Oracle® Banking Origination
Troubleshooting Guide

Release 14.8.0.0.0

ORACLE®

Oracle Banking Origination Troubleshooting Guide, Release 14.8.0.0.0

G28922-02

Contents

Preface

1 Troubleshooting Technical Flows

2   Health Checks

3   Troubleshooting Application Workflows

4   Business Error Codes

Index

Preface

This topic contains the following sub-topics:

Purpose

This guide provides guidance to users for the issues within the application. It describes various methods to figure out the error and then troubleshoot it.

Before you begin

Kindly refers to [Getting Started user Guide](#) for common elements, including Symbols and Icons, Conventions Definitions, and so forth.

Pre-requisite

Specify the **User Name** and **Password**, and login to **Home screen**.

Audience

This guide is intended for the software developers and software testers.

Documentation Accessibility

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

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, refer to the following documents:

*   Getting Started User Guide

*   Oracle Banking Microservices Platform Foundation User Guide

*   Oracle Banking Common Core User Guide

*   Routing Hub Configuration User Guide

*   Oracle Banking Security Management System User Guide

*   Observability User Guide

*   Oracle Banking Origination Installation Guide

*   Oracle Banking Origination Integration Guides

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

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

Table 1   Acronyms and Abbreviations

Acronyms and Abbreviations	Description
API	Application Programming Interface
ELK	Elasticsearch, Logstash, and Kibana
HTTPS	Hypertext Transfer Protocol Secure
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
UI	User Interface
URL	Uniform Resource Locator

Symbols and Icons

The list of common icons are available on all screen as follows.

Table 2   Icons - Common

Icon	Function
	Calendar
	Alerts
	Refresh
	Unlock Option
	View Option

Table 2   (Cont.) Icons - Common

Icon	Function
	New
	Copy
	Delete
	Save
	Export
	Sorting

The list of icons available on the view screens are as follows.

Table 3   Icons - Widget

Icon	Function
	Open status
	Authorized status
	Unauthorized status
	Rejected status
	Closed status
	Modification Number

The list of symbols available on audit details screens as follows.

Table 4   Symbols - Audit Details

Symbols	Function
	A user
	Application Date and time

Table 4   (Cont.) Symbols - Audit Details

Symbols	Function
	Calendar Date and time
	Branch
	Remarks
	Authorized or Open status
	Unauthorized or Closed status
	Rejected status

Post requisite

After finishing all the requirements, please log out from the Home screen.

1

Troubleshooting Technical Flows

This topic describes about various programming issues, possible causes, and solutions to resolve the issues.

This topic contains the following subtopics:

- [Where is the Problem](#)
  This topic describes about troubleshooting the problem in the distributed system.

- [Preliminary Checks from UI](#)
  This topic provides systematic instructions to launch the application and check for the basic errors.

- [Preliminary Checks from Service Log Files](#)
  This topic describes about preliminary checks from service log files.

- [Troubleshoot Errors Using Zipkin Traces](#)
  This topic describes the systematic instructions to troubleshoot the errors using the Zipkin Traces.

- [Troubleshooting Logs using ELK Stack](#)
  This topic describes about Troubleshooting Logs using ELK Stack to access Kibana, search logs in Kibana, and export logs.

- [Troubleshooting Kafka](#)
  This topic provides systematic instruction to resolve the issues while running Kafka.

- [Troubleshooting Environmental Issues](#)
  This topic describes about the troubleshooting environmental issues while deploying services, logging in, or launching a screen.

1.1 Where is the Problem

This topic describes about troubleshooting the problem in the distributed system.

Troubleshooting the problem in the distributed system can be challenging if not understood fully. Each product has UI application components and service side application components. Each side requires different troubleshooting techniques and various logs that can be used to confirm the problem.

Figure 1-1   Distributed Flow across Micro Services

Distributed flow across micro services
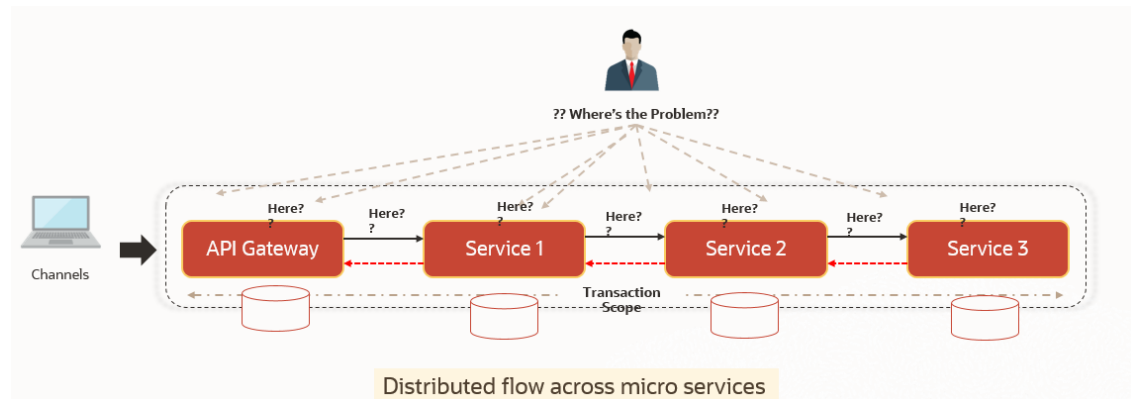
The above image shows that it is important to establish the area of the problem. This can be achieved by complete understanding of UI, service side flows along with the data architecture of application.

1.2 Preliminary Checks from UI

This topic provides systematic instructions to launch the application and check for the basic errors.

1. Log in to the application homepage.

    For information on how to log in, refer to the *Getting Started User Guide*.

2. Press **F12** key and select **Inspect**.

    The **Network** screen displays.

Figure 1-2   Call Responses
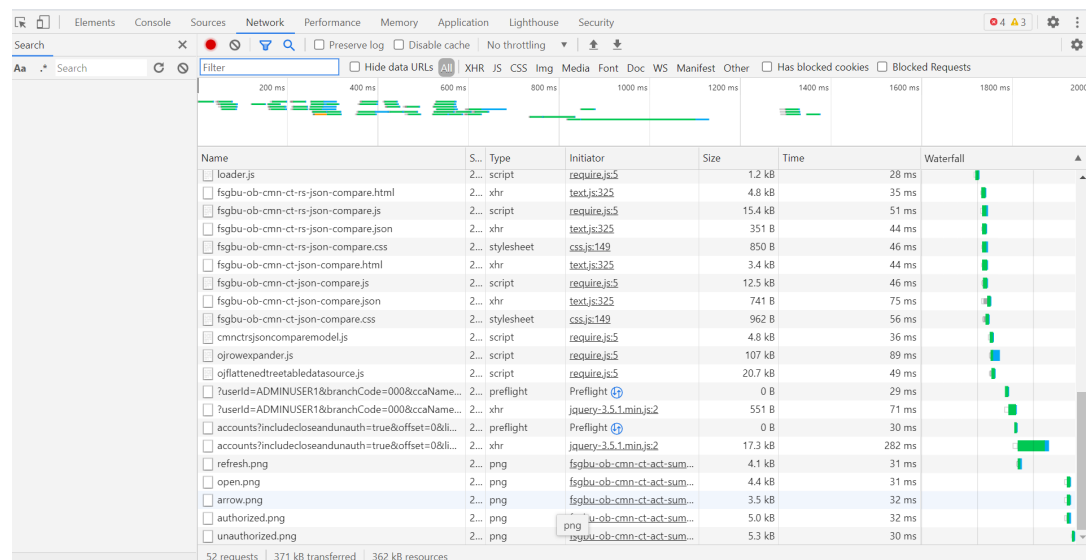
3. Verify that all the call responses are successful.

> (i) **Note**
>
> Usually red color indicates a non-2xx HTTP response.

Figure 1-3   Non-2xx Response
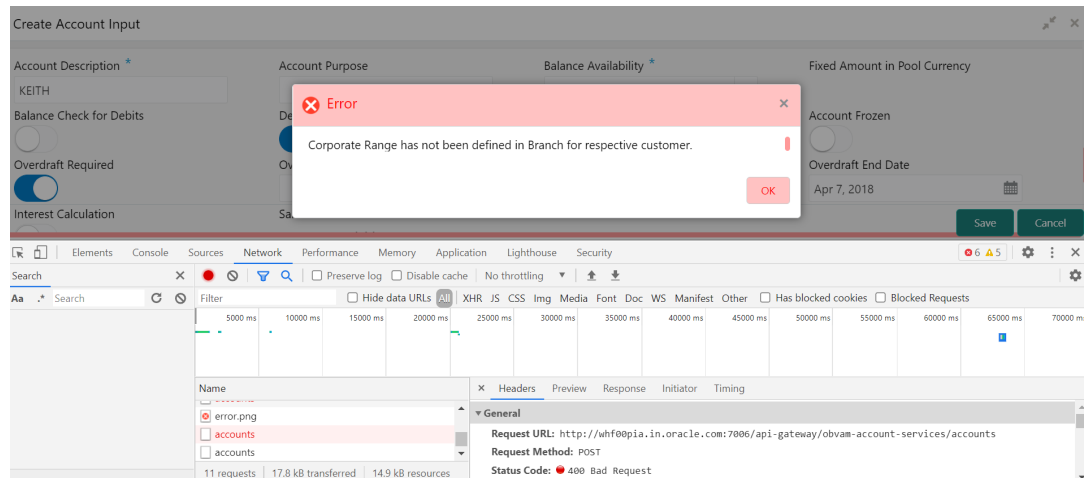
4. Click **Export** to export the traces from browsers.

   The export option is shown below in Chrome.

Figure 1-4   Export Option
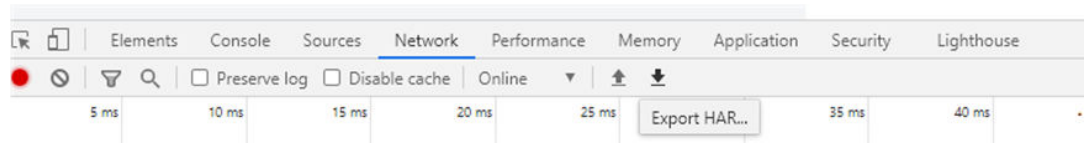
> (i) **Note**
>
> The tools such as **Fiddler** and **Wireshark** can be used to get the browser to API gateway web traffic. This helps to investigate the exact request and response payloads exchanged between UI and API Gateway.

1.3 Preliminary Checks from Service Log Files

This topic describes about preliminary checks from service log files.

The war deployments for each microservice sub-domain can generate the log files in the WebLogic server.

The configuration of this log can be found at `logback.xml`:

```
<root level="INFO">
            <appender-ref ref="FILE" />
</root>
```

In production scenarios, make sure that the root level is configured as **ERROR** so that log files do not get overwhelmed. Refer to **Oracle WebLogic Server Documentation Library** to know the path where these files are generated. In on-premises cases, the log files can be zipped and sent for remote troubleshooting purposes.

1.4 Troubleshoot Errors Using Zipkin Traces

This topic describes the systematic instructions to troubleshoot the errors using the Zipkin Traces.

Set up the Zipkin server. For information on how to set it up, refer to the **Observability User Guide**.
To perform troubleshooting using Zipkin Traces:

1. Launch the Zipkin URL.

   The basic layout of Zipkin displays.

   Figure 1-5   Layout of Zipkin

2. Use **Search** to find the traces of required API calls and services.

   > ⓘ **Note**
   >
   > The search options given in the user interface are self-explanatory, and there is another UI option (**Try Lens UI**). It is given a different user interface with the same functionality.

   Some error API calls are made to showcase how to track errors. The blue listing shows the successful API hits, and the red listing indicates the errors. Each block indicates a single trace in the listing.

   The list of traces are displayed.

Figure 1-6   List of Traces
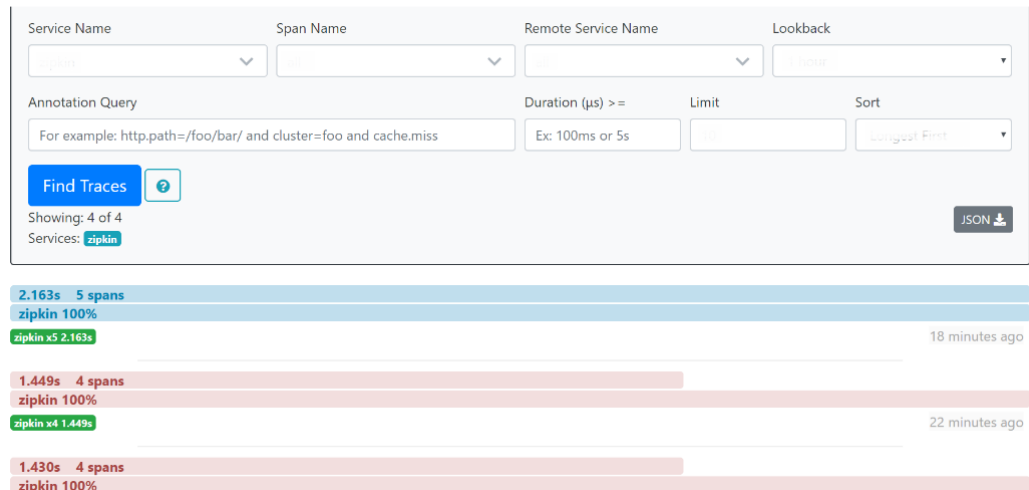
3. Open the individual trace.

   It describes the time taken for each block. As the two custom spans are created inside two service calls, user can find a total of four blocks. The time taken for an individual block is shown below.

   The details of an individual trace displays.

Figure 1-7   Individual Trace
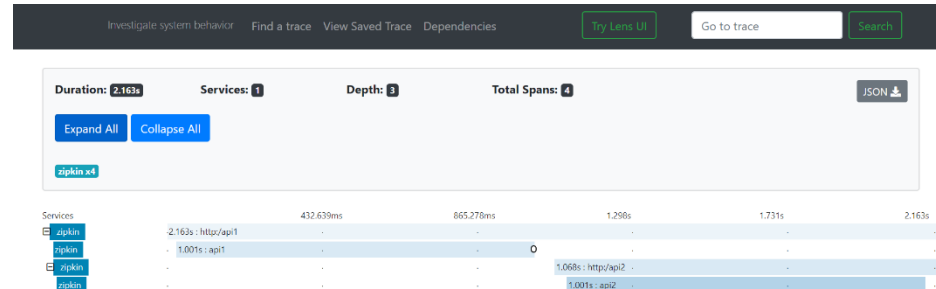
4. Click on the individual block.

   The details of an individual block are displayed.

Figure 1-8   Details of Individual Block
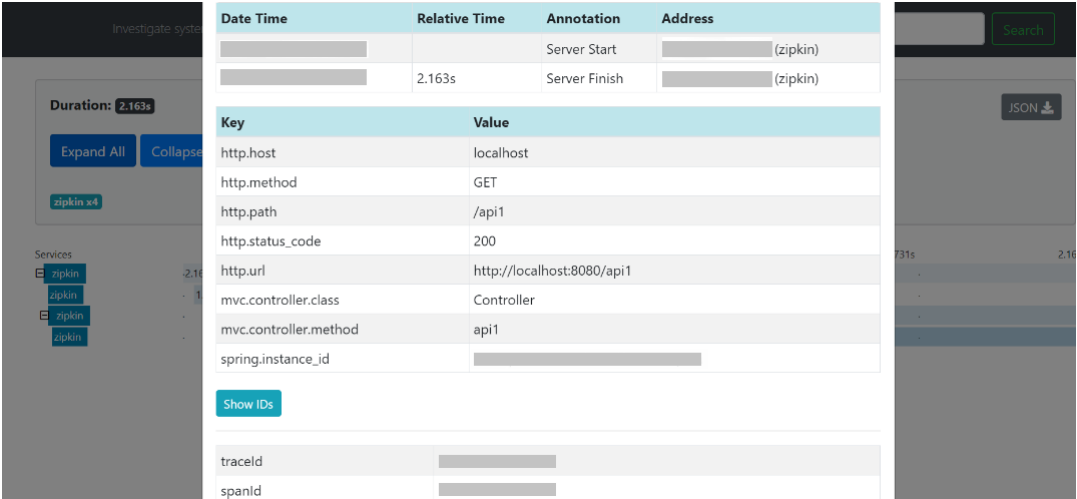
The user can also view the logging events in the Zipkin UI as small circular blocks. An example of an error log is shown below.

Figure 1-9   Sample Error Log
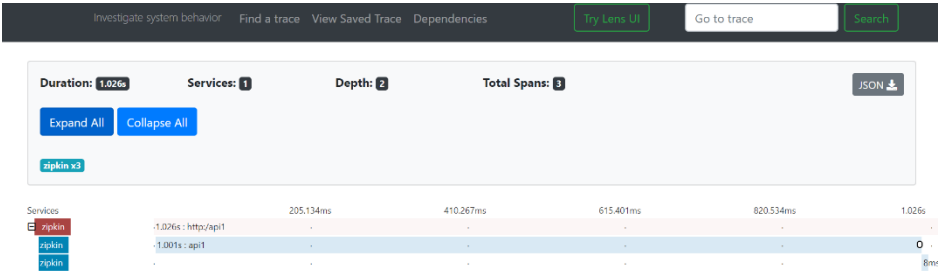
5.   Click the error portion.

The details about the error and where the error has arisen are displayed.

Figure 1-10   Details of Error
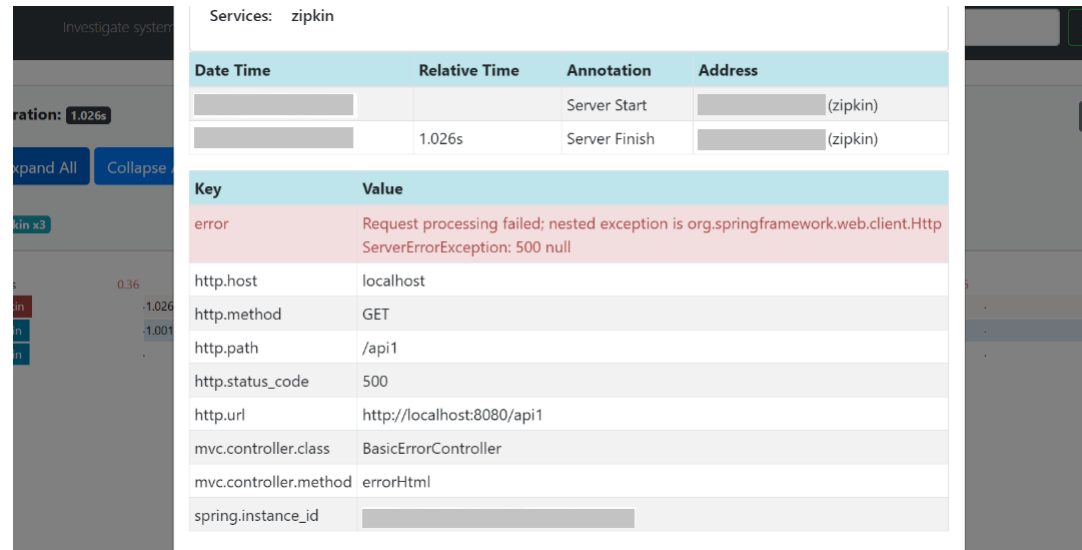
> **ⓘ Note**
>
> If the **Lens UI** is used in Zipkin, the above figures are not applicable but are relatable to the **Lens UI** as well. Traces of the application can be found using **TraceId**. The **TraceId** can be found in the debug logs of the deployment when `spring-cloud-sleuth` is included in the dependencies (included in `spring-cloud-starter-zipkin` dependency).

6. Click the **Dependencies** tab.

The dependency graph information between micro-services is displayed.

Figure 1-11   Sample Dependency Graph
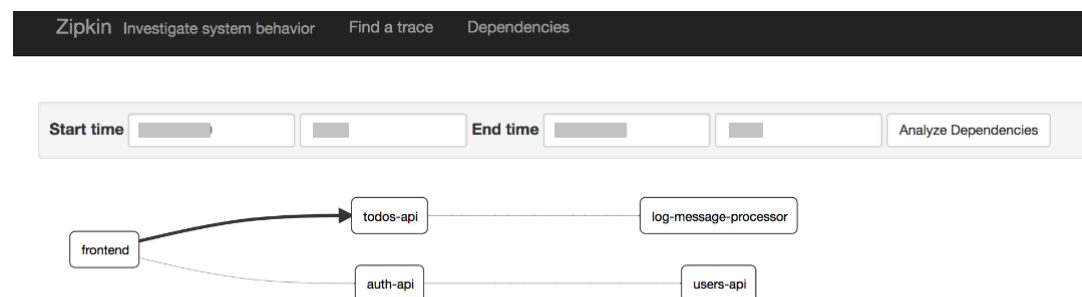
This topic contains the following sub-topic:

- [Known Issues for Zipkin](#)
  This topic describes the known issues encountered when using Zipkin and how to work around them.

1.4.1 Known Issues for Zipkin

This topic describes the known issues encountered when using Zipkin and how to work around them.

The below known issues addressed when using Zipkin:

- **Application Service is not Registered**
  Perform the following steps to find the cause of this error:

  1. Check the applications, which are sending the trace report to the Zipkin server from **Service Name** drop-down list.

  Figure 1-12   Find Traces

  2. If the required application is not listed in Zipkins, check the `application.yml` file for Zipkin base URL configuration.

  Figure 1-13   `application.yml` File

  > (i) **Note**
  >
  > The shipped `application.yml` should have the Zipkin entry. Every service should have `spring-cloud-sleuth-zipkin` dependency added in the build gradle file for the service to generate and send *trace Id* and *span Id*.

  3. The necessary values are as follows:

     - Compile group: `org.springframework.cloud`

     - name: `spring-cloud-sleuth-zipkin`

– version: `2.1.2.RELEASE`

Figure 1-14   Origination Common Services
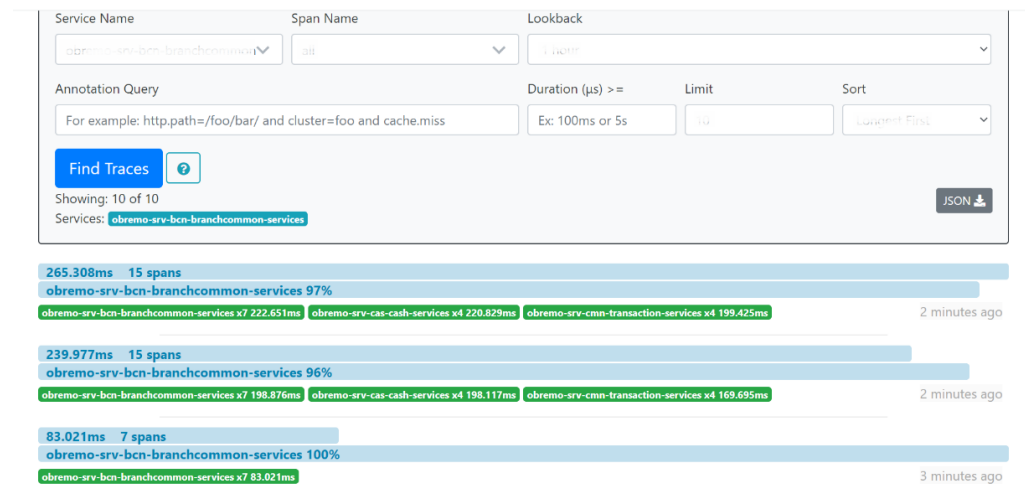
Figure 1-15   Origination Common Services Trace
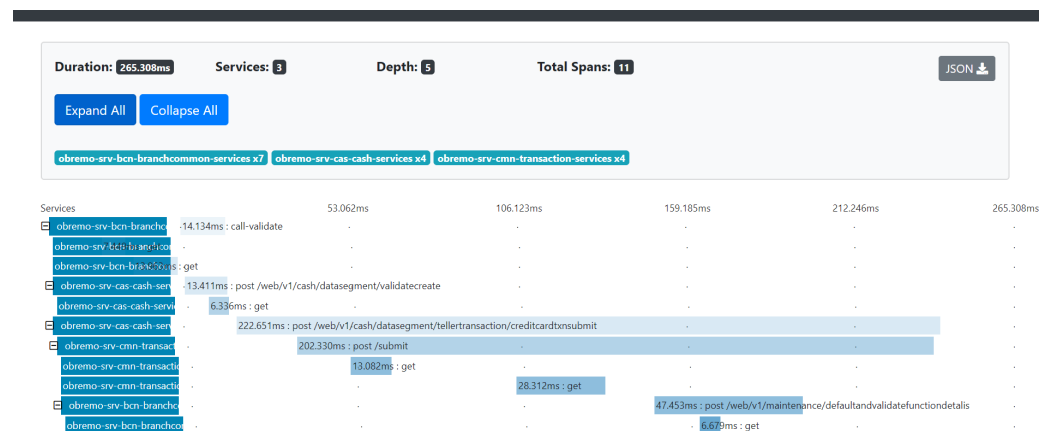
- **Page Not Found**

  If there is a 404 error, check if the `zipkin-server.jar` is running in the system where the application is deployed. To check this, execute the following command:

  ```
  netstat -ltnup | grep ':9411'
  ```

  A sample output is shown below:

  ```
  tcp6    0    0    :::9411    :::*    LISTEN    10892/java
  ```

  > ⓘ **Note**
  >
  > In the sample output, `10892` is the PID.

- **Unable to Change Zipkin Default Port Number**

The default port number of the Zipkin is not editable. Hence, make sure that port *9411* is available to start the `Zipkin-server.jar` file.

1.5 Troubleshooting Logs using ELK Stack

This topic describes about Troubleshooting Logs using ELK Stack to access Kibana, search logs in Kibana, and export logs.

This topic contains the following subtopics:

- Set Up ELK
  This topic provides the links to setup ELK for troubleshooting the logs using ELK stack.

- Access Kibana
  This topic provides systematic instructions to access Kibana.

1.5.1 Set Up ELK

This topic provides the links to setup ELK for troubleshooting the logs using ELK stack.

The prerequisites are as follows:

- Download the Elastic search from https://www.elastic.co/downloads/elasticsearch.

- Download the Kibana from https://www.elastic.co/downloads/kibana.

- Download the Logstash from https://www.elastic.co/downloads/logstash.

Figure 1-16   ELK Setup

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "whf00peb"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://localhost:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
```

> **ⓘ Note**
>
> The default ports are as follows:
>
> - Elastic search - 9200
> - Kibana - 5601

To run the ELK:

1. Run the `elasticsearch.sh` file present in the folder path `/scratch/software/ELK/elasticsearch-6.5.1/bin`.

   - Edit **network.host** to **localhost** and port if necessary. This should be enough for it to run.
   - Start: **nohup bin/elasticsearch &**

2. Configure the Kibana to point the running instance of elastic search in the `kibana.yml` file.

3. Configure the Logstash.

   For more information on configurations, refer to the table below.

Table 1-1   Configurations for Logstash

Configuration	Description
Input	This configuration is required to provide the log file location for the Logstash to read from.
Filter	Filters in Logstash is basically used to control or format the read operation (Line by line or Bulk read).
Output	This provides the running elastic search instance to send the data for persisting.

Figure 1-17   Logstash Configuration

```
logstash.conf

#Point to the application logs
input {
  beats {
    port => 5044
  }
}
#Provide the parsing logic to transform logs into JSON
filter {
 # Adding @metadata needed for index sharding to Filebeat logs
  mutate {
    copy => {
      "[fields][app_name]" => "[@metadata][app_name]"
      "[fields][env]" => "[@metadata][envt]"
    }
  }

  #If log line contains tab character followed by 'at' then we will tag that entry as stacktrace
  if [message] =~ "\tat" {
    grok {
      match => ["message", "^(\tat)"]
      add_tag => ["stacktrace"]
    }
  }
```

Figure 1-18   Elasticsearch

```
#Grokking Spring Boot's default log format
grok {
  match => [ "message", "%{TIMESTAMP_ISO8601:timestamp}\s+%{LOGLEVEL:severity}\s+\[%{DATA:service},%{DATA:trace},%{DATA:span}
}


#Parsing out timestamps which are in timestamp field thanks to previous grok section
date {
  match => [ "timestamp" , "yyyy-MM-dd HH:mm:ss.SSS" ]
}
fingerprint {
  source => "message"
  target => "[@metadata][fingerprint]"
  method => "MD5"
  key => "test"
}

ruby {
  code => "event.set('[@metadata][prefix]', event.get('@timestamp').to_i.to_s(16))"
}
}

#Ingest logs to Elasticsearch
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "%{[@metadata][app_name]}-%{[@metadata][envt]}-%{+YYYY.MM.dd}"
    document_id => "%{[@metadata][prefix]}%{[@metadata][fingerprint]}"
  }
  stdout { codec => rubydebug }
}
```

1.5.2 Access Kibana

This topic provides systematic instructions to access Kibana.

1. Go to path `/kibana-7.8.1-linux-x86_64/config/kibana.yml`.

2. Edit server.host: "0.0.0.0" for access outside host and server.port: <any port, defaults to 5601>.

3. Validate elasticsearch properties.

   It defaults to localhost:9200

4. Go to **http://host:port**

   The Kibana console UI is displayed.
   Kibana requires elasticsearch to be UP as it creates indexes and fetch logs.

5. Start the nohup bin or Kibana.

   The Kibana screen is displayed.

Figure 1-19   Kibana
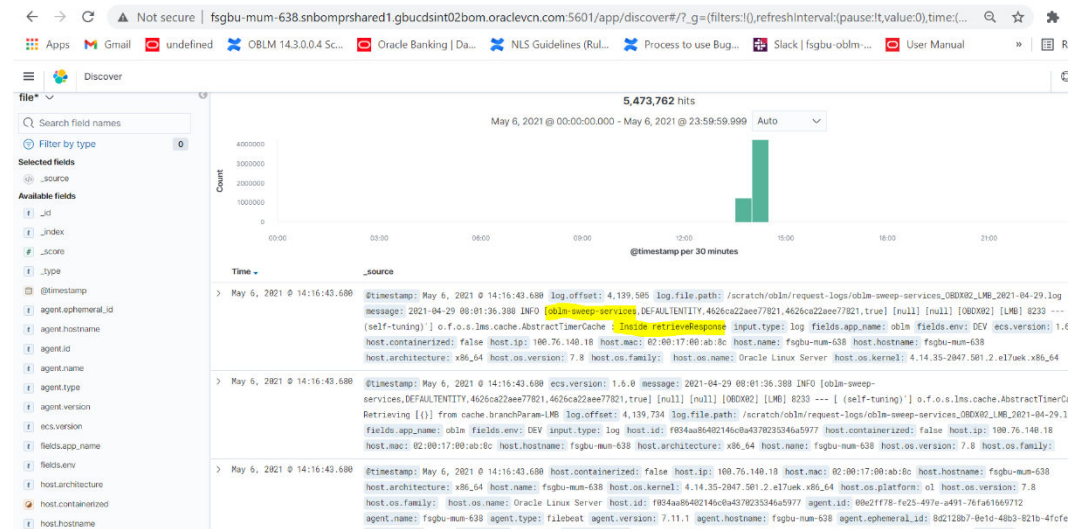
1.6 Troubleshooting Kafka

This topic provides systematic instruction to resolve the issues while running Kafka.

The possible issues that may occur while running Kafka.

1. Execute the command $ `netstat -tlnp | grep :9092`.

> ⓘ **Note**
>
> 9092 is the default port of kafka.

Kafka may not be starting because Zookeeper has not started yet.

2. Execute the command $ `netstat -tlnp | grep :2181`.

> ⓘ **Note**
>
> 2181 is default port of zookeeper.

If any service is not running on this port, it means Zookeeper is down.

3. Verify that if any permission issue is there for kafka log folder.

   • To create console producer and consumer for troubleshooting, refer to http://cloudurable.com/blog/kafka-tutorial-kafka-from-command-line/index.html.

   • To more about Kafka, refer to https://docs.cloudera.com/documentation/kafka/latest/topics/kafka_faq.html

1.7 Troubleshooting Environmental Issues

This topic describes about the troubleshooting environmental issues while deploying services, logging in, or launching a screen.

This topic contains the following subtopics:

- **Possible Issues While Deploying Services**
  This topic describes the possible issues that may occur in the environment while deploying services.

- **Possible Issues in Login and Screen Launch**
  This topic describes the possible issues that may occur while logging in to the application and launching the screens.

1.7.1 Possible Issues While Deploying Services

This topic describes the possible issues that may occur in the environment while deploying services.

- If the service deployment is failing due to flyway, verify that the object or record is already present and make changes in the flyway scripts accordingly.

- The other possible issue while deploying services could be multiple versions of dependency jars present in the war file. For example, `weblogic.application.naming.EnvironmentException`: duplicate persistence units with the name `PLATO` in scope `cmc-customer-services-(Version).war`.

1.7.2 Possible Issues in Login and Screen Launch

This topic describes the possible issues that may occur while logging in to the application and launching the screens.

This topic contains the following sub-topics:

- **Login Page is not Launching**
  This topic provides the systematic instructions to resolve the issue if login page is not launching.

- **Unable to login after launching the application**
  This topic describes the issues to login after launching the application.

- **Unable to login after restarting the services**
  This topic describes the issues to login after the services is restarted.

- **Menus are not displayed after logging in**
  This topic describes the issues when menus are not displayed after logging in.

- **Screens are not launching after logging in**
  This topic describes the issues when screens are not launched after logging in.

1.7.2.1 Login Page is not Launching

This topic provides the systematic instructions to resolve the issue if login page is not launching.
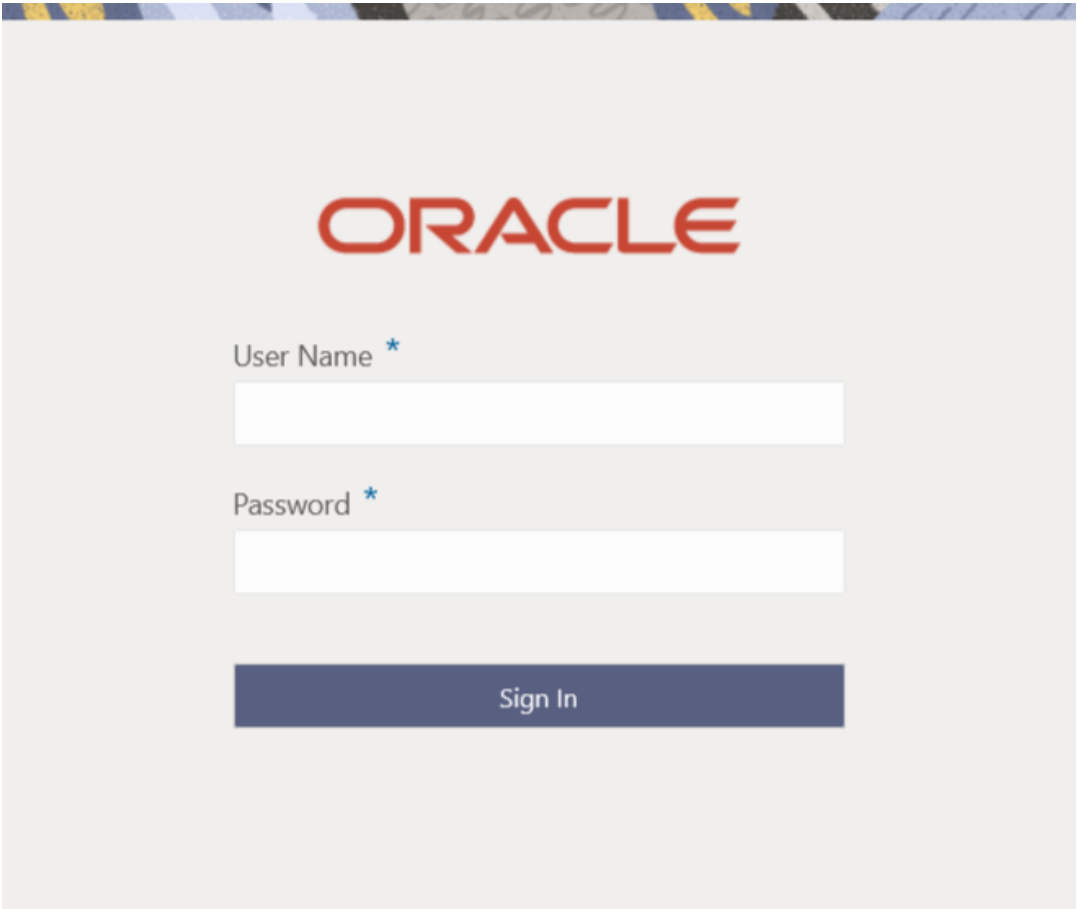
Perform the following checks if the login page is not launching.

1. Check whether the app-shell war file is deployed.

2. Make sure that the war file is up and running in the deployed managed server and try to login again.

3. Check whether the user has logged in with the appshell URL according to the war file deployed.

4. Check whether the required component-server wars like cmc-component-server, obo-component-server etc are also deployed along with the app-shell.

   Example: `http://<localhost>:<Port>/app-shell/index.jsp` will load the login page of the application.
   In the above URL, the name `app-shell` is dynamic which depends on the name of war file deployed.

Figure 1-20   Login Page

1.7.2.2 Unable to login after launching the application

This topic describes the issues to login after launching the application.

Perform the following check if user is not able to login after the application is launched.

- Make sure that the `plato-api-gateway service`, `plato-ui-config service`, `sms-core-service`, and common core services are up and running.

Figure 1-21   Services

PLATO-API-GATEWAY	n/a (1)	(1)	UP (1) - fsgbu-phx-54.snphxprshared1.gbucdsint02phx.oraclevcn.com:plato-api-gateway:5012
PLATO-DISCOVERY-SERVICE	n/a (1)	(1)	UP (1) - fsgbu-phx-54.snphxprshared1.gbucdsint02phx.oraclevcn.com:plato-discovery-service:5012
PLATO-UI-CONFIG-SERVICES	n/a (1)	(1)	UP (1) - fsgbu-phx-54.snphxprshared1.gbucdsint02phx.oraclevcn.com:plato-ui-config-services:5012
SMS-CORE-SERVICES	n/a (1)	(1)	UP (1) - fsgbu-phx-54.snphxprshared1.gbucdsint02phx.oraclevcn.com:sms-core-services:5012

1.7.2.3 Unable to login after restarting the services

This topic describes the issues to login after the services is restarted.

Perform the following check if user is not able to login after restarting the services.

- Make sure that the LDAP server is up and running, and check if the entered credentials are correct.

1.7.2.4 Menus are not displayed after logging in

This topic describes the issues when menus are not displayed after logging in.

If the menus are not displayed after logging in, map the functional activity codes in the table `SMS_TM_ROLE_ACTIVITY`.

Once it is mapped, check if the corresponding role is assigned to your user ID.

1.7.2.5 Screens are not launching after logging in

This topic describes the issues when screens are not launched after logging in.

If the screens are not launching after logging in, make sure that the respective services are up and running.

> ⓘ **Note**
>
> Verify the VPN connection while trying to troubleshoot the issues related to page launching, etc.

2

Health Checks

This topic provides information about health checks.

Until the heath check APIs are implemented, the health need to be monitored using WebLogic JVM managed server status and Eureka instance.

Figure 2-1   Health Checks

Application	AMIs	Availability Zones	Status
CMC-ACCOUNT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-account-services:7005
CMC-ADVICE-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-advice-services:7005
CMC-BASE-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-base-services:7005
CMC-BRANCH-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-branch-services:7005
CMC-BUSINESSOVERRIDES-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-businessoverrides-services:7005
CMC-CHECKLIST-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-checklist-services:7005
CMC-COMMENTS-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-comments-services:7005
CMC-CURRENCY-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-currency-services:7005
CMC-CUSTOMER-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-customer-services:7005
CMC-DATASEGMENT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-datasegment-services:7005
CMC-DOCUMENT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-document-services:7005
CMC-EXTERNAL-CHART-ACCOUNT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-external-chart-account-services:7005
CMC-OBCBS-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-obcbs-services:7005
CMC-OBRH-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-obrh-services:7005

This topic contains the following subtopics:

- Configure Data Sources in WebLogic
  This topic describes systematic instructions to configure the data sources in WebLogic.

2.1 Configure Data Sources in WebLogic

This topic describes systematic instructions to configure the data sources in WebLogic.

1. On the WebLogic console, in the **Domain Structure** panel, click **Data Sources**.

   The **Summary of JDBC Data Sources** screen is displayed.

Figure 2-2 Summary of JDBC Data Sources

2. On the **Summary of JDBC Data Sources** screen, click **New** and add the data source providing the required details.

   The **Create a New JDBC Data Source** screen is displayed.

Figure 2-3   Create a New JDBC Data Source

Figure 2-4   Create a New JDBC Data Source

Home >Summary of Servers >Summary of JDBC Data Sources >PLATO >**Summary of JDBC Data Sources**

Create a New JDBC Data Source

| Back | Next | Finish | Cancel |

Connection Properties

Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name: OBLMDB

What is the name or IP address of the database server?

Host Name: whf00bqa.in.oracle.com

What is the port on the database server used to connect to the database?

Port: 1521

What database account user name do you want to use to create database connections?

Database User Name: OBLM144DEVPLATO

What is the database account password to use to create database connections?

Password: ••••••••••••••

Confirm Password: ••••••••••••••

Additional Connection Properties:

oracle.jdbc.DRCPConnectionClass:
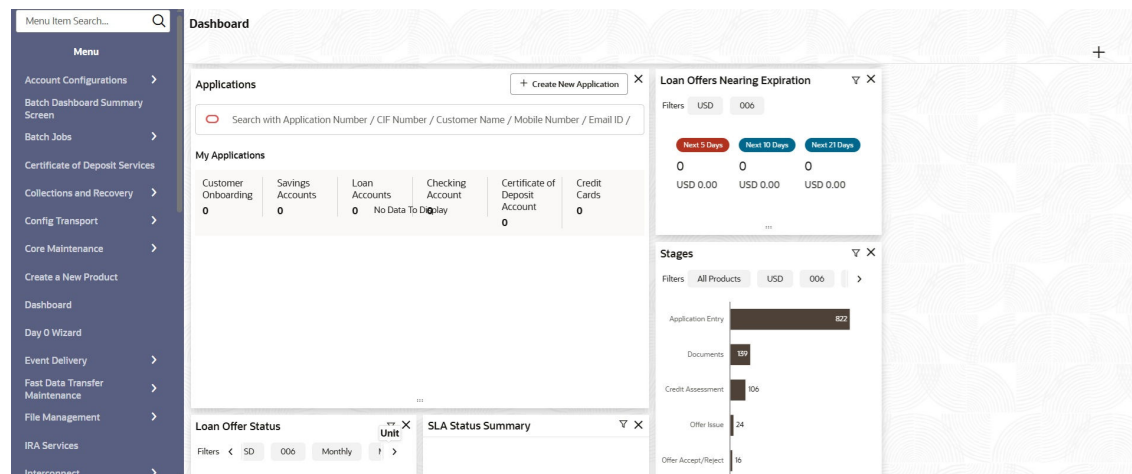
| Back | Next | Finish | Cancel |

3

Troubleshooting Application Workflows

This topic provides information about troubleshooting application workflows.

On successful login, the Oracle Banking Origination dashboard screen displays depending on the user privileges.

Figure 3-1   Oracle Banking Origination Dashboard

To troubleshooting User Role Issues:

Role Profile defines access rights to the functional activities that are common to a group of users. Linking a user to a Role Profile grants access to all the functional activities included in that profile.

> ⓘ **Note**
>
> Only authorized users can access the system with the help of a unique User Login ID and password.

1. From the **Home screen**, click **Security Management**. Under **Security Management**, click **Role**.

2. Under **Role**, click **Create Role**.

   The **Role Maintenance** screen displays.

Figure 3-2   Role Maintenance

The user profile of a user contains the details of the user in four sections - User details, Status, Other details and User Role branches.

3. From the **Home screen**, click **Security Management**. Under **Security Management**, click **User**.

4. Under **User**, click **Create Users**.

   The **Users Maintenance** screen displays.

Figure 3-3   Users Maintenance

> ⓘ **Note**
>
> Make sure that the required Role and User Applications are mapped to the user.

This topic contains the following sub-topics:

- First level issues
  This topic describes the possible issues that may occur during the basic investigation.

- Transaction data verification
  This topic provides information about the transaction data verification.

- Party Module Integration Troubleshooting
  This topic describes the possible issues that may occur in Party Module integration.

- FLEXCUBE Host Integration Troubleshooting
  This topic describes the possible issues that may occur in FLEXCUBE Universal Banking Solution integration.

3.1 First level issues

This topic describes the possible issues that may occur during the basic investigation.

This topic contains the following subtopics:

- Error Message not shown
  This topic describes about Error Message not shown.

- Set Log File Path
  This topic describes about Setting Log file path.

- Dynamic Log Generation Issues
  This topic describes about Dynamic log generation issues and how to work around them.

- Call is Failing in Gateway
  This topic describes about the Call is failing in Gateway.

- Code error in GCS side
  This topic describes about the error code in GCS Side.

- 404 Error
  This topic describes about the 404 error and how to work around them.

- 500 internal error
  This topic describes about the 500 Internal Error and how to work around them.

3.1.1 Error Message not shown

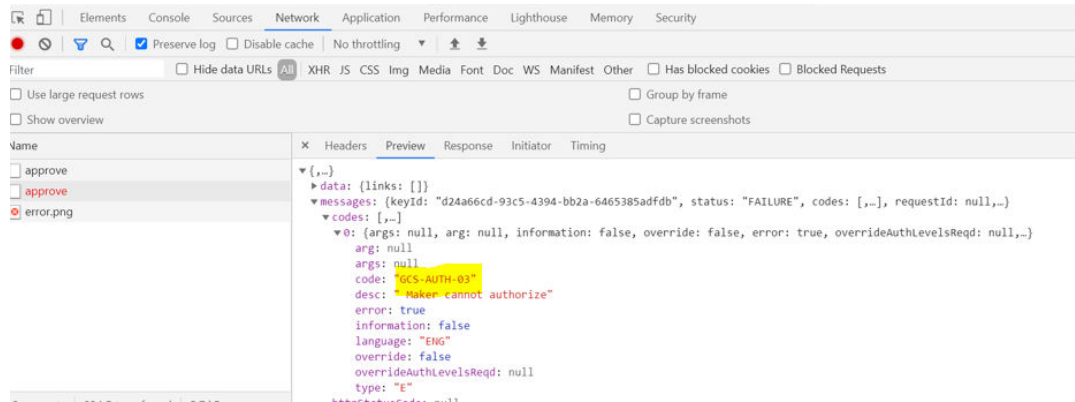This topic describes about Error Message not shown.

If there are any improper calls, check the `ERTB_MSGS` table to understand the cause of the error.

1. Press **F12**.

   The **Networks** screen is displayed.

2. Check the error code in the response.

   Query: SELECT * FROM ERTB_MSGS WHERE ERR_CODE='GCS_AUTH-03'

Figure 3-4   Error Message not Shown

3.1.2 Set Log File Path

This topic describes about Setting Log file path.

Log generation path needs to be defined in PLATO_LOGGER_PARAM_CONFIG table of PLATO schema.

Query: Select * from PLATO_LOGGER_PARAM_CONFIG;

Figure 3-5   Setting Log File Path
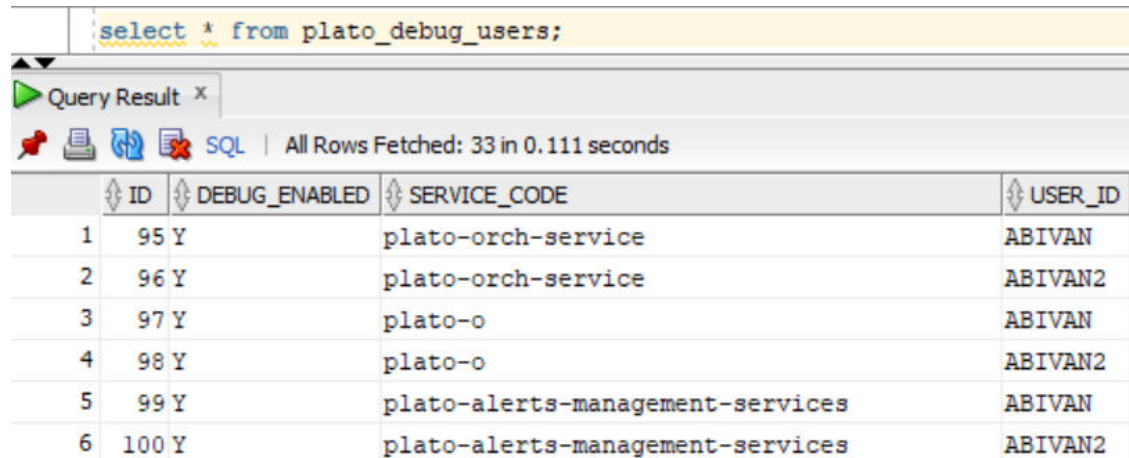
3.1.3 Dynamic Log Generation Issues

This topic describes about Dynamic log generation issues and how to work around them.

For generating dynamic service logs, insert the data to **PLATO_DEBUG_USERS** table.
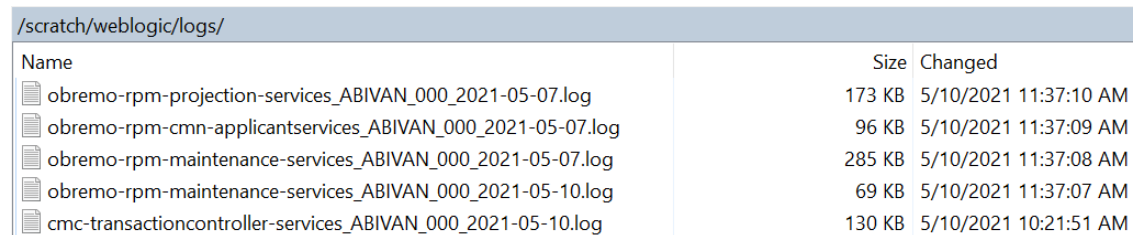
Query: Select * from PLATO_DEBUG_USERS;

Figure 3-6   Dynamic Log Generation Issues

> **ⓘ Note**
>
> Login to WINSCP and check server logs. Log files for each service will be generated based on the user_id, branch_code and date at the path provided in the `plato_logger_param_config` table.

Figure 3-7   Server Logs

3.1.4 Call is Failing in Gateway

This topic describes about the Call is failing in Gateway.

If any API call is failing in Gateway, hit the same API endpoint without passing through `api-gateway` through the postman.

Restart the specific services if required.

3.1.5 Code error in GCS side

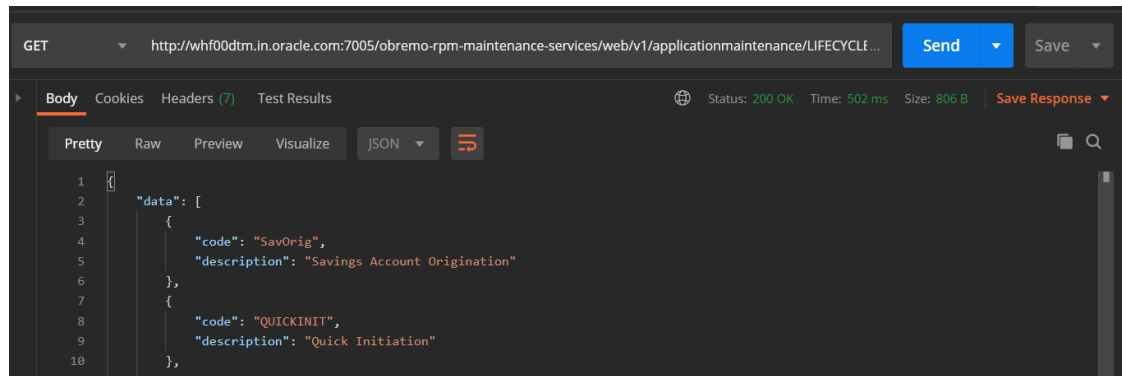This topic describes about the error code in GCS Side.

If there is any error in GCS side codes, use `java de-complier` to debug the error.

3.1.6 404 Error

This topic describes about the 404 error and how to work around them.

The possible causes for 404 error are as follows:

- Check service is not running on Eureka
- Check if service is deployed in WebLogic

3.1.7 500 internal error

This topic describes about the 500 Internal Error and how to work around them.

The possible causes for 500 internal error are as follows:

- Issue with Oracle Banking Microservices Architecture entries
- Issue with Eureka
- Service may not be up
- Issue with any peace of code

The server-side debugging is needed for the above-mentioned issues, if it is not captured in logs.

3.2 Transaction data verification

This topic provides information about the transaction data verification.

Follow the best practices mentioned below to avoid getting any errors:

- In the IN request and OUT response, verify that all the field data is going to service side.
- If there is any error related to SMS, check for the availability of SMS entries.
- Validate the endpoints and data.
- Validate the request headers passed during the API call.

- Verify that the data entered in the screen is accurate.

This topic contains the following sub-topic:

- [Troubleshoot Apply Now in Product Catalogue](#)
  This topic provides information about the issue in the Appy now button in the Product Catalogue screen.

3.2.1 Troubleshoot Apply Now in Product Catalogue

This topic provides information about the issue in the Appy now button in the Product Catalogue screen.

If **Apply Now** in Product Catalogue is failing, troubleshoot using the below points:

- Check if conductor war and `plato-orch-service` war is deployed in WebLogic.

- Check whether `PLATO-O` and `PLATO-ORCH-SERVICE` is registered in Eureka.

Figure 3-8   Plato Services

- Check whether the INITIATION workflow DSL is imported.

  1. From the Homepage, Click **Menu**. Under Menu click **Tasks**.

  2. Under **Task**, Click **Business Process Maintenance**.
     The **Workflow Maintenance** screen displays.

Figure 3-9   Workflow Maintenance
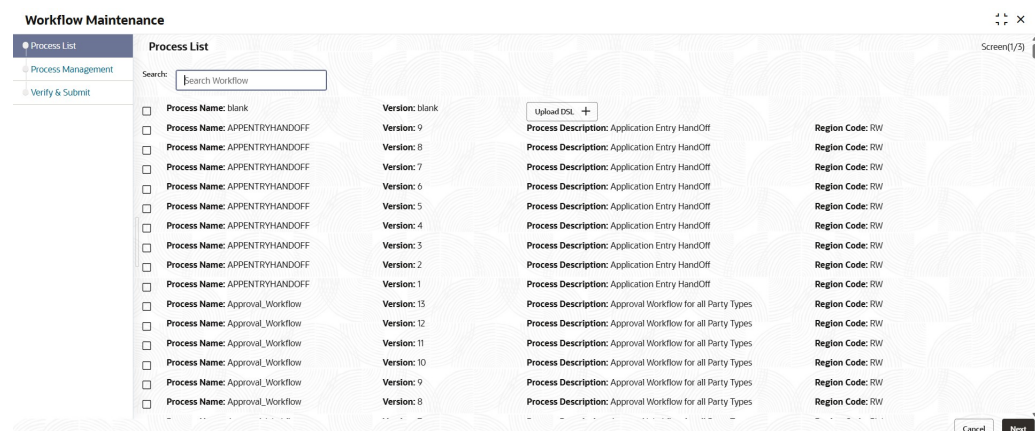
  3. Specify the Process Name in the **Search** box.
     The list of search results displays.

  4. Select the Process that wants to upload.

  5. Click the **Upload DSL** button to upload the DSL files.

- Check whether `obremo-rpm-projection-services` is up and running as this service is required during INITIATION(Apply Now).

- Check whether Sequence Generator service is up and running.

Figure 3-10   Sequence Generator Service

| SEQUENCEGENERATORSERVICE | n/a (1)   (1) | UP (1) - whf00dtm.in.oracle.com:sequencegeneratorservice:7020 |

If any API call is failing, refer to the **[Preliminary Check for UI](#)** topic in this guide.

3.3 Party Module Integration Troubleshooting

This topic describes the possible issues that may occur in Party Module integration.

The possible issues and causes are described in the following subtopics.

- [Existing Customer Details Fetch is failing](#)
  This topic provides the systematic instructions to fetch the existing customer details.

- [Customer Information Data-segment Drop-downs not Fetching](#)
  This topic describes the systematic instructions to fetch the Customer Information Data-segment Drop-downs.

## 3.3.1 Existing Customer Details Fetch is failing

This topic provides the systematic instructions to fetch the existing customer details.

In the **Customer Information** data-segment, the existing customer details is not fetching, follow the below steps:
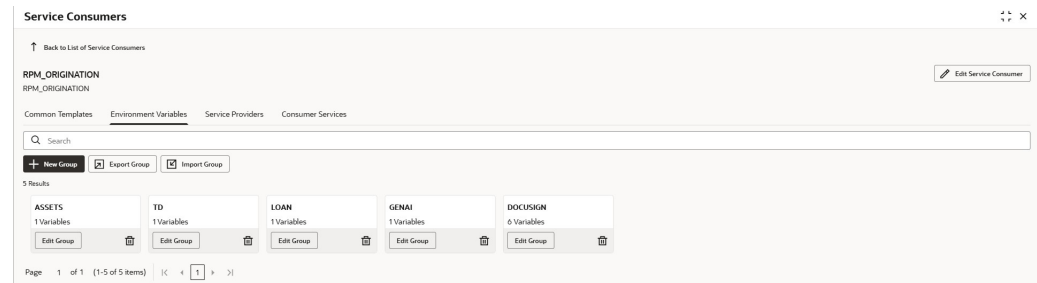
> ⓘ **Note**
>
> Refer to the **Preliminary Check for UI** topic to view if any Party API is failing.

1. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Party Module has failed.

    a. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing**.

    b. Under **Routing**, click **Service Consumers**.

    The **Service Consumers** screen displays.

    Figure 3-11   Service Consumers
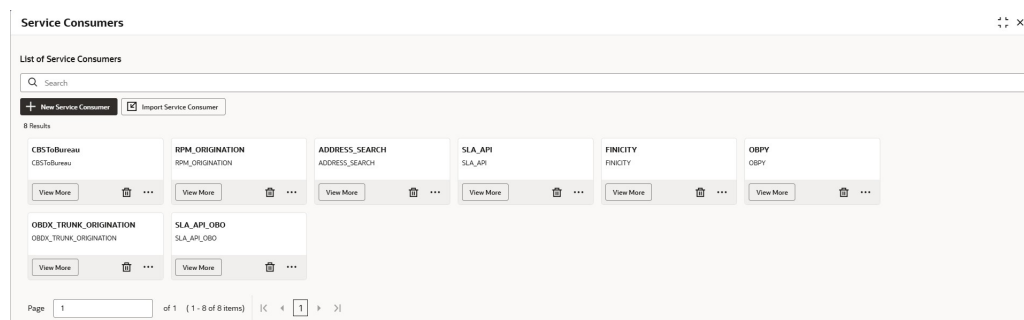
    c. On the **Service Consumers**, Click **RPM_ORIGINATION** tile.

    The **RPM_ORIGINATION** screen displays.

Figure 3-12   RPM_ORIGINATION

d. On the **RPM_ORIGINATION** screen, click the **Consumer Services** tab.

The **Consumer Services** tab displays.

Figure 3-13   Consumer Services
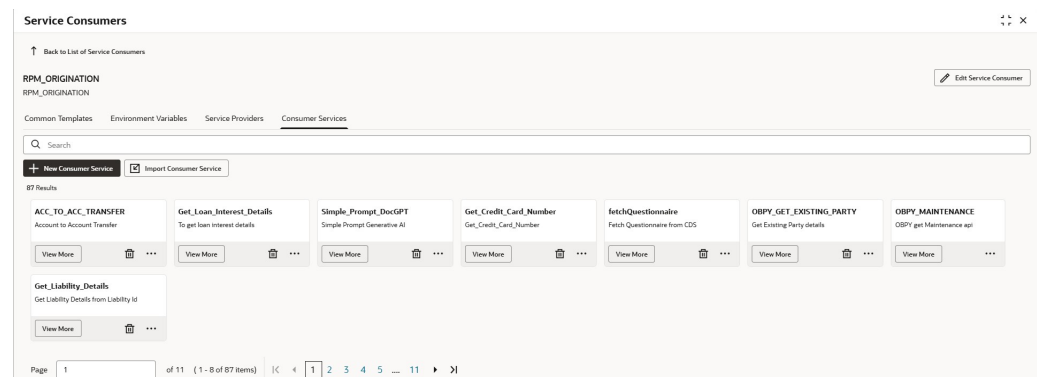
e. On the **Consumer Services** screen, click **OBPY_GET_EXISTING_PARTY** tile.

> **ⓘ Note**
>
> If any Oracle Banking Routing Hub configuration named **OBPY_GET_EXISTING_PARTY** is not found, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.

The **OBPY_GET_EXISTING_PARTY** screen displays.

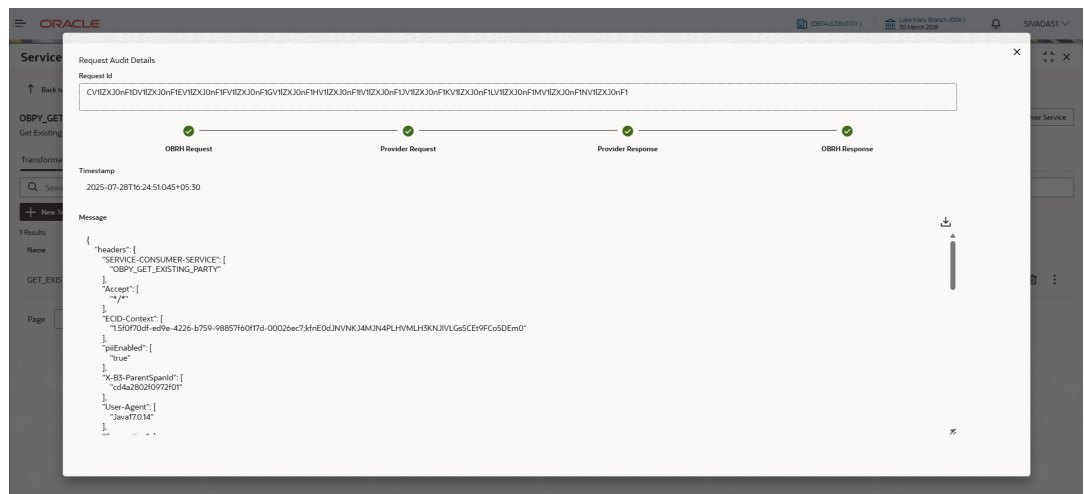**Figure 3-14   OBPY_GET_EXISTING_PARTY**
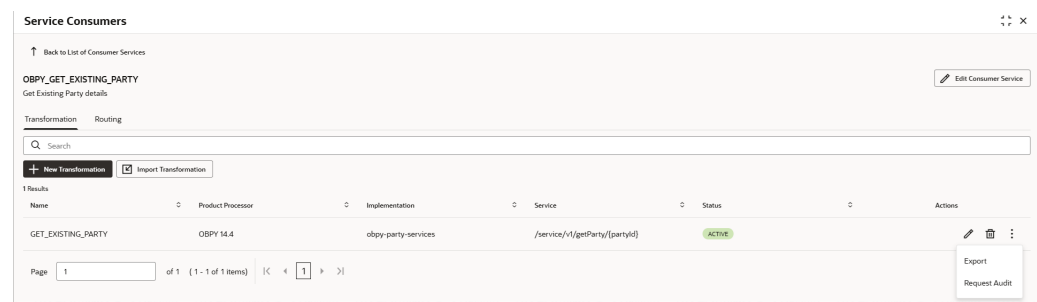
2. From the **Actions**, click on **Request Audit**.

    The **Request Audit** screen displays.

Figure 3-15   Request Audit
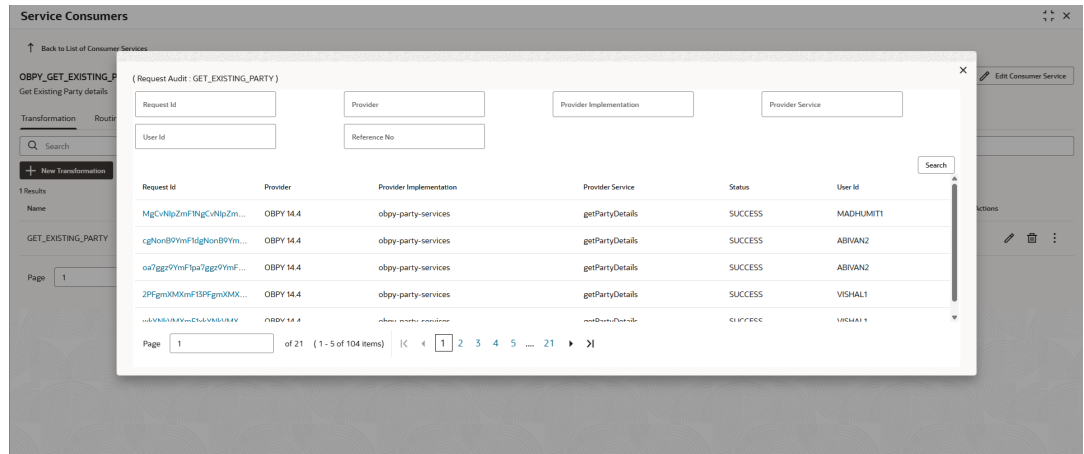
3. Check the latest **getPartyDetails** Oracle Banking Routing Hub call.

4. Click on the **Request ID** and check the **Provider Response** to check for any errors.

    The **Request Audit details** screen displays.

Figure 3-16   Request Audit details

3.3.2 Customer Information Data-segment Drop-downs not Fetching

This topic describes the systematic instructions to fetch the Customer Information Data-segment Drop-downs.

In the **Customer Information** data-segment, the Customer Information datasegment is not fetching, follow the below steps:

> ⓘ **Note**
>
> Refer to the **Preliminary Check for UI** topic to view if any Party API is failing.

1. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Party Module has failed.

   a. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing**.

   b. Under **Routing**, click **Service Consumers**.

   The **Service Consumers** screen displays.
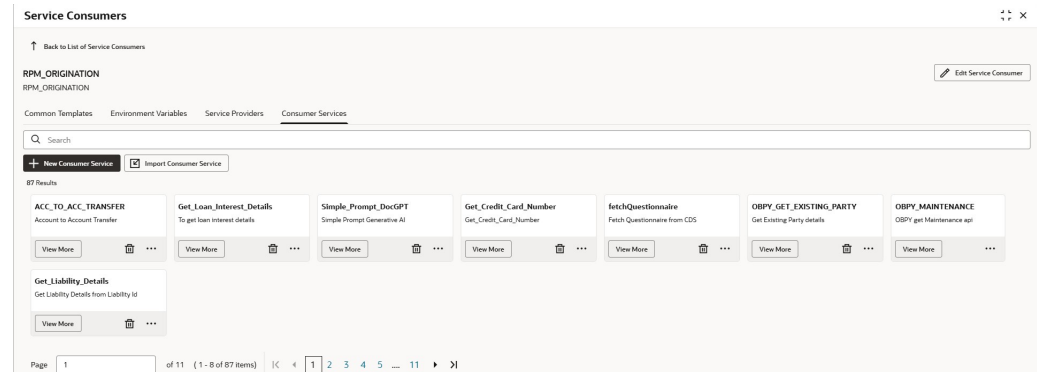
   Figure 3-17   Service Consumers
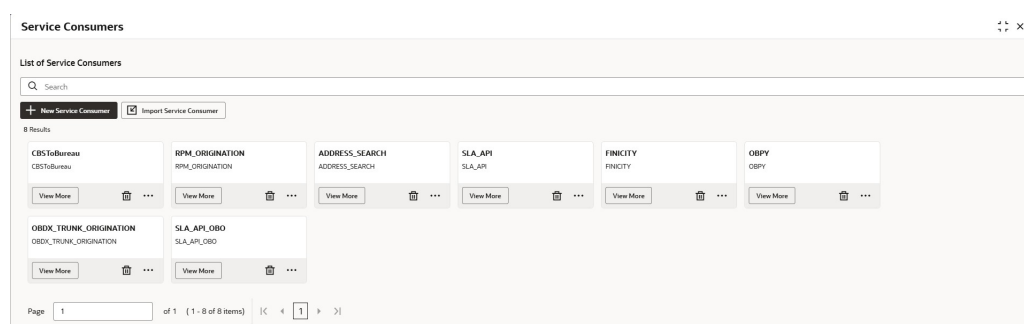
   c. On the **Service Consumers**, Click **RPM_ORIGINATION** tile.

   The **RPM_ORIGINATION** screen displays.

   Figure 3-18   RPM_ORIGINATION
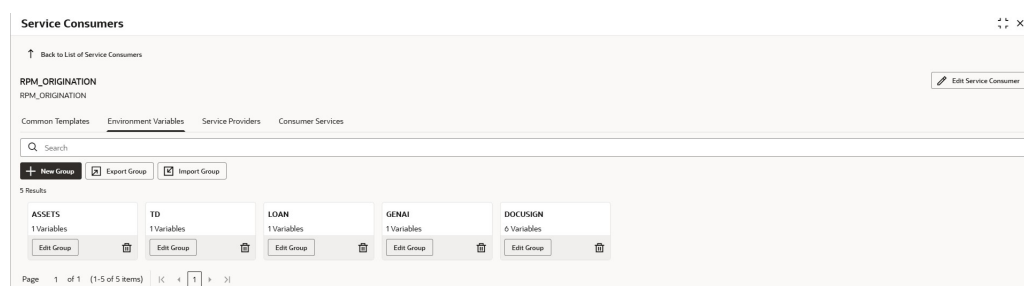
   d. On the **RPM_ORIGINATION** screen, click the **Consumer Services** tab.
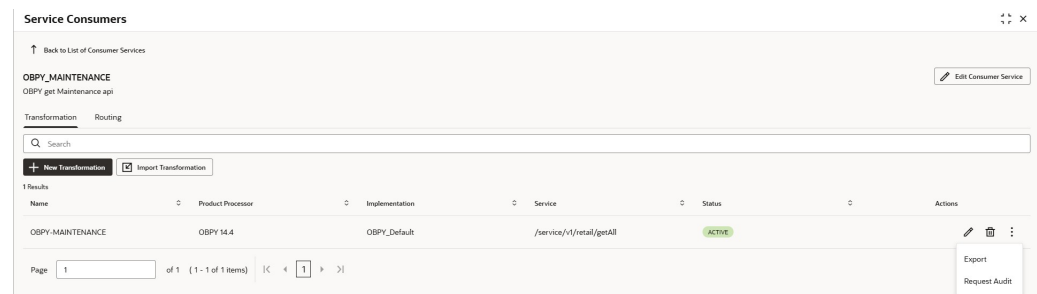
   The **Consumer Services** tab displays.

Figure 3-19   Consumer Services

e. On the **Consumer Services**, click **OBPY_MAINTENANCE** tile.

> ⓘ **Note**
>
> If any Oracle Banking Routing Hub configuration named
> **OBPY_MAINTENANCE** is not found, the Oracle Banking Routing Hub
> configurations are not fully imported. Import the Oracle Banking Routing Hub
> configuration available in the source folder.

The **OBPY_MAINTENANCE** screen displays.

Figure 3-20   OBPY_MAINTENANCE

2. From the **Actions**, click on **Request Audit**.

The **Request Audit** screen displays.

Figure 3-21   Request Audit

3. Check the latest **getPartyMaintenance** Oracle Banking Routing Hub call.

4. Click on the **Request ID** and check the **Provider Response** to check for any errors.

   The **Request Audit details** screen is displayed.

Figure 3-22   Request Audit details

3.4 FLEXCUBE Host Integration Troubleshooting

This topic describes the possible issues that may occur in FLEXCUBE Universal Banking Solution integration.

Host Calls Failure

Host call failure may be due to various reasons ranging from improper Oracle Banking Routing Hub configuration to absence of maintenance in the Oracle FLEXCUBE Universal Banking environment. Host call may fail during Business Product Host Product listing, Interest or Charge Details data-segment fetch or during Oracle FLEXCUBE Universal Banking Account creation time.
To find the root issue, follow the below steps:

1. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Oracle FLEXCUBE Universal Banking Module has failed.
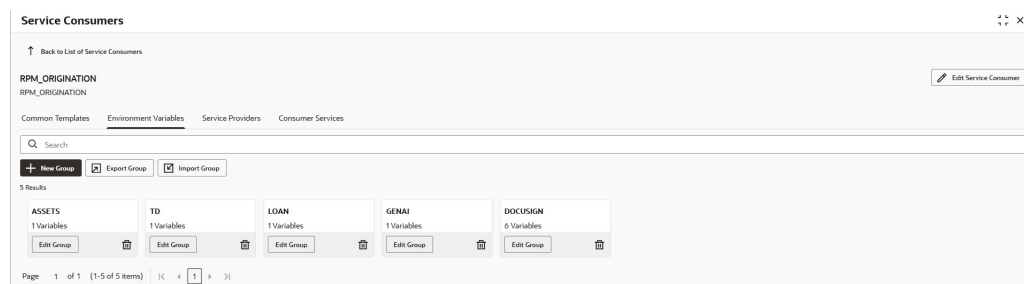
a. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing**.

b. Under **Routing**, click **Service Consumers**.

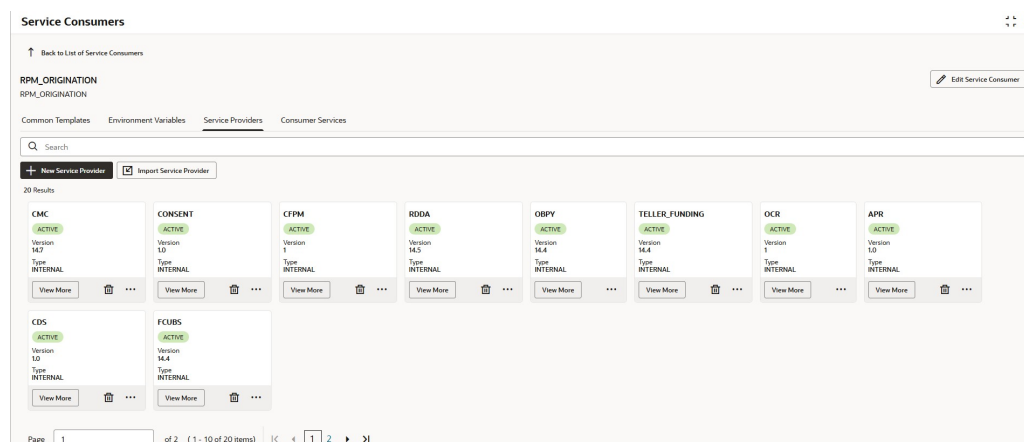The **Service Consumers** screen displays.

Figure 3-23   Service Consumers
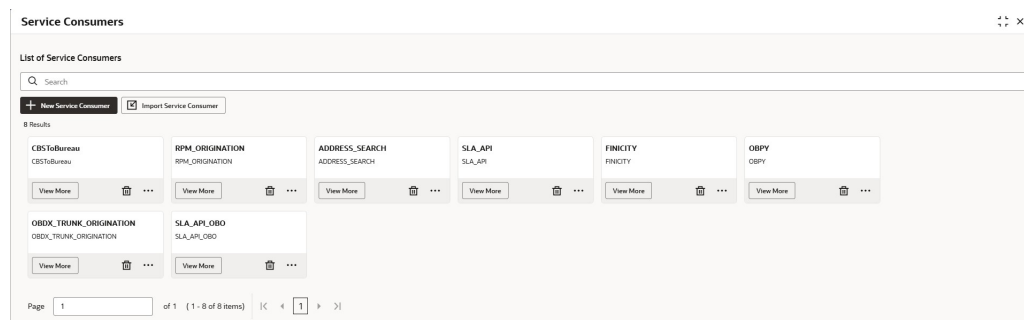
c. On the **Service Consumers**, Click **RPM_ORIGINATION** tile.

The **RPM_ORIGINATION** screen displays.

Figure 3-24   RPM_ORIGINATION

d. On the **RPM_ORIGINATION** screen, click the **Service Provider** tab.

The **Service Provider** tab displays.

Figure 3-25   Service Provider

e. On the **Service Provider** tab, click the **FCUBS** tile.

> ⓘ **Note**
>
> If you do not find any Oracle Banking Routing Hub configuration for Oracle FLEXCUBE Universal Banking, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.
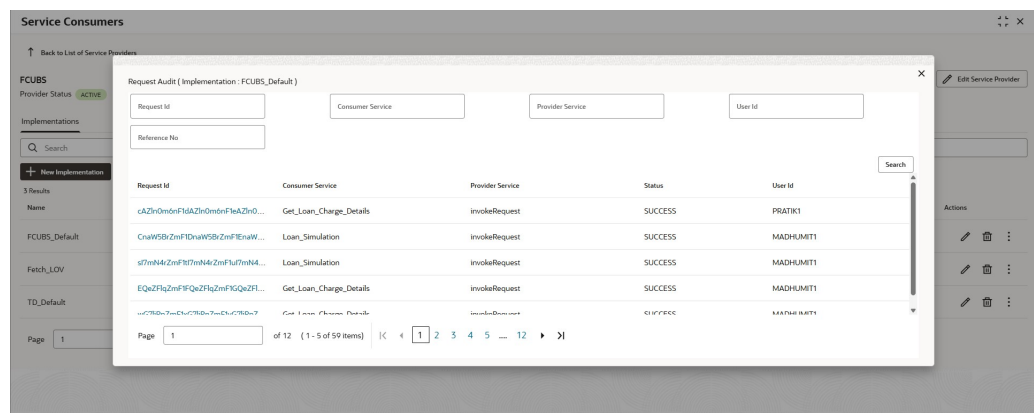
The **FCUBS** screen displays.

Figure 3-26   Oracle FLEXCUBE Universal Banking
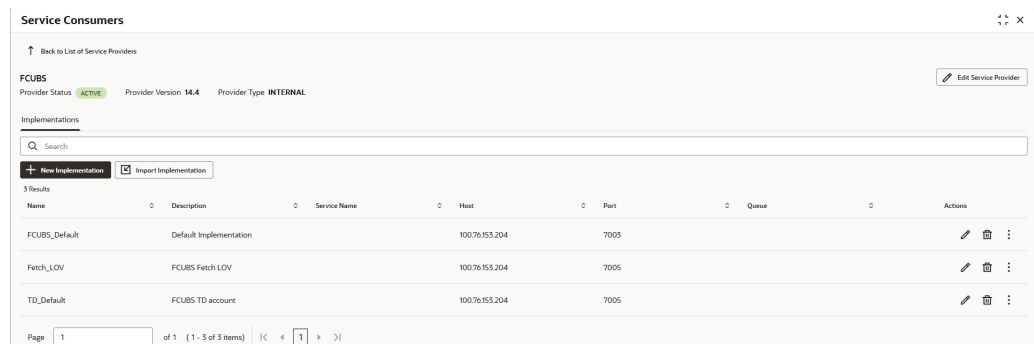
f. From the **Actions**, click on **Request Audit**.

The **Request Audit** screen displays.

Figure 3-27   Request Audit

g. Check the most recent transformation on which the operation was performed.

h. Click on the **Request ID** and check the **Provider Response** to check for any errors.

The **Request Audit details** screen displays.

Figure 3-28   Request Audit details
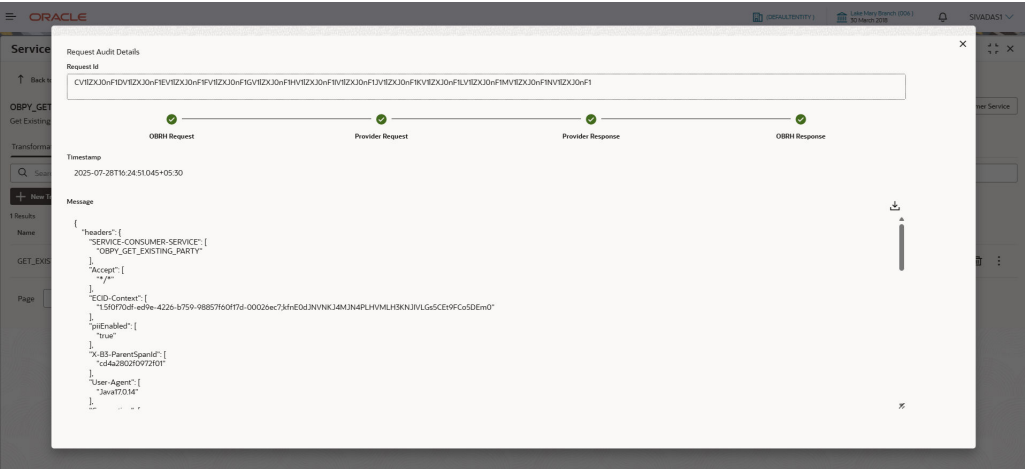
2. If there is no Oracle Banking Routing Hub call, yet the host call is failing (especially during Account Creation), the failure may be at the workflow task level. To debug this scenario, follow the steps below.

    a. Using the Application Number, call the `plato-orch-service` search API (API details given below) using Postman.

    Table 3-1   API details

API components	Description
API URL	http://whf00dtm.in.example.com:7011/plato-orch-service/api/v1/extn/custom-actions/queries/tasks?offset=0&limit=100
Body	Refer to the image below.
Headers	• **Content-Type**: application/json • **userId**: • **appId**: platoorch • **branchCode**: • **entityId**: DEFAULTENTITY

    The search result displays.

    Figure 3-29   Request Body

    b. From the response, search for **subWorkflowId**.

    The search result displays.

Figure 3-30   subWorkflowId


```
        "taskType": "SUB_WORKFLOW",
        "status": "COMPLETED",
        "inputData": {
            "workflowInput": {
                "TASK_DESCRIPTION": "Savings Origination
                    HandOff",
                "applicationDate": 1585218545000,
                "applicationNumber": "000APP000016729",
                "processRefNumber": "000INSTAS0007184",
                "branch": "000",
                "user": "ABIVAN",
                "processName": "INSTANTACCOUNT",
                "processCode": "SavOrig",
                "stage": "Account Creation",
                "stageCode": "RPM_INSTACC_HNDOFF",
                "currentBranchCode": "000"
            },
            "subWorkflowId":
                "ad194dd5-738f-4ce3-b9b9-2a9f72bb59c6",
            "subWorkflowName": "CASAHOSTORCH",
```

c. Use this `subWorkflowId` as parameter in the below API.

Table 3-2   API details

API components	Description
API URL	http://whf00dtm.in.example.com:7011/plato-orch-service/api/workflow/ad194dd5-738f-4ce3-b9b9-2a9f72bb59c6
Headers	• **Content-Type**: application/json • **userId**: • **appId**: platoorch • **branchCode**: • **entityId**: DEFAULTENTITY

The response shows the actual error for HTTP task to fail.
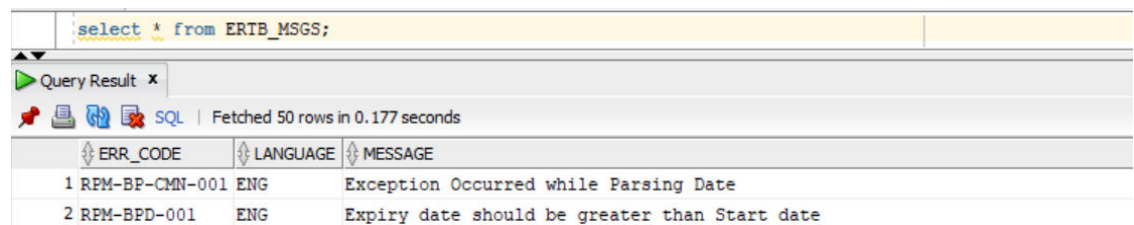
4

Business Error Codes

This topic provides information about business error codes.

The list of overrides, informational messages, and error codes that may encountered during application usage is available in the `ERTB_MSGS` table within the corresponding service schema.

For example, if an error occurs in Business Product maintenance screen and the error code displays in the table. Then, connect with Business Product schema and search for that particular error code in the `ERTB_MSGS` table.

Figure 4-1   Error Codes and Messages

Index