose](#)
- [Audience](#)
- [Acronyms and Abbreviations](#)
- [List of Topics](#)
- [Symbols and Icons](#)

## 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
<a href="#">Introduction</a>	This topic provides the general information about the guide.
<a href="#">Configuration</a>	This topic provides the information about the Configuration.
<a href="#">Service Consumer</a>	This topic provides the information about the Service Consumer.

Table 2 (Cont.) List of Topics

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

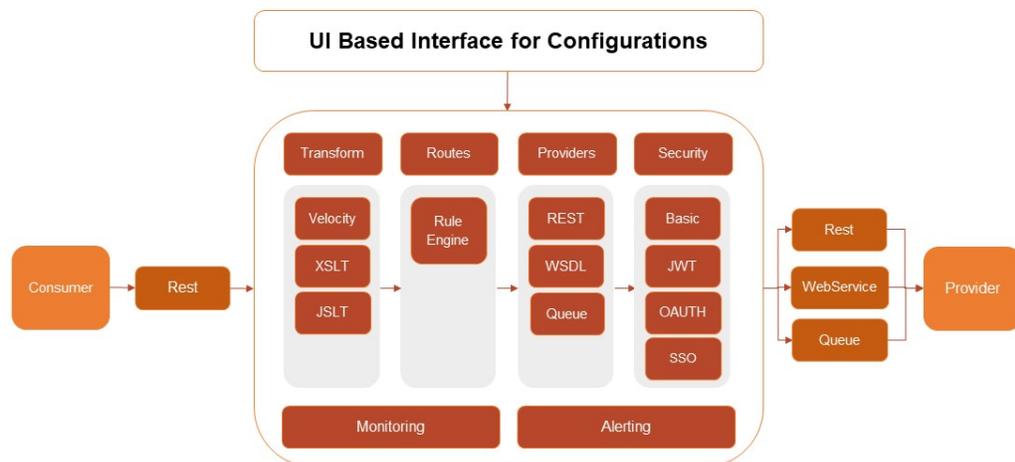
Consumer application does not need to know the points given below.

- 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.

End-user can configure the properties w.r.t. monitoring, alert and export.

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**

The screenshot shows a web-based configuration interface titled "Configuration". It is divided into four main sections, each with a collapse/expand icon on the left:

- Monitoring:** Contains "Window Type" with radio buttons for "Count" (selected) and "Time", and "Window Size" with a numeric input field set to "100" and up/down arrows.
- Alert:** Contains "Minimum number of calls" with a numeric input field set to "100" and up/down arrows, and "Failure rate threshold" with a percentage input field set to "50%" and up/down arrows.
- Email Alert:** Contains "Email Addresses" with a text input field containing "ddd".
- Export:** Contains "Mark data as factory shipped" with a toggle switch currently turned on.

At the bottom right of the form, there are three buttons: "Clear", "Reset", and "Save".

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
<b>Window Type</b>	Select the type of the window. The available options are: <ul style="list-style-type: none"> <li><b>Count</b>: The count-based sliding window aggregates the outcome of the last N calls (<b>Window Size</b>).</li> <li><b>Time</b>: The time-based sliding window aggregates the outcome of the calls of the last N seconds (<b>Window Size</b>).</li> </ul>
<b>Window Size</b>	Specify the window size to record the outcome of the calls when the circuit breaker is closed. <ul style="list-style-type: none"> <li>For <b>Count</b> window type, The window size is N calls.</li> <li>For <b>Time</b> window type, The window size has N seconds.</li> </ul>
<b>Minimum number of calls</b>	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.
<b>Failure rate threshold</b>	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.
<b>Email Addresses</b>	Specify the E-mail address. The user can use semi-colon to add more email addresses. Once the failure rate crosses the <b>Failure rate threshold</b> , a mail is sent to the end-user about the event.
<b>Mark data as factory shipped</b>	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
<b>Window Type</b>	Count
<b>Window Size</b>	20
<b>Minimum number of calls</b>	10
<b>Failure rate Threshold</b>	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.

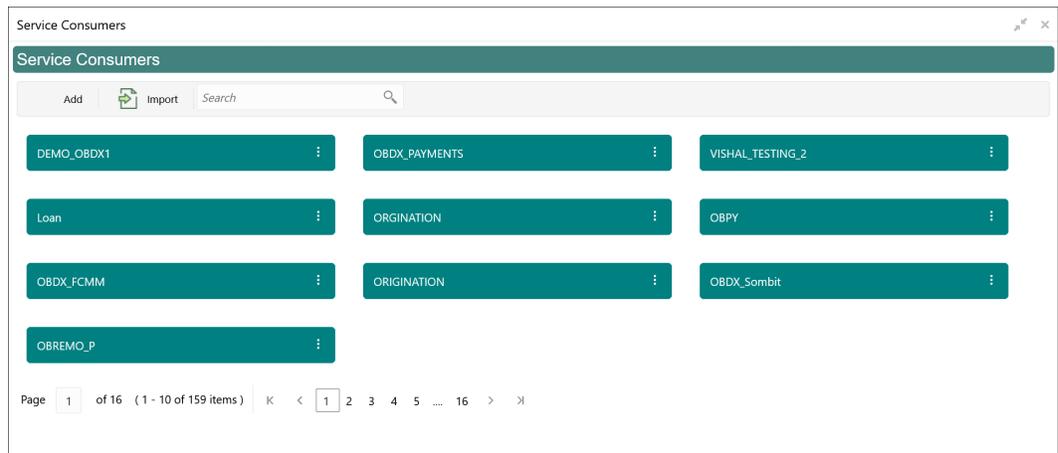
The **Service Consumer** is an Oracle product that invokes Oracle Banking Routing Hub API. Oracle Banking Routing Hub analyses, evaluates the destination product processor, and transforms the data into a format of the same. It comprises the source and destination integration details.

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**

- 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
<b>Name</b>	Specify the unique service consumer name. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>Enter 0 to maximum of 255 characters.</li> <li>No numeric value at beginning and no space allowed.</li> </ul> </div>

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

Field	Description
<b>Add</b>	To add, refer to step 5. Select the group from the drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>Group</b></li> <li>• <b>Variable</b></li> </ul>
<b>Group</b>	Select the group from the drop-down list.
<b>Action</b>	Displays the action. The user can edit or delete the header.
<b>Name</b>	Displays the name of the header.
<b>Value</b>	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.

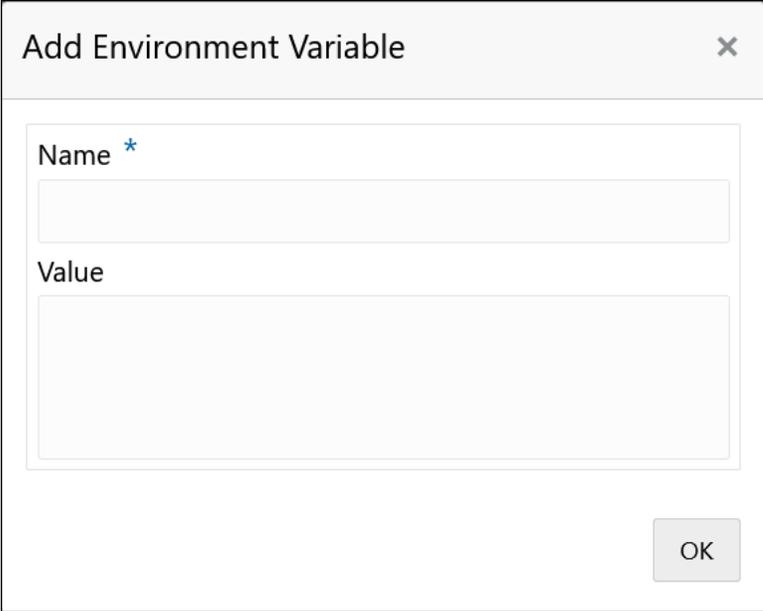
For more information on fields, refer to the field description table.

**Table 3-2 Add Environment Group - Field Description**

Field	Description
<b>Name</b>	Specify the name of the environment group.  <div style="border: 1px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• No numeric value at beginning and no space allowed.</li> </ul> </div>

- 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.

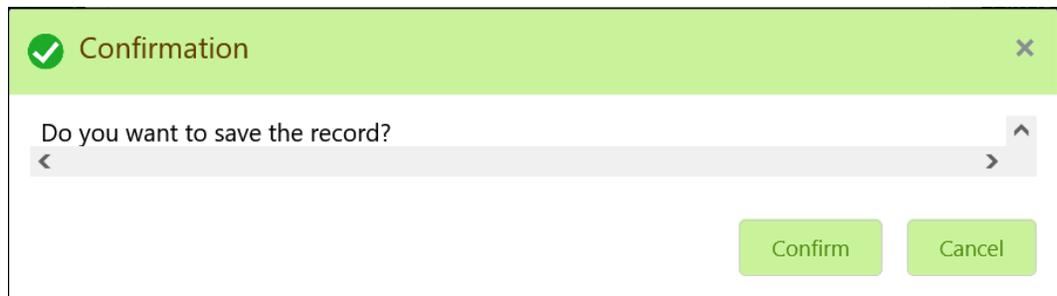
For more information on fields, refer to the field description table below.

**Table 3-3 Add Environment Variable - Field Description**

Field	Description
<b>Name</b>	Specify the name of the environment variable.  <div style="border: 1px solid #0070c0; padding: 5px; background-color: #e6f2ff;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• No numeric value at beginning and no space allowed.</li> </ul> </div>
<b>Value</b>	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**

Import Service Consumer

File \*

Select Extract

Name \*

Overwrite extended templates

Yes  No

Service Providers

Name

No data to display.

Import

- 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**

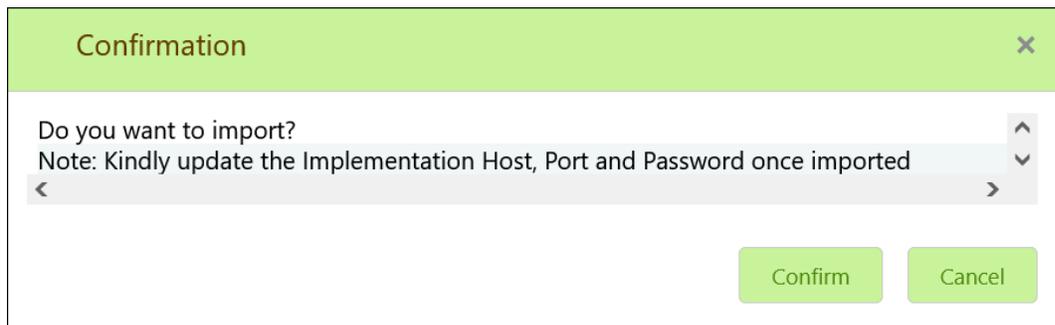
Field	Description
<b>File</b>	Select the file using <b>Select</b> .   <b>Note:</b> Allows only to select one file and accepts JSON and ZIP file.
<b>Extract</b>	Extracts the consumer name and service provider list from JSON file only and displays it in the respective elements.

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

Field	Description
<b>Name</b>	Specify the name of the service provider.  <div style="border: 1px solid #0070c0; padding: 5px; background-color: #e6f2ff;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>Name cannot be blank and required only for JSON file.</li> <li>Enter 0 to maximum of 255 characters.</li> <li>No numeric value at beginning and no space allowed.</li> </ul> </div>
<b>Overwrite extended templates</b>	Select the respective radio button to overwrite the extended templates. The available options are: <ul style="list-style-type: none"> <li><b>Yes</b> - This option overwrites the extended templates.</li> <li><b>No</b> - This option retains the existing extended templates.</li> </ul>
<b>Service Providers</b>	Displays the service provider details.
<b>Name</b>	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**

The screenshot shows a window titled "View Service Consumer". It contains a form with the following elements:

- A "Name" text input field.
- A section header "Environment Variables" with a small triangle icon to its left.
- A "Group" dropdown menu.
- A table with three columns: "Actions", "Name", and "Value".

The table contains two rows of data, each with a vertical ellipsis (three dots) in the "Actions" column and placeholder boxes in the "Name" and "Value" columns.

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

**Edit Service Consumer** [X]

Name \*

Environment Variables

Add ▾

Group \*

Select ▾

Actions	Name	Value
No data to display.		

Save

- Click **Save** to save the modified consumer details.  
The **Confirmation** screen displays.

Figure 3-10 Confirmation - Edit Service Consumer

**Confirmation** [X]

Do you want to save the record?

Confirm Cancel

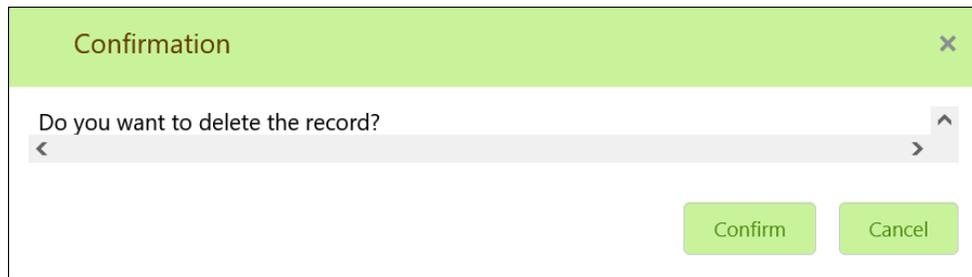
- Click **Confirm**.

#### Delete Service Consumer

The user can delete the Service Consumer.

- 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**

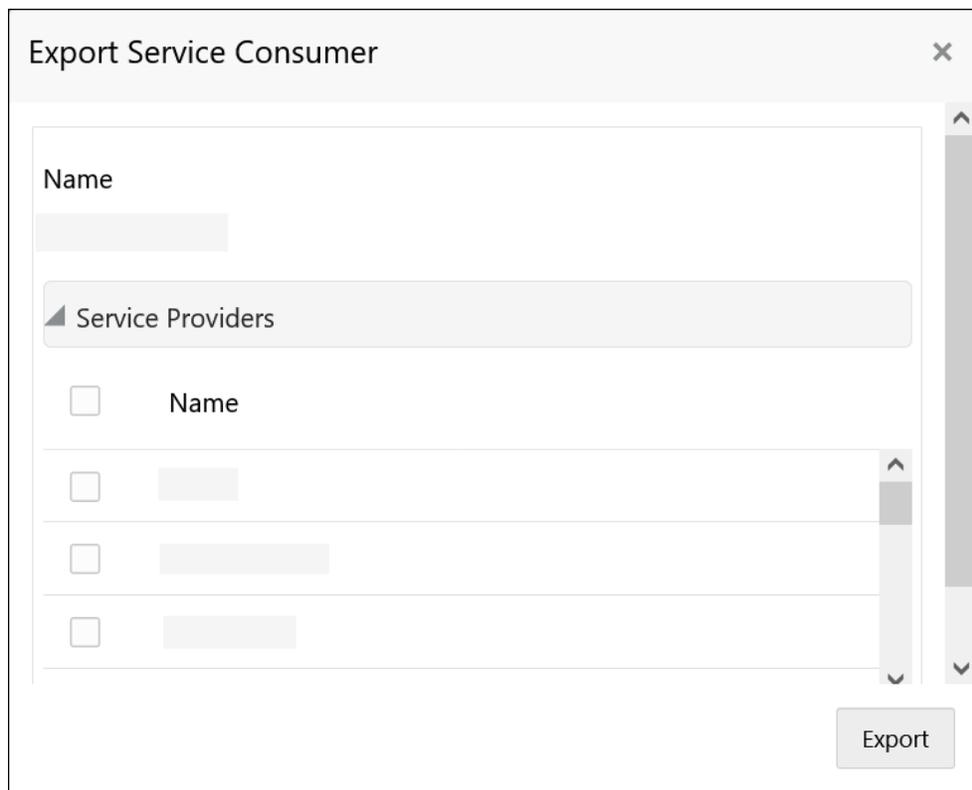
18. Click **Confirm** to delete the service consumer.

### 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**

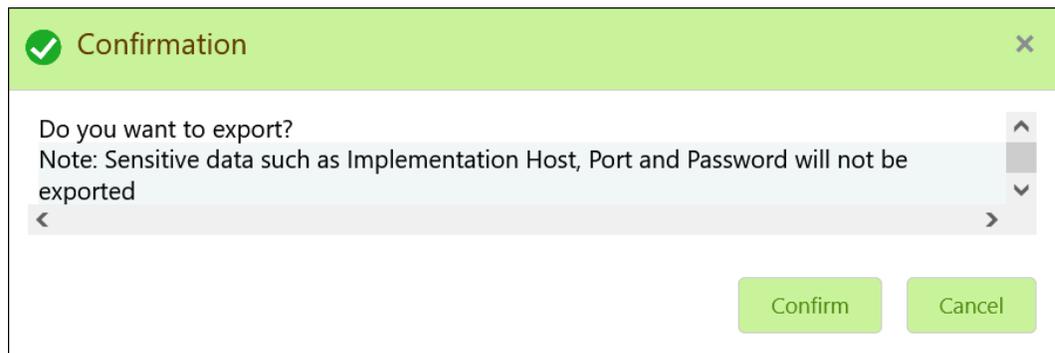
 **Note:**

- 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

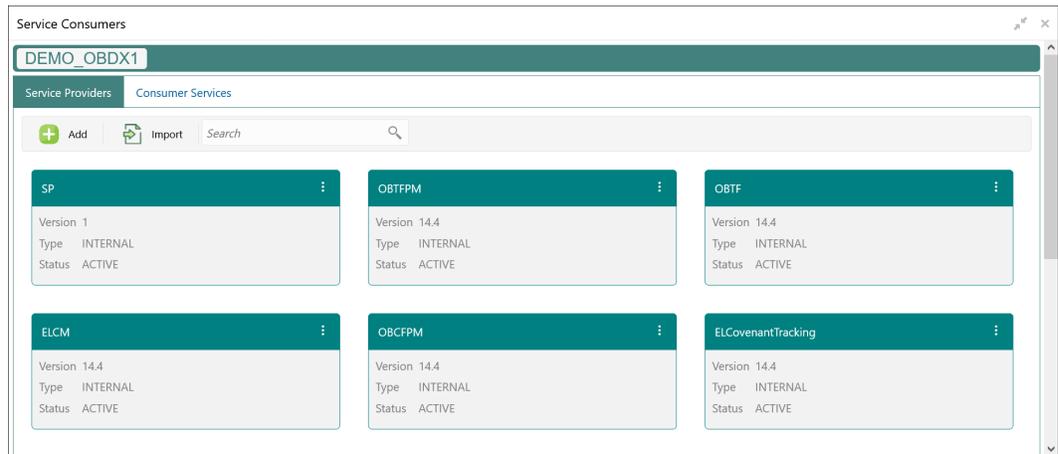
This topic describes the systematic instructions to configure the service providers.

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

Product Name \*

Type \*

Version \*

Active

Headers

+ Add

Actions	Name	Value
No data to display.		

Service

Type URL

WSDL Import

Service	Operation
No data to display.	

Save

- 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
<b>Product Name</b>	Specify the product name of the service provider.   <b>Note:</b> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• No numeric value at beginning and no space allowed.</li> </ul>
<b>Type</b>	Select the type of service provider from drop-down list The available options are: <ul style="list-style-type: none"> <li>• <b>INTERNAL</b>: Used for Oracle products</li> <li>• <b>EXTERNAL</b>: Used for non-Oracle products</li> </ul>
<b>Version</b>	Specify the provider version.   <b>Note:</b> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• Only numeric or decimal values are allowed.</li> </ul>
<b>Active</b>	Predefined values are Active / Inactive If provider is marked as inactive, then all related routes will be stopped.
<b>Add</b>	To add, refer to the below steps.
<b>Actions</b>	Displays the action. The user can edit or delete the header.
<b>Name</b>	Displays the name of the header.
<b>Value</b>	Displays the value of the header.
<b>Type</b>	Select the type of service from drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>WSDL</b></li> <li>• <b>SWAGGER</b></li> <li>• <b>OTHERS</b></li> </ul>
<b>Name</b>	Specify the name of the operation.   <b>Note:</b> <p>This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>

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

Field	Description
<b>Http Method</b>	<p>Select the HTTP method. The available options are:</p> <ul style="list-style-type: none"> <li>• <b>GET</b></li> <li>• <b>POST</b></li> <li>• <b>PUT</b></li> <li>• <b>PATCH</b></li> <li>• <b>DELETE</b></li> </ul> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>
<b>Endpoint</b>	<p>Specify the endpoint URL for the operation.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>
<b>URL</b>	<p>Specify the service URL of the file location.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>WSDL</b> and <b>SWAGGER</b>.</p>
<b>Import</b>	<p>Click <b>Import</b> to extract the service information from URL.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>WSDL</b> and <b>SWAGGER</b>.</p>
<b>Gateway Prefix</b>	<p>Gateway Prefix is context path of below formatted URL http://host:port/gateway-prefix/endpoint</p>

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

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

Import Service Provider

File \*

Select

Overwrite extended templates

Yes  No

Import

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

**Table 4-3 Import Service Provider - Field Description**

Field	Description
<b>File</b>	<p>Select the file using <b>Select</b> button.</p> <p> <b>Note:</b></p> <p>Allows only to select one file and accepts JSON and ZIP file.</p>
<b>Overwrite extended templates</b>	<p>Select the respective radio button to overwrite extended templates. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Yes</b> - This option overwrites the extended templates in configuration.</li> <li>• <b>No</b> - This option retains the existing extended templates in configuration.</li> </ul> <p> <b>Note:</b></p> <p>This field appears only if the ZIP File is selected.</p>

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**

The screenshot shows a window titled "View Service Provider" with a close button in the top right. The main content area displays the following details:

Product Name	Type
SP	INTERNAL
Version	Active
1	On

Below the details is a section titled "Headers" with a table that has three columns: "Actions", "Name", and "Value". The table content is empty, with the text "No data to display." below the headers.

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**

The screenshot shows a window titled "Edit Service Provider" with a close button in the top right. The main content area displays the following input fields:

- Product Name \***: Text input field containing "SP".
- Type \***: Dropdown menu showing "INTERNAL".
- Version \***: Text input field containing "1".
- Active**: Toggle switch currently turned on (blue).

Below the input fields is a section titled "Headers" with a table that has three columns: "Actions", "Name", and "Value". The table content is empty, with the text "No data to display." below the headers. There is an "Add" button below the table. A "Save" button is located in the bottom right corner of the window.

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

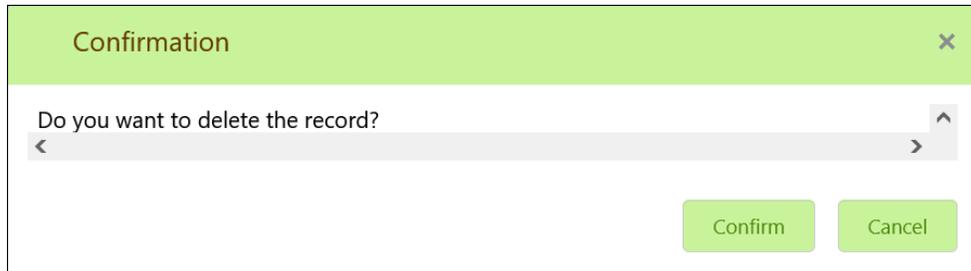
The **Confirmation** screen displays.

#### **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**



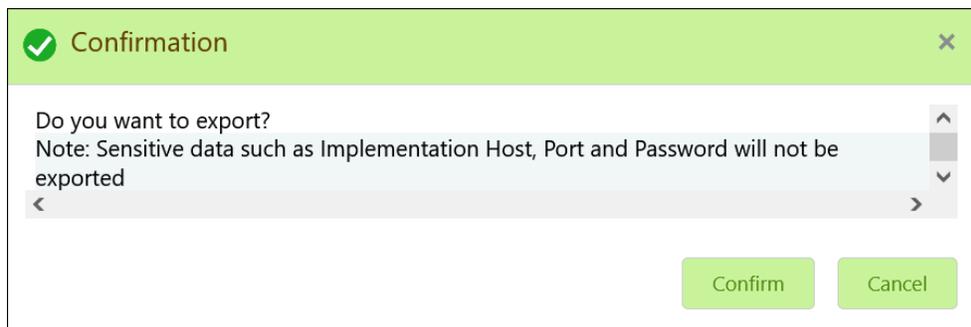
14. Click **Confirm** to delete the selected Service Provider.

#### **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.

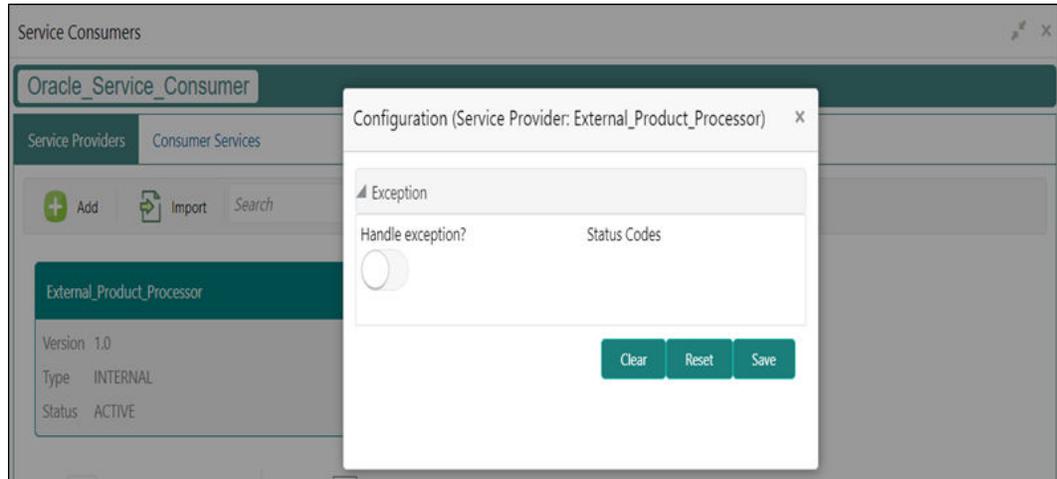
#### **Configuration**

End-user can configure the properties for failing the routing hub requests.

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

The **Configuration** screen displays.

**Figure 4-9 Configuration**



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

**Table 4-4 Configuration Service Provider - Field Description**

Field	Description
<b>Handle exception</b>	<p>This property is used to fail the routing hub request for failed provider requests.</p> <p><b>Note:</b> Default value is false.</p>
<b>Status Codes</b>	<p>This property is used to fail routing hub request for specific status codes of failed provider requests. If not specified, then routing hub request will fail for all 4xx and 5xx status codes of failed provider requests.</p>

### Request Audit

18. 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.

19. 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**

Actions	Name	Description	Service Name	Host	Port	Queue
:	FCUBS_Default	Default Implementation		xxxx	0	

### Add Implementation

The user can create the implementation manually.

2. Click **Add**.

The **Add Implementation** screen displays.

Figure 5-2 Add Implementation

The screenshot shows the 'Add Implementation' dialog box overlaid on the 'Service Consumers' application. The dialog contains the following fields and controls:

- Name \***: A text input field.
- Description \***: A text input field.
- Type \***: A dropdown menu currently set to 'DEFAULT'.
- Default**: A radio button, currently unselected.
- Eureka Instance**: A radio button, currently selected.
- Scheme \***: A dropdown menu currently set to 'Select'.
- Service Name \***: A text input field.
- Headers**: A section with a collapsed arrow.
- Save**: A button at the bottom right.

- 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
<b>Name</b>	Specify the name of the implementation.   <b>Note:</b> <ul style="list-style-type: none"> <li>Enter 0 to maximum of 255 characters.</li> <li>No numeric value at beginning and no space allowed.</li> </ul>
<b>Description</b>	Specify the description of the implementation.   <b>Note:</b> <ul style="list-style-type: none"> <li>Enter 0 to 1000 characters.</li> <li>No space allowed at beginning or ending of the characters.</li> </ul>

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

Field	Description
<b>Type</b>	Select the type of implementation from drop-down list The available options are: <ul style="list-style-type: none"> <li>• <b>DEFAULT</b></li> <li>• <b>QUEUE</b></li> </ul> <b>DEFAULT</b> type is for REST and SOAP API calls.
<b>Default</b>	Toggle the button if user wants to default. Each type can have one default implementation.
<b>Eureka Instance</b>	<b>Eureka Instance</b> is available only for internal providers and default type. By default, <b>Eureka Instance</b> will be toggled ON for internal providers and OFF for external providers.
<b>Scheme</b>	Select the scheme from drop-down list The available options are: <ul style="list-style-type: none"> <li>• <b>http</b></li> <li>• <b>https</b></li> </ul> <b>Scheme</b> option is available only for default type.
<b>Service Name</b>	If <b>Eureka Instance</b> is toggled ON and type is default, then only service name is required.
<b>Host</b>	Specify the host. <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• Host cannot be blank.</li> <li>• Enter 0 to 255 characters.</li> <li>• Space is not allowed.</li> </ul> </div> <p>If <b>Eureka Instance</b> is toggled OFF and type is default, then only host and port is required.</p>
<b>Port</b>	Specify the port number. <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• Enter 0 to 6 characters.</li> <li>• Enter only numeric value.</li> </ul> </div> <p>If <b>Eureka Instance</b> is toggled OFF and type is default, then only host and port is required.</p>
<b>Authentication</b>	The below fields appear only if <b>Eureka Instance</b> is toggled OFF and Implementation <b>Type</b> is selected as <b>Default</b> .

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

Field	Description
<b>Type</b>	Select the type of authentication from drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>BASIC</b></li> <li>• <b>JWT_TOKEN</b></li> <li>• <b>OAUTH_TOKEN</b></li> <li>• <b>SSO</b></li> <li>• <b>OAUTH_TOKEN_OIC</b></li> </ul>
<b>Username</b>	Specify the name of the user.   <b>Note:</b> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• No numeric value at beginning and no space allowed.</li> </ul>
<b>Password</b>	Specify the password.
<b>Headers</b>	The below fields appear only if the Implementation <b>Type</b> is selected as <b>Default</b> .
<b>Add</b>	Click this button to add header.
<b>Actions</b>	Displays the action. The user can edit or delete the header.
<b>Name</b>	Displays the name of the header.
<b>Value</b>	Displays the value of the header.
<b>Service</b>	The below fields appear only if the Implementation <b>Type</b> is selected as <b>Default</b> .
<b>Type</b>	Select the type of service from drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>WSDL</b></li> <li>• <b>SWAGGER</b></li> <li>• <b>OTHERS</b></li> </ul>
<b>URL</b>	Specify the service URL of the file location.   <b>Note:</b> <p>This field appears only if the <b>Type</b> is selected as <b>WSDL</b> and <b>SWAGGER</b>.</p>
<b>Name</b>	Specify the name of the operation.   <b>Note:</b> <p>This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>

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

Field	Description
<b>Http Method</b>	<p>Select the HTTP method. The available options are:</p> <ul style="list-style-type: none"> <li>• <b>GET</b></li> <li>• <b>POST</b></li> <li>• <b>PUT</b></li> <li>• <b>PATCH</b></li> <li>• <b>DELETE</b></li> </ul> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>
<b>Endpoint</b>	<p>Specify the endpoint URL for the operation.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>
<b>Gateway Prefix</b>	<p>Gateway Prefix is context path of below formatted URL. http://host:port/gateway-prefix/endpoint</p>
<b>Import</b>	<p>Click <b>Import</b> to extract the service information from URL and displays it in the Service list.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>WSDL</b> and <b>SWAGGER</b>.</p>
<b>Service Headers</b>	<p>Specify the Endpoint specific headers. Value can either be hardcoded or can be Velocity mapping.</p> <p> <b>Note:</b> This field appears only if the <b>Type</b> is selected as <b>OTHERS</b>.</p>

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

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

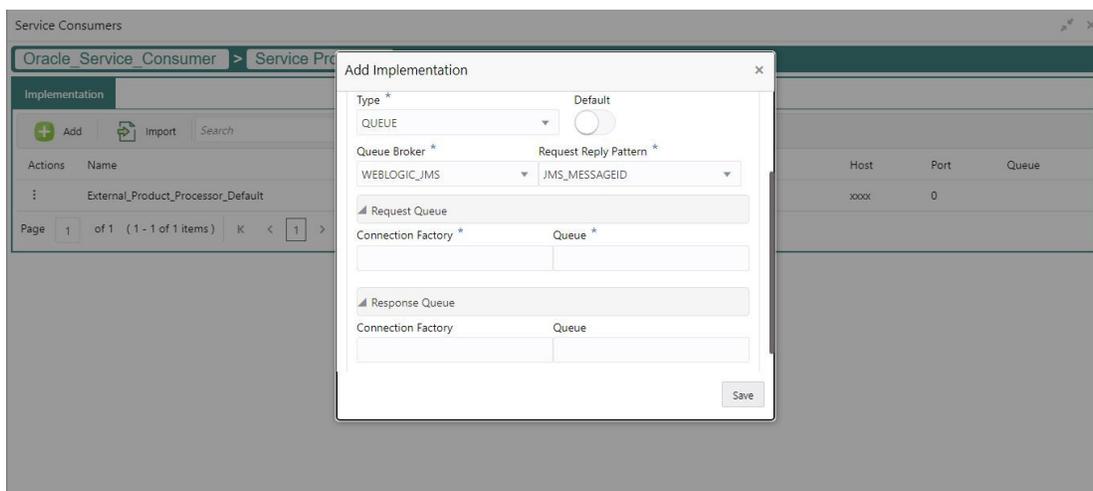
OTHERS option is selected for adding REST API details manually when provider does not have swagger file.

**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
<b>Type</b>	Select the type of implementation from drop-down list The available options are: <ul style="list-style-type: none"> <li>• <b>DEFAULT</b></li> <li>• <b>QUEUE</b></li> </ul> <b>DEFAULT</b> type is for REST and SOAP API calls.
<b>Default</b>	Select the toggle if the user wants to default.

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

Field	Description
<b>Queue Broker</b>	Select the queue broker from drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>WEBLOGIC_JMS</b></li> </ul>
<b>Request Reply Platform</b>	Select the queue broker from drop-down list. The available options are: <ul style="list-style-type: none"> <li>• <b>JMS_MESSAGEID</b></li> <li>• <b>JMS_CORRELATIONID</b></li> </ul> <b>JMS_MESSAGEID</b> is default request-reply pattern.
<b>Connection Factory</b>	Specify the connection factory. Connection Factory is JNDI based connection factory name which is used to create connection for JMS client.
<b>Queue</b>	Specify the queue. Queue Name is JNDI based destination name.
<b>Connection Factory</b>	Specify the connection factory. Response Connection Factory is needed when destination is going to respond back after processing the request.
<b>Queue</b>	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
<b>Name</b>	Specify the name for the header.
<b>Value</b>	Specify the value for the header.

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

The **Confirmation** screen displays.

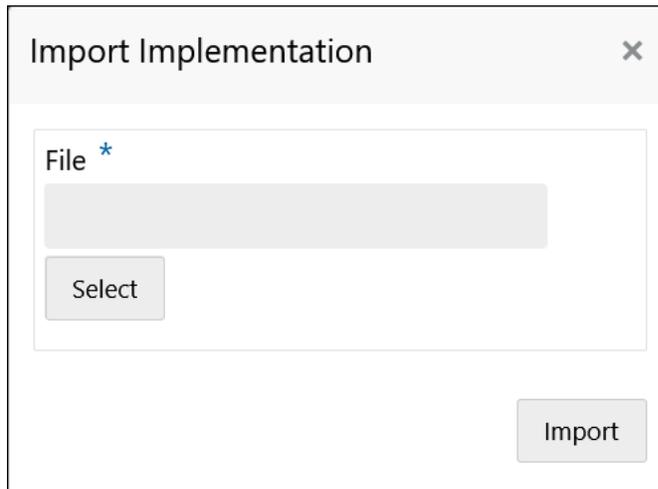
**Import Implementation**

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
<b>File</b>	<p>Click <b>Select</b> to select the file.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> <b>Note:</b></p> <p>Allows only to select one file and accepts JSON and ZIP file.</p> </div>

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

The screenshot shows a dialog box titled "View Implementation". It features a title bar with a pencil icon and a close button. The main content area is divided into several sections, each with a label and an input field:

- Name:** A single-line text input field.
- Description:** A multi-line text input field.
- Type:** A single-line text input field.
- Scheme:** A single-line text input field.
- Port:** A single-line text input field.
- Default:** A single-line text input field.
- Host:** A single-line text input field.

**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.

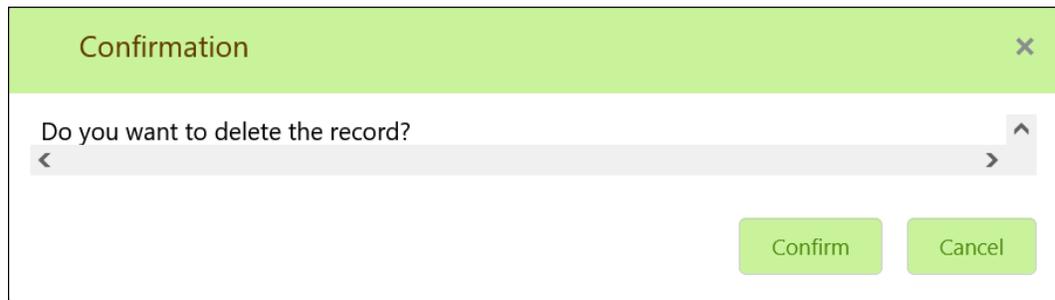
**Figure 5-7 Edit Implementation**

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.

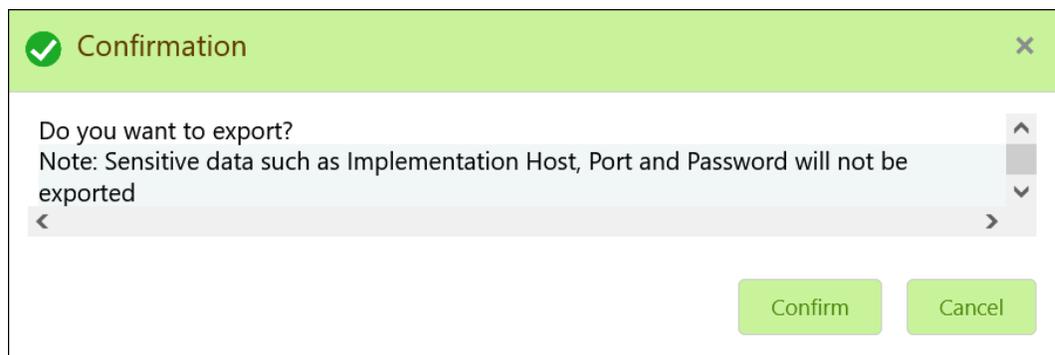
11. On **Implementation** screen, click **Operation menu** (3 dots button) and click **Delete**.  
The **Confirmation** screen displays.

**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

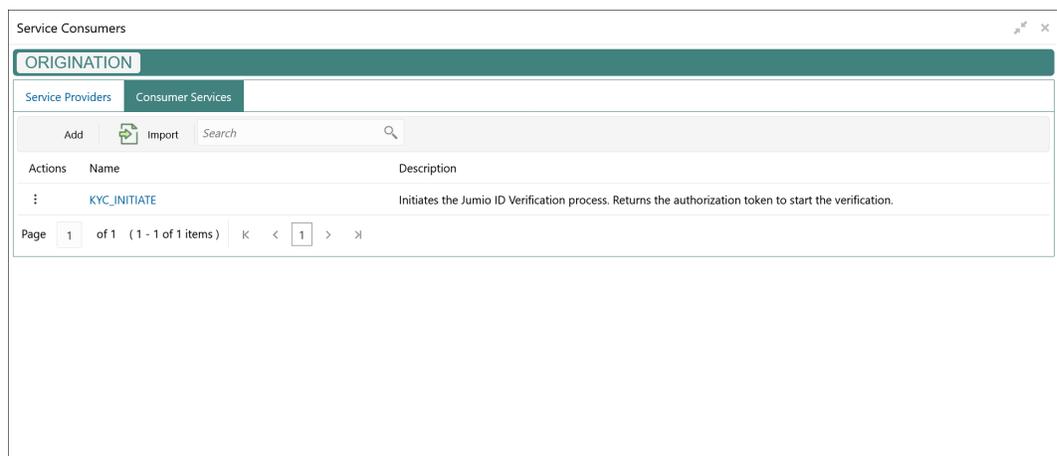
This topic describes the systematic instructions to configure the consumer services.

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

The screenshot shows a web form titled "Add Service". At the top right is a close button (X). The form contains the following elements:

- ID \***: A text input field with an asterisk indicating it is mandatory.
- Status**: A blue toggle switch that is currently turned on.
- Description \***: A large text area with an asterisk indicating it is mandatory.
- + Add**: A button with a green plus icon and the text "Add".
- Table**: A table with three columns: "Actions", "Attribute Name", and "json path". The table content is "No data to display."
- Save**: A button at the bottom right of the form.

- 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
<b>ID</b>	Specify the ID of the consumer service.   <b>Note:</b> <ul style="list-style-type: none"> <li>Enter 0 to maximum of 255 characters.</li> <li>No numeric value at beginning and no space allowed.</li> </ul>

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

Field	Description
<b>Status</b>	Active / Inactive If consumer service is marked as inactive, then all related routes will be stopped.
<b>Description</b>	Specify the description of the consumer service.  <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>• Enter 0 to 1000 characters.</li> <li>• No space allowed at beginning or ending of the characters.</li> </ul> </div>
<b>Add</b>	To add, refer to the below step.
<b>Actions</b>	Displays the action. The user can edit or delete the header.
<b>Attribute Name</b>	Displays the name of the attribute.
<b>json path</b>	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**

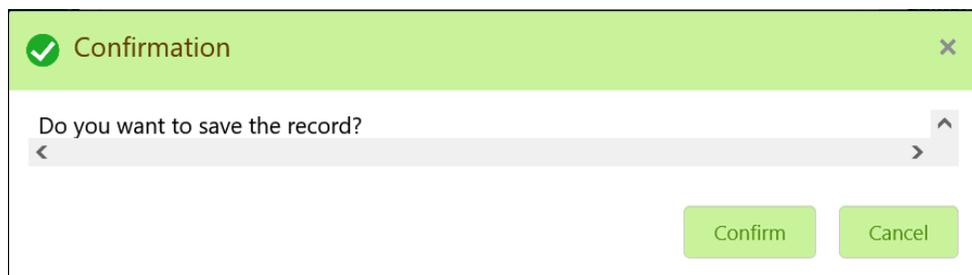
Field	Description
<b>Name</b>	Specify the name of the attribute.
<b>json path</b>	Specify the json path.

**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**

The screenshot shows a web interface for importing a service. At the top, there is a header bar with the text 'Import Service' and a close button (X). Below this is a form area. The first part of the form is labeled 'File \*' and contains a text input field that is currently empty, followed by a 'Select' button. Below the input field, there is a section for 'Overwrite extended templates' with two radio buttons: 'Yes' (which is unselected) and 'No' (which is selected). At the bottom right of the form area, there is an 'Import' button.

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

**Table 6-3 Import Service - Field Description**

Field	Description
<b>File</b>	Select the file using <b>Select</b> button.  <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #0070c0;"> <p> <b>Note:</b> Allows only to select one file and accepts only JSON file.</p> </div>
<b>Overwrite extended templates</b>	Select the respective radio button to overwrite the extended templates. The available options are: <ul style="list-style-type: none"> <li>• Yes - This option overwrites the extended templates.</li> <li>• No - This option retains the existing extended templates.</li> </ul>

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

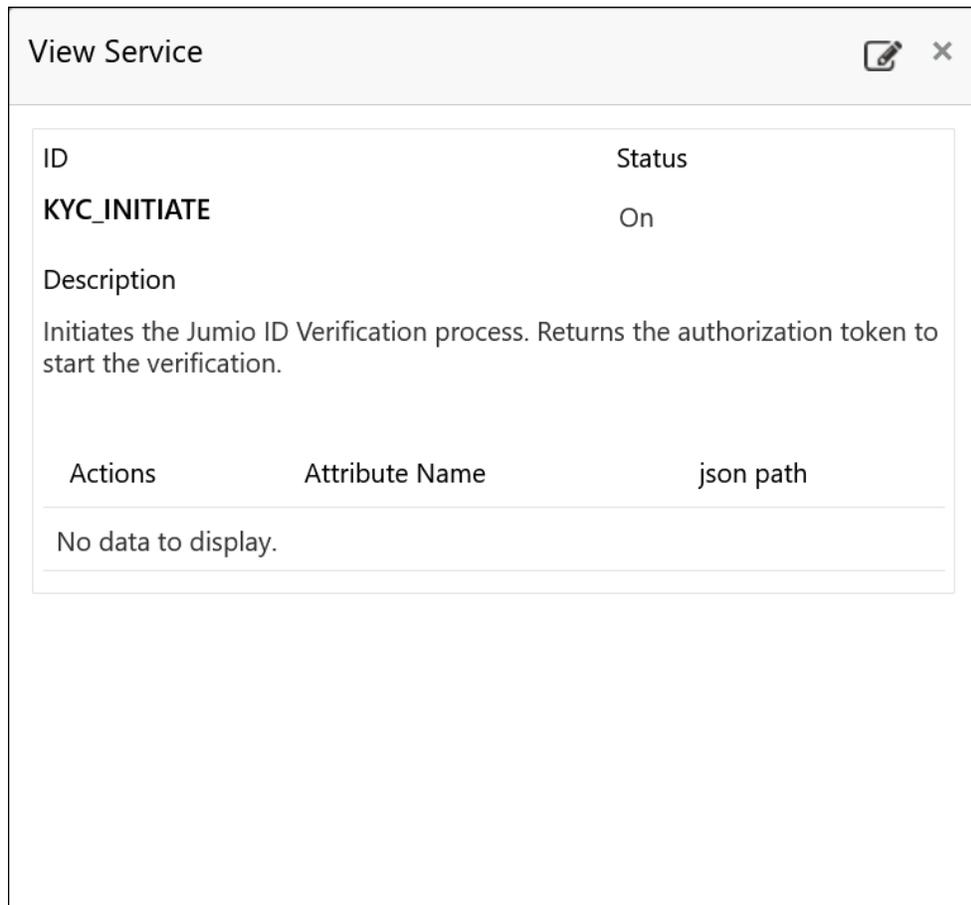
The **Confirmation** screen displays.

#### **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**.

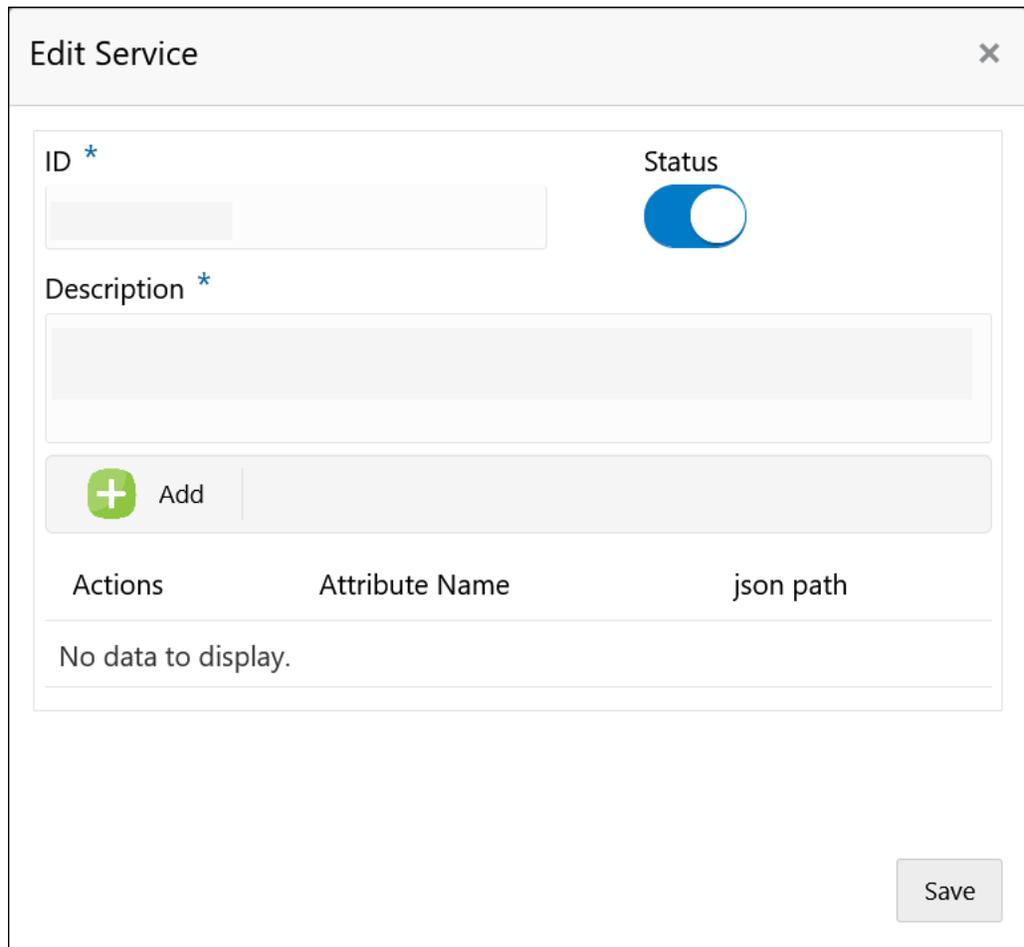
The **View Service** screen displays.

**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



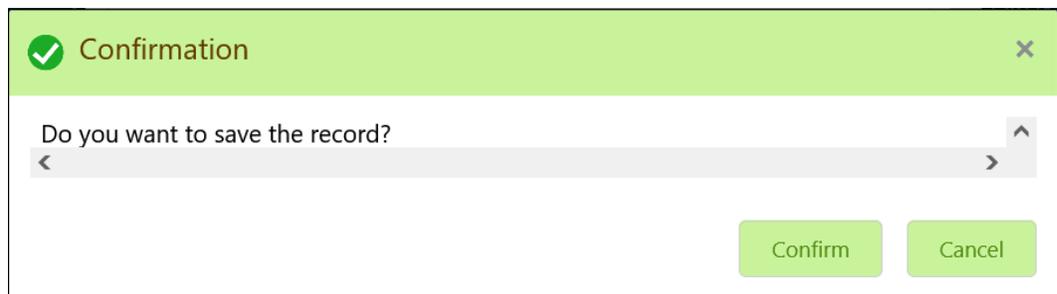
ID \* Status  
 [Text Input]   
 Description \*  
 [Text Area]  
 + Add  

Actions	Attribute Name	json path
No data to display.		

Save

- Click **Save** once the edit is done.  
The **Confirmation** screen displays.

Figure 6-8 Confirmation - Edit

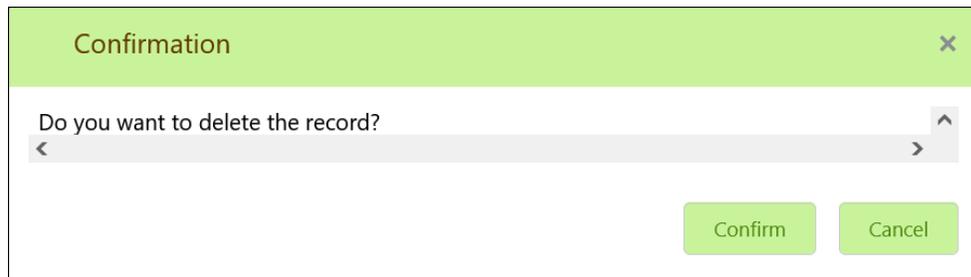


✓ Confirmation ×  
 Do you want to save the record?  
 [Text Area]  
Confirm Cancel

### Delete Consumer Service

The user can delete the consumer service.

- 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

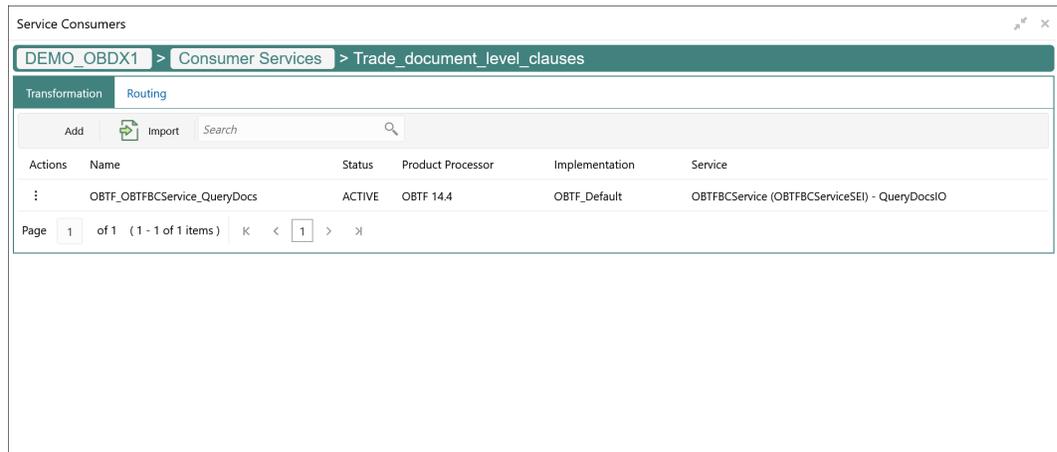
This topic describes the systematic instructions to configure the transformation.

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**



Actions	Name	Status	Product Processor	Implementation	Service
:	OBTF_OBTFBCService_QueryDocs	ACTIVE	OBTF 14.4	OBTF_Default	OBTFBCService (OBTFBCServiceSEI) - QueryDocsiO

### Add Transformation

The user can create transformation manually.

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

The **Add Transformation** screen displays.

Figure 7-2 Add Transformation

Add Transformation
✕

Name \*

Active

Product Processor

Product Processor \*

Implementation

Service \*

Service

Service

Operation

Headers

Actions	Name	Value
No data to display.		

Request Transformation

Template Type \*

Template

Extended Template

Response Headers

+ Add

Actions	Name	Value
No data to display.		

Response Transformation

Template Type \*

Template

Mocking required?

Mock Template

Extended Template

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
<b>Name</b>	Specify the name for the transformation.   <b>Note:</b> <ul style="list-style-type: none"> <li>• Enter 0 to maximum of 255 characters.</li> <li>• No numeric value at beginning and no space allowed.</li> </ul>
<b>Active</b>	Active / Inactive If transformation is marked as inactive, then the user will not be able to select transformation in routing.
<b>Product Processor</b>	Displays the Product Processor details.
<b>Product Processor</b>	Select the product processor from the drop-down list.
<b>Implementation</b>	Select the implementation from the drop-down list.
<b>Service</b>	Select the service from the drop-down list.
<b>Service</b>	Displays the service details of the selected service.
<b>Headers</b>	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.
<b>Path Params</b>	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.
<b>Query Params</b>	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.
<b>Request Transformation</b>	Displays the Request Transformation details.

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

Field	Description
<b>Body Type</b>	<p>Select the body type for the Request Transformation from the drop-down list.</p> <p>The available options are:</p> <ul style="list-style-type: none"> <li>• <b>RAW</b></li> <li>• <b>FORM DATA</b></li> </ul> <div style="border-left: 2px solid #0070C0; padding-left: 10px; margin-top: 10px;"> <p> <b>Note:</b></p> <p>This field appears only if the selected service is REST service.</p> </div>
<b>Template Type</b>	<p>Select the template type for the Request Transformation from the drop-down list.</p> <p>The available options are:</p> <ul style="list-style-type: none"> <li>• <b>VELOCITY</b></li> <li>• <b>JSLT</b></li> <li>• <b>XSLT</b></li> </ul>
<b>Template</b>	<p>Specify the template for the Request Transformation in which provider accepts.</p> <p>Refer to <a href="#">Transformation Type</a> for syntax.</p>
<b>Extended Template</b>	<p>Specify the custom template in order to extend the kernel template.</p> <p>Refer to Extensibility and Transformation Type for syntax.</p> <div style="border-left: 2px solid #0070C0; padding-left: 10px; margin-top: 10px;"> <p> <b>Note:</b></p> <p>This field appears only if the <b>Body Type</b> is selected as <b>FORM DATA</b>.</p> </div>
<b>Response Header</b>	<p>Specify the additional headers required to be part of Routing Hub response headers.</p> <p>Value can either be hardcoded or can be Velocity mapping.</p>
<b>Response Transformation</b>	Displays the response transformation details.
<b>Template Type</b>	<p>Select the template type for the Response Transformation from drop-down list.</p> <p>The available options are:</p> <ul style="list-style-type: none"> <li>• <b>VELOCITY</b></li> <li>• <b>JSLT</b></li> <li>• <b>XSLT</b></li> </ul>
<b>Template</b>	<p>Specify the kernel template in which consumer accepts.</p> <p>Refer to <a href="#">Transformation Type</a> for syntax.</p>
<b>Mocking required?</b>	<p>Select the toggle if the mocking is required for the Response Transformation or not.</p> <p>If the toggle is <b>ON</b>, the Routing Hub will return the mocked template output (with extended template output if mentioned) to consumer without invoking provider API.</p>

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

Field	Description
<b>Mock Template</b>	Specify the kernel template for the Response Transformation in which the consumer accepts. Refer <a href="#">Transformation Type</a> for syntax.

- Click **Save** to save the details.  
The **Confirmation** screen displays.
- 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).

- 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
<b>File</b>	Select the file using <b>Select</b> button.  <div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p> <b>Note:</b> Allows only to select one file and accepts JSON and ZIP file.</p> </div>

**Table 7-2 (Cont.) Import Transformation - Field Description**

Field	Description
<b>Overwrite extended templates</b>	Select the respective radio button to overwrite the extended templates. The available options are: <ul style="list-style-type: none"> <li>• Yes - This option overwrites the extended templates.</li> <li>• No - This option retains the existing extended templates.</li> </ul>

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

View Transformation
✎ ✕

Name	Active	
PostTransformation	On	
<span style="font-size: 0.8em;">▲ Product Processor</span>		
Product Processor	Implementation	Service
<span style="font-size: 0.8em;">▲ Service</span>		
Service		
Operation		
<span style="font-size: 0.8em;">▲ Headers</span>		
Actions	Name	Value
No data to display.		
<span style="font-size: 0.8em;">▲ Request Transformation</span>		
Template Type		
VELOCITY		
Template		
Extended Template		
<span style="font-size: 0.8em;">▲ Response Headers</span>		
Actions	Name	Value
No data to display.		
<span style="font-size: 0.8em;">▲ Response Transformation</span>		
Template Type	Template	
VELOCITY		
Mocking required?	Mock Template	
Off		
	Extended Template	

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

Name \*
Active

Product Processor \*
Implementation
Service \*

Service

Operation

Headers

Actions	Name	Value
No data to display.		

Request Transformation

Template Type \*

Template

Extended Template

Response Headers

+
Add

Actions	Name	Value
No data to display.		

Response Transformation

Template Type \*

Template

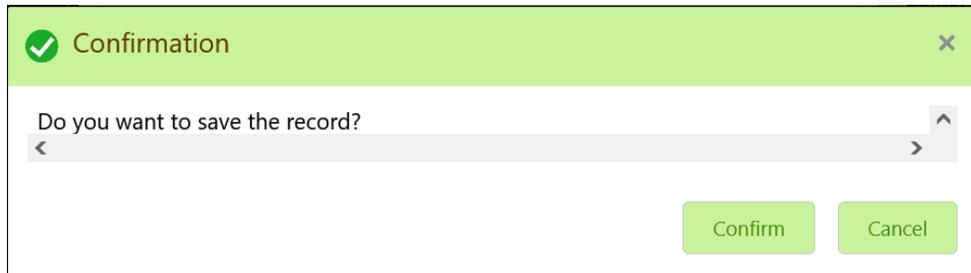
Mocking required?

Mock Template

Extended Template

10. Click **Save** once the edit is done.  
The **Confirmation** screen displays.

**Figure 7-6 Confirmation**

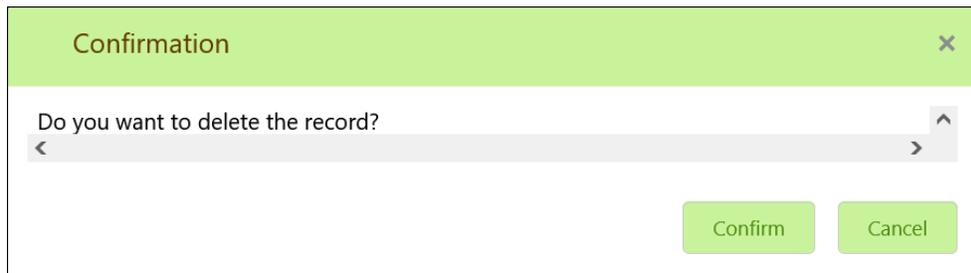


#### Delete Transformation

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**

Actions	Name	Start/Stop	Rule	Product Processor	Implementation	Service
:	OBTF_OBTFBCService_QueryDocs_R1	On	OBTF 14.4	OBTF_Default	OBTFBCService (OBTFBCServiceSEI) - QueryDocsIO	

### Add Route

The user can create routing manually.

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

The **Add Route** screen displays.

**Figure 8-2 Add Route**

The screenshot shows the 'Add Route' dialog box with the following elements:

- Name \***: A text input field.
- Start/Stop**: Two buttons, 'START' (blue) and 'STOP' (grey).
- Auto Shutdown**: A toggle switch currently turned on.
- Default Rule**: An unselected radio button.
- Custom Rule**: A selected radio button.
- Rule**: A section header with a dropdown arrow.
- Expression Editor \***: A large text area for entering the rule expression.
- Transformations**: A section header with a dropdown arrow.
- Add**: A button with a green plus icon to add transformations.
- Table**: A table with columns: 'Actions', 'Product Processor', 'Implementation', and 'Transformation'. The table is currently empty, displaying 'No data to display.'
- Save**: A button at the bottom right of the dialog.

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
<b>Name</b>	Specify the name for the route.  <div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>Enter 0 to maximum of 255 characters.</li> <li>No numeric value at beginning and no space allowed.</li> </ul> </div>
<b>START / STOP</b>	<b>START / STOP</b> If routing is marked as STOP, then consumer request fails at routing hub level only.
<b>Auto Shutdown</b>	<b>ON / OFF</b> 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.
<b>Rule Type</b>	Select the rule type. The available options are: <ul style="list-style-type: none"> <li><b>Default Rule</b></li> <li><b>Custom Rule</b></li> </ul>
<b>Expression Editor</b>	Displays the expression that is formed through expression editor.
<b>Add</b>	To add, refer to the below steps.
<b>Actions</b>	Displays the action. The user can edit or delete the header.
<b>Product Processor</b>	Displays the product processor.
<b>Implementation</b>	Displays the implementation.
<b>Transformation</b>	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
<b>Attribute</b>	Select attribute relevant to consumer service from drop-down list.
<b>Operator</b>	Select the logical operators to form an expression from drop-down list.
<b>Value</b>	Specify the value.   <b>Note:</b> Enter 0 to 255 characters.
<b>Condition Type</b>	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

Add Transformation
✕

Product Processor \*

Implementation \*

Transformation \*

▾ Headers

Actions	Name	Value
No data to display.		

▾ Product Processor

Service

▾ Request Transformation

Template Type	Template

▾ Response Transformation

Template Type	Template

Mocking required?	Mock Template
-------------------	---------------

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

 **Note:**

The fields, which are marked with an asterisk, are mandatory.

ORACLE

8-6

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

**Table 8-3 Add Transformation - Field Description**

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

**View Route**

Name:

Start/Stop:

Auto Shutdown:

Default Rule  Custom Rule

Transformations

Actions	Product Processor	Implementation	Transformation
⋮	<input type="text"/>	<input type="text"/>	<input type="text"/>

**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**

The screenshot shows the 'Request Audit' interface. At the top, there are search filters for Request Id, Consumer, Consumer Service, Provider, Provider Implementation, Provider Service, Transformation, Route, and User Id. A 'Search' button is located below the filters. Below the filters is a table with columns: Request Id, Consumer, Consumer Service, Provider, Provider Implementation, Provider Service, Transformation, Route, Status, and User Id. The table currently displays 'No data to display.' and a pagination bar showing 'Page 1 of 0 (1 - 0 of 0 items)'.

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
<b>Request ID</b>	Specify the request ID.
<b>Consumer</b>	Specify the consumer.
<b>Consumer Service</b>	Specify the consumer service.
<b>Provider</b>	Specify the provider.

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

Field	Description
<b>Provider Implementation</b>	Specify the provider implementation.
<b>Provider Service</b>	Specify the provider service.
<b>Transformation</b>	Specify the transformation name.
<b>Route</b>	Specify the route.
<b>User ID</b>	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**

The screenshot shows a window titled "Request Audit Details" with a close button (X) in the top right corner. The content includes:

- Request Id:** eqwiLi5IXb4N1dZTT8KKZimRnxj3Q16tby8u5n0Vkv62B0dtlsNFN1hZcJmWrgCuxhwCn2vIcBkbbqNHuiELNt38
- Progress Bar:** A horizontal line with four green checkmarks in circles above it, indicating successful completion of all steps. The steps are labeled: OBRH Request, Provider Request, Provider Response, and OBRH Response.
- Timestamp:** 2021-02-23T12:34:42.128+05:30
- Message:** [{"headers":{"branchCode":["004"],"Authorization":["Bearer eyJhbGciOiJIUzUxMiJ9.eyJ0aWQiOiIiLCJzdWiiOiJlPQkRYX0FOSUtFVCIslmF1ZCI6IldFQilslmIhdCI6MjYxNDQ2Mzc4NiwiZXhwIjoxNjE0MDY0Mzg2fQ.R-Jo42xYSH7cEb7k2BxUvWHN53W\_Nd9z3RJixrKMDSsbju9hWpG7RB8yY\_dnTtO8tO5sA8S6lIGBqa-j-P5Taw"],"Content-Type":["application/json"],"User-Agent":["Mozilla/5.0 (Windows NT 10.0; Win64; x64)"]}]

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

**Table 9-2 Request Audit Details - Field Description**

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

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

<b>Field</b>	<b>Description</b>
<b>OBRH Response</b>	Displays the status of Routing Hub response.
<b>Timestamp</b>	Displays the date and time.
<b>Message</b>	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

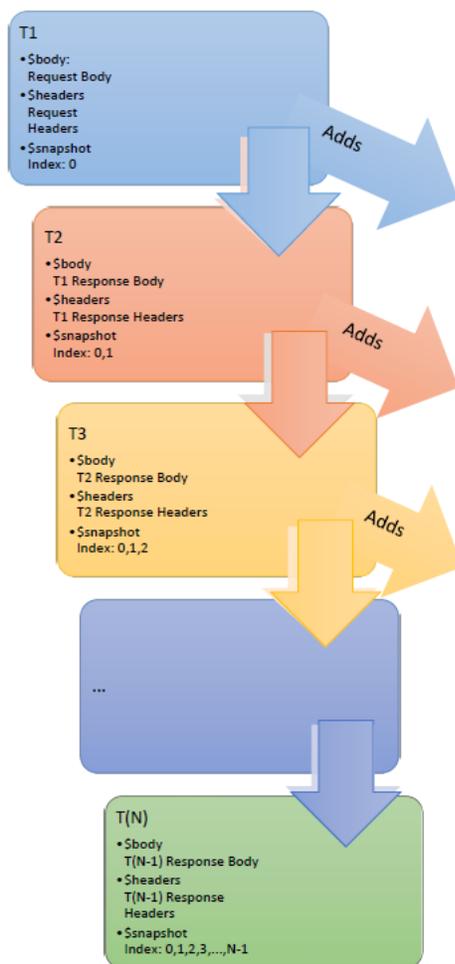


Table 10-1 Snapshot List

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

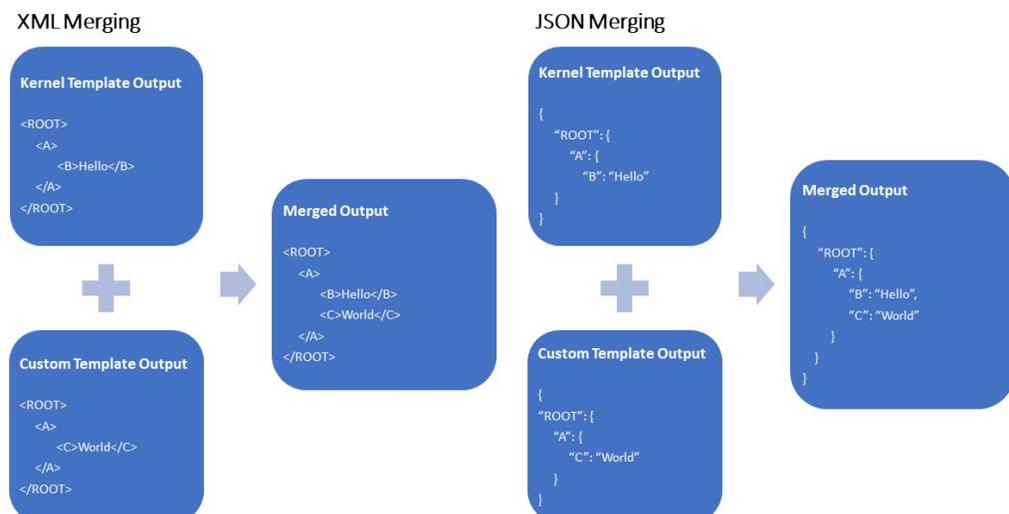
## Extensibility

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

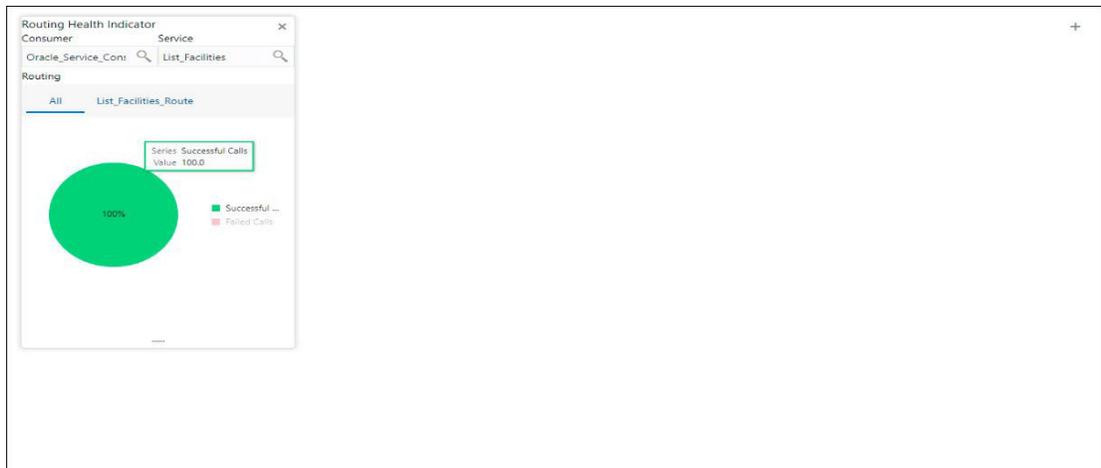
The user can view the metric information Successful calls vs Failed calls ratio of each routing.



#### 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:**
  - appld : SECSR001
  - 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": ""  
}
```



- 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.

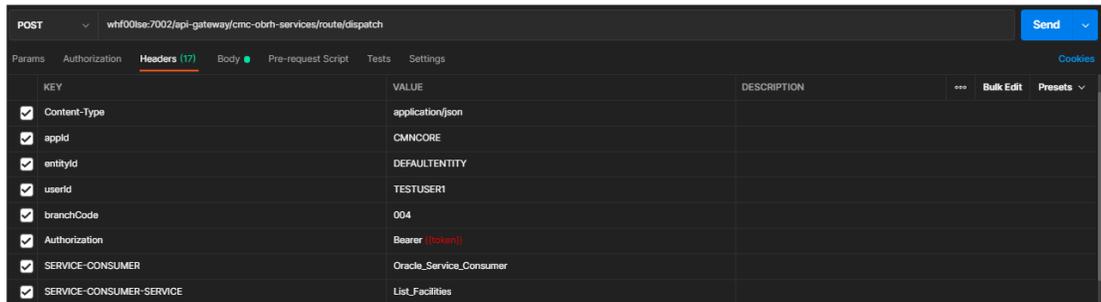
- **Response Body:**

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

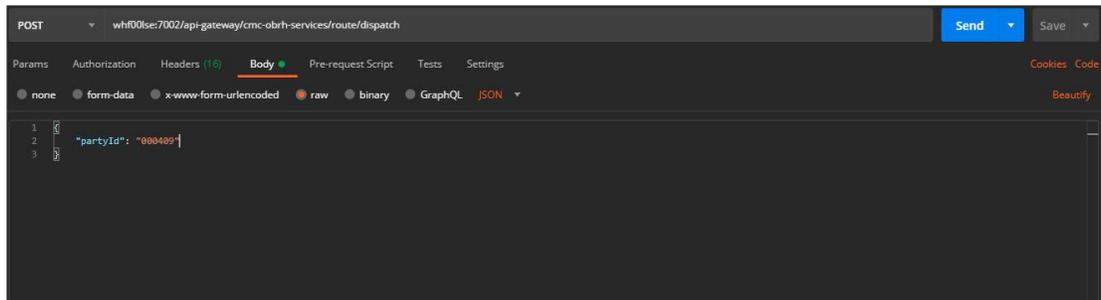
- If the route invocation succeeds, data JSON member would contain the transformed (optional) response of the provided service. If it is a provided web service and no response transformation template is supplied, XML Soap Body of response would be converted into JSON object and sent in data JSON member.
- 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**



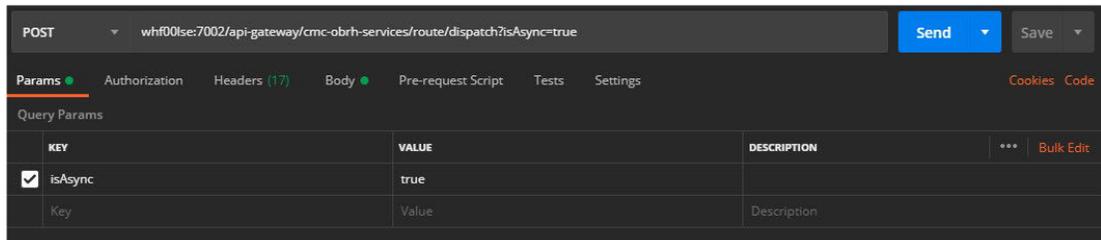


- **Request Body:**
  - Any valid JSON payload which shall act as input to the transformation template in request transformer.
- **Response Body:**

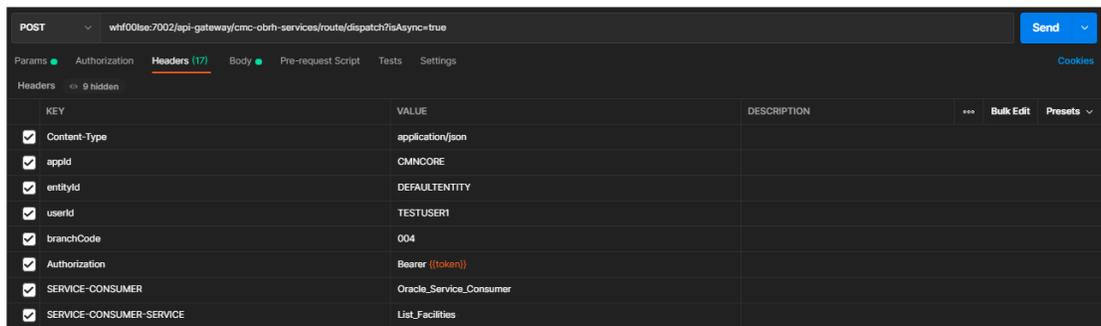
```
{
  "data": { "correlationId" : "" },
  "messages": {}
}
```

**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**

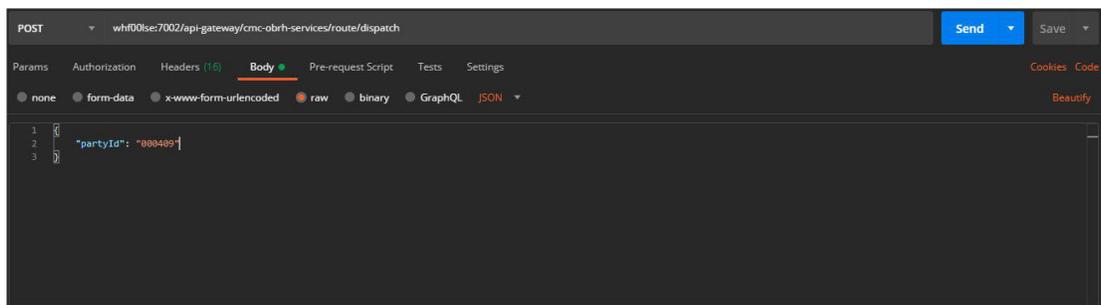


Figure 16-11 Response Payload

```

1  {
2    "data": {
3      "correlationId": "UZUwPK3LvkKCUaduX2x2u1IH9RCFNrDP17pyiu999A1qgHTJXKPAD44tV9Q9ktkv8rVpr7Um6J291Pr2gox2RBS"
4    },
5    "messages": {
6      "keyId": null,
7      "status": "SUCCESS",
8      "codes": [
9        {
10         "args": null,
11         "arg": null,
12         "information": true,
13         "override": false,
14         "error": false,
15         "overrideAuthLevelsReqd": null,
16         "desc": "Request is being processed",
17         "type": "I",
18         "code": "CHC-DBRH-023",
19         "language": "ENG"
20       }
21     ],
22     "requestId": null,
23     "httpStatusCode": "OK",
24     "overrideAuthLevelsReqd": null
25   }
26 }

```

## 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**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Content-Type	application/json	
<input checked="" type="checkbox"/> appId	CMNCORE	
<input checked="" type="checkbox"/> entityId	DEFAULTENTITY	
<input checked="" type="checkbox"/> userId	TESTUSER1	
<input checked="" type="checkbox"/> branchCode	004	
<input checked="" type="checkbox"/> Authorization	Bearer (token)	

**Figure 16-13 Response Payload when request is still processing**

```

1  {
2    "data": null,
3    "messages": {
4      "keyId": null,
5      "status": "SUCCESS",
6      "codes": [
7        {
8          "args": null,
9          "arg": null,
10         "information": true,
11         "override": false,
12         "error": false,
13         "overrideAuthLevelsReqd": null,
14         "desc": "Request is being processed",
15         "type": "I",
16         "code": "CMC-OBRR-023",
17         "language": "ENG"
18       }
19     ],
20     "requestId": null,
21     "httpStatusCode": "OK",
22     "overrideAuthLevelsReqd": null
23   }
24 }

```

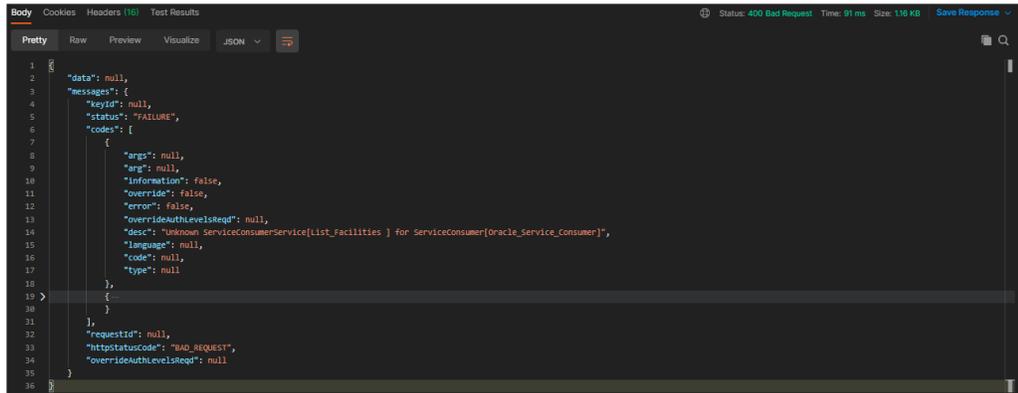
**Figure 16-14 Response Payload when request is processed (on Successful Dispatch)**

```

1  {
2    "data": {
3      "facilitydtos": [...
81    ]
82  },
83  "messages": {
84    "keyId": null,
85    "status": "SUCCESS",
86    "codes": [],
87    "requestId": null,
88    "httpStatusCode": "OK",
89    "overrideAuthLevelsReqd": null
90  }
91 }

```

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



```
1 {
2   "data": null,
3   "messages": {
4     "keyid": null,
5     "status": "FAILURE",
6     "codes": [
7       {
8         "args": null,
9         "arg": null,
10        "information": false,
11        "override": false,
12        "error": false,
13        "overrideAuthLevelReqd": null,
14        "desc": "Unknown ServiceConsumerService[list_facilities ] for ServiceConsumer[Oracle_service_consumer]",
15        "language": null,
16        "code": null,
17        "type": null
18      }
19    ]
20  },
21  "requestId": null,
22  "httpStatusCode": "BAD_REQUEST",
23  "overrideAuthLevelReqd": null
24 }
```

# 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 overwrite the customization that is not part of configuration JSON,

```
-Dobrh.import.override=<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_POOLSIZ> (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

## A

---

Asynchronous Dispatch API Specification, [16-4](#)  
Asynchronous Dispatch Response API  
Specification, [16-6](#)  
Audit Purging / Archiving, [12-1](#)

## C

---

Chaining, [10-1](#)  
Configuration, [2-1](#)  
Consumer Services, [6-1](#)

## D

---

Dashboard, [14-1](#)

## E

---

Extensibility, [11-1](#)

## I

---

Implementation, [5-1](#)

## J

---

JSLT, [15-3](#)

## M

---

Multipart Request, [13-1](#)

## O

---

Oracle Banking Routing Hub Integration  
Specification, [16-1](#)  
Oracle Banking Routing Hub VM Arguments,  
[17-1](#)

## R

---

Request Audit, [9-1](#)  
Routing, [8-1](#)  
Routing Health Indicator Widget, [14-1](#)

## S

---

Service Consumers, [3-1](#)  
Service Providers, [4-1](#)  
Synchronous Dispatch API Specification, [16-2](#)

## T

---

Token Generation, [16-1](#)  
Transformation, [7-1](#)  
Transformation Type, [15-1](#)

## V

---

Velocity, [15-1](#)

## X

---

XSLT, [15-3](#)