Oracle® Banking Origination

Troubleshooting Guide

Release 14.7.4.0.0
F97671-01
June 2024

Oracle Banking Origination Troubleshooting Guide, Release 14.7.4.0.0

F97671-01

Contents

Preface

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.

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

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.

Conventions

The following text conventions are used in this document:

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

1

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.
- Login to Zipkin
  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.
- Check if Kafka is Running
  This topic provides information about Kafka is Running.
- Troubleshooting Environmental Issues
  This topic describes about the troubleshooting environmental issues.

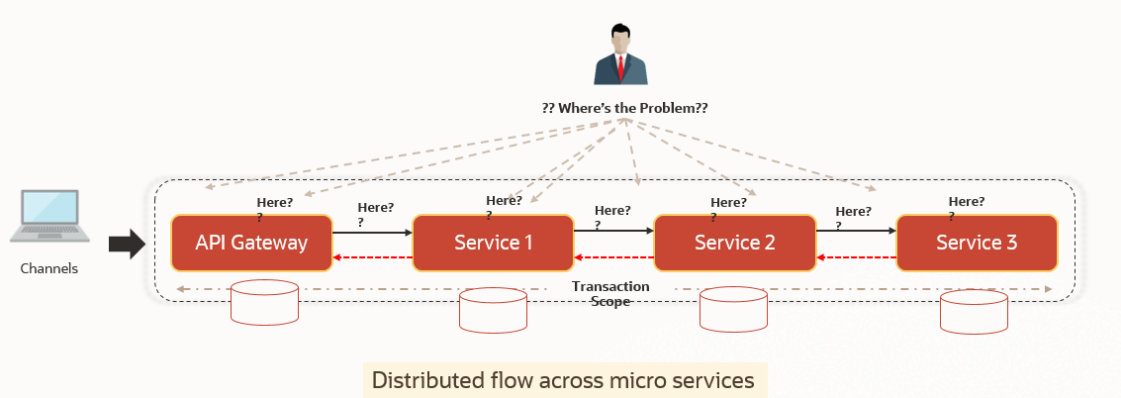
1.1 Where is the Problem

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 corroborate the problem.

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.

Figure 1-1 Distributed Flow across Micro Services

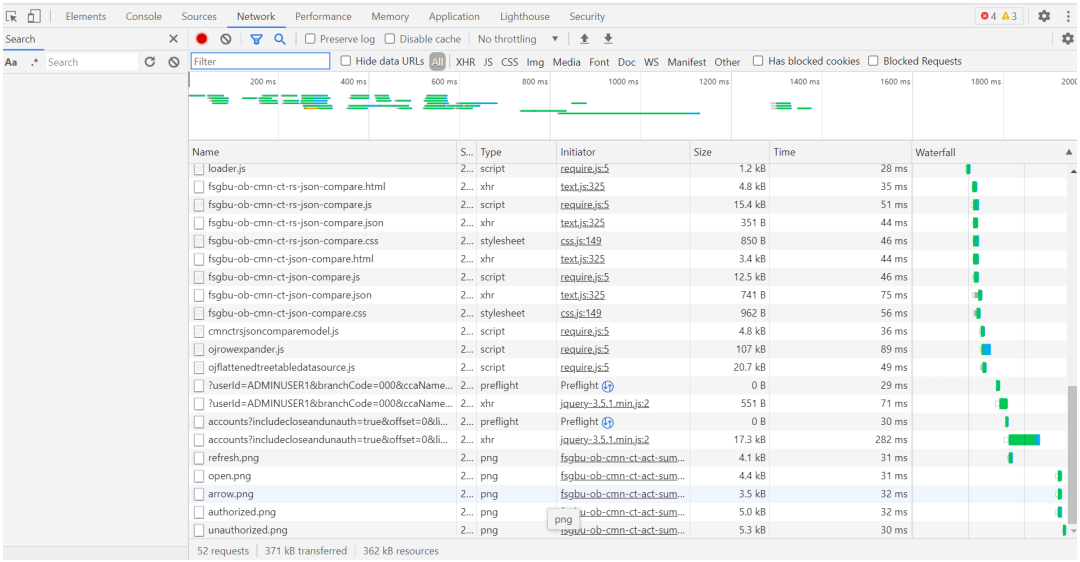


1.2 Preliminary Checks from UI

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

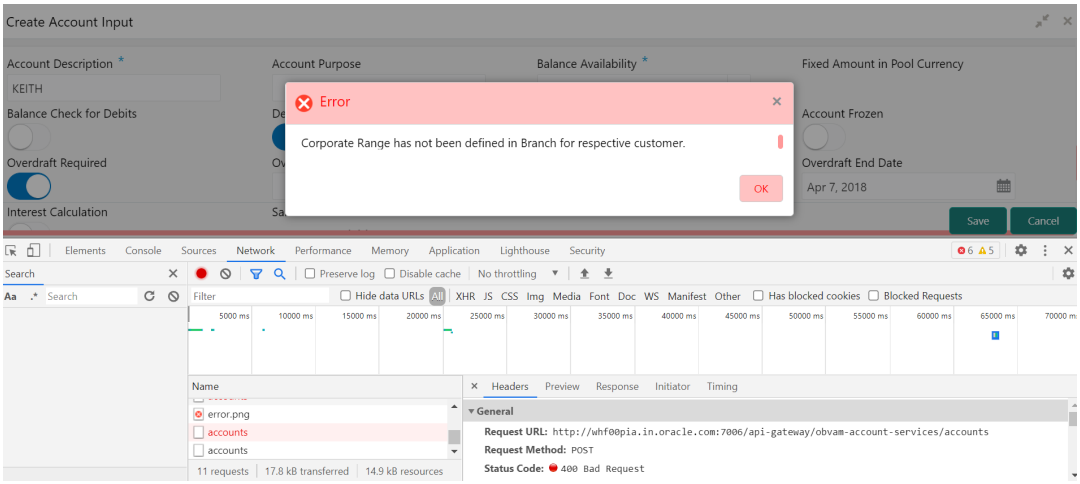
1. Launch the application with delegated URL.
2. Press **F12** key and select **Inspect and See network**.
3. Verify that all the call responses are successful.

Figure 1-2 Call Responses



> **Note:**
>
> Usually red color indicates a non-2xx HTTP response.

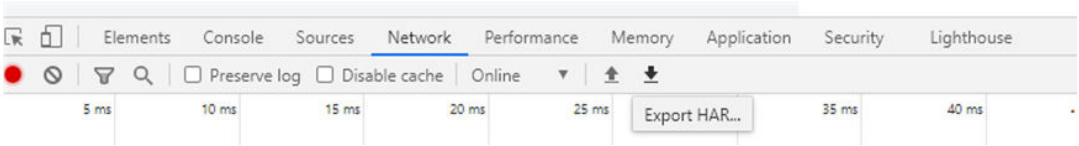
Figure 1-3 Non-2xx Response



4. Export the trace using the **Export** in browsers.

   Example: The user can see the export option as shown below in Chrome.

Figure 1-4 Export Option



> **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.

> **Note:**
>
> 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.

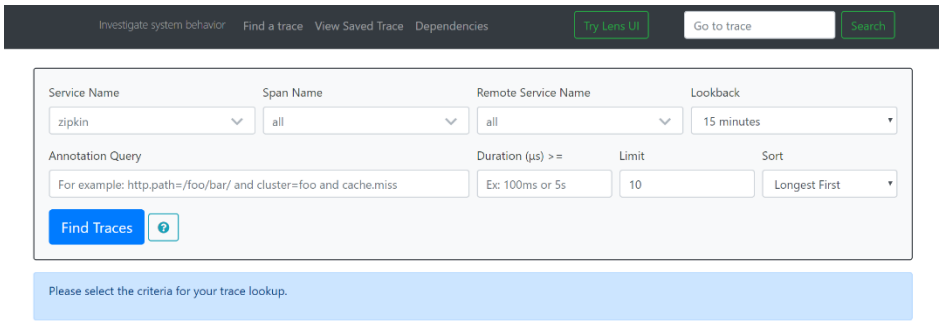
1.4 Login to Zipkin

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

1. Launch the Zipkin URL.

> **Note:**
>
> The basic layout of Zipkin displays.

Figure 1-5 Layout of Zipkin

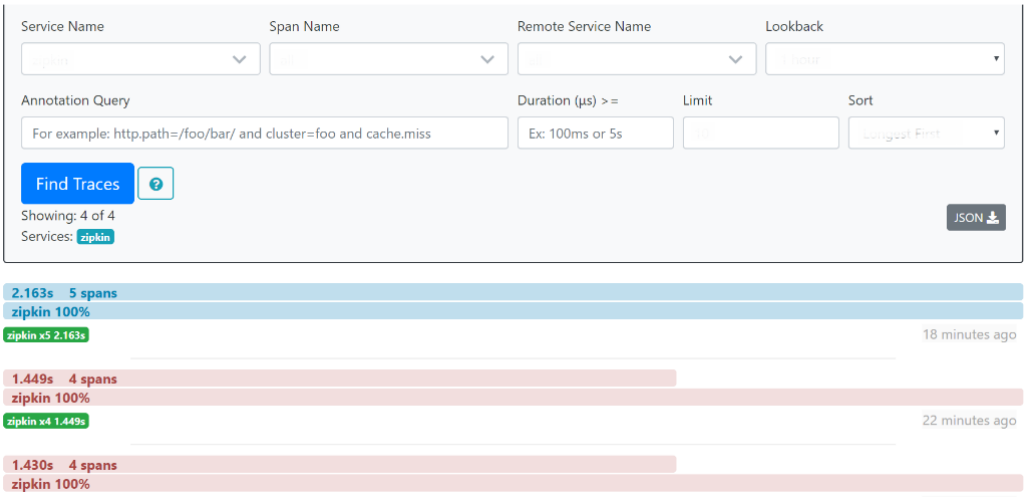


2. Use **Search** to find the traces of required API calls and services.

> **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 below figure shows the list of traces.

Figure 1-6 List of Traces



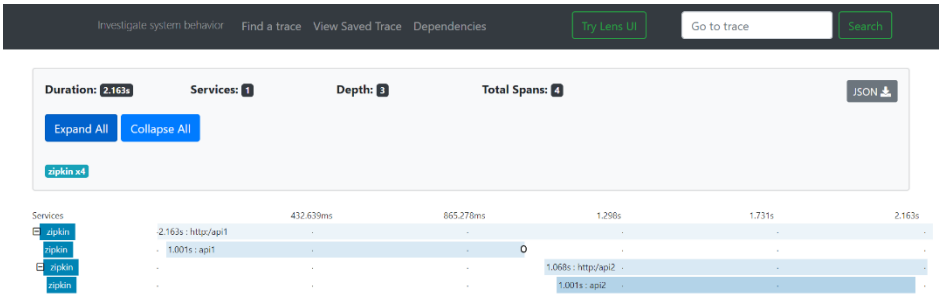
3. Open the individual trace.

   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.

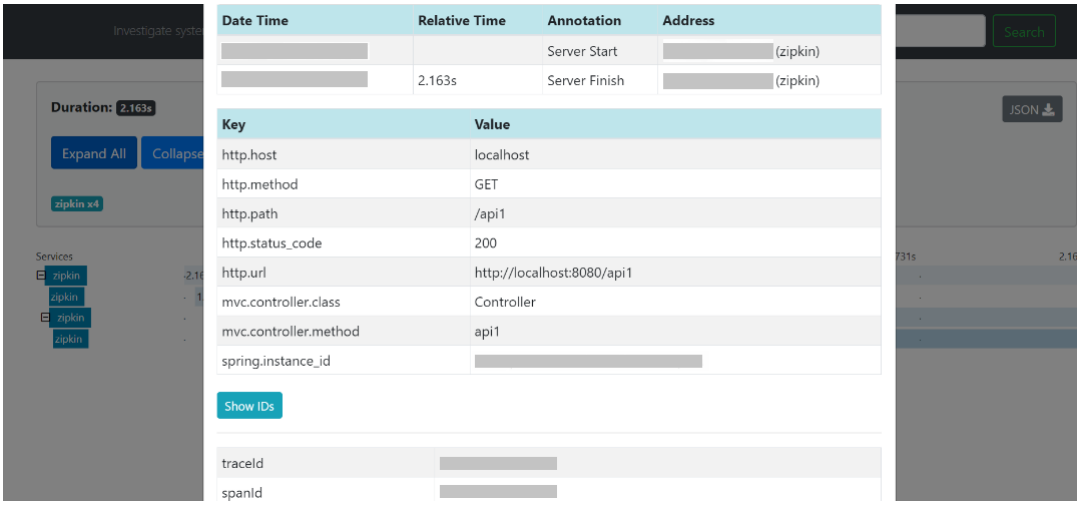
   The details of an individual trace displays.

Figure 1-7 Individual Trace



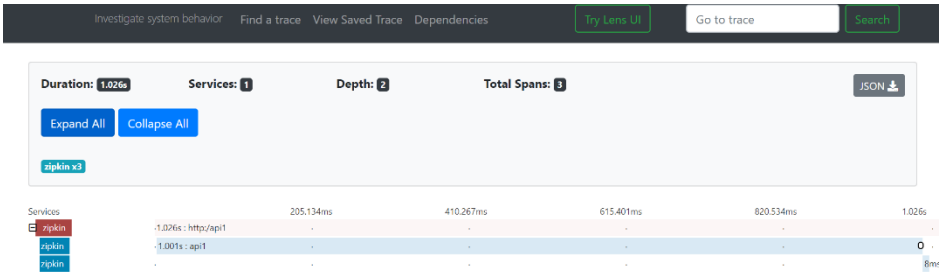
4. Click on the individual block to display the details.

Figure 1-8 Details of Individual Block



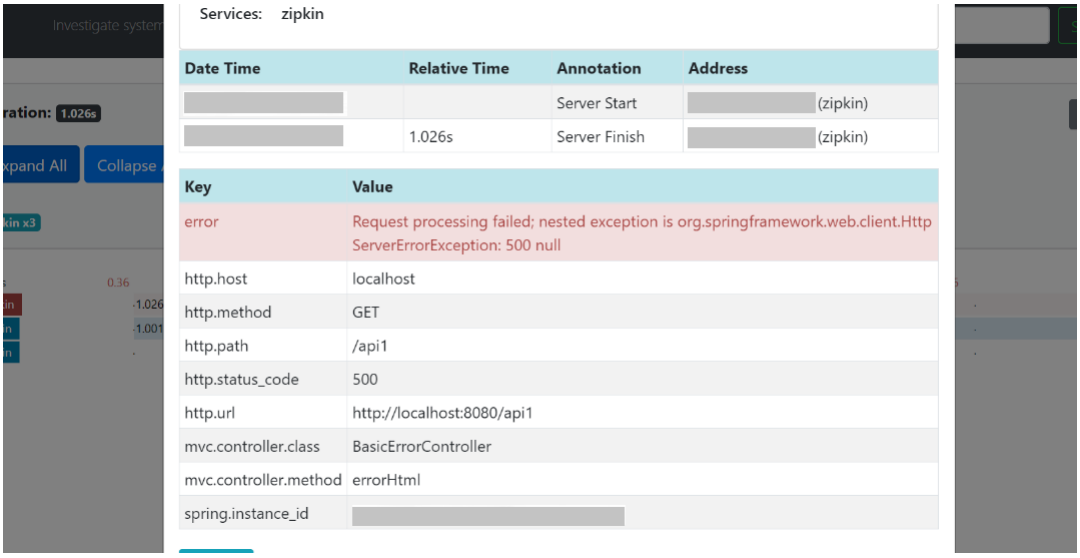
The user can also view the logging events in the Zipkin UI as small circular blocks. An example of an error log is shown below.

Figure 1-9 Sample Error Log



5. Click the error to get clear details and place of the error.

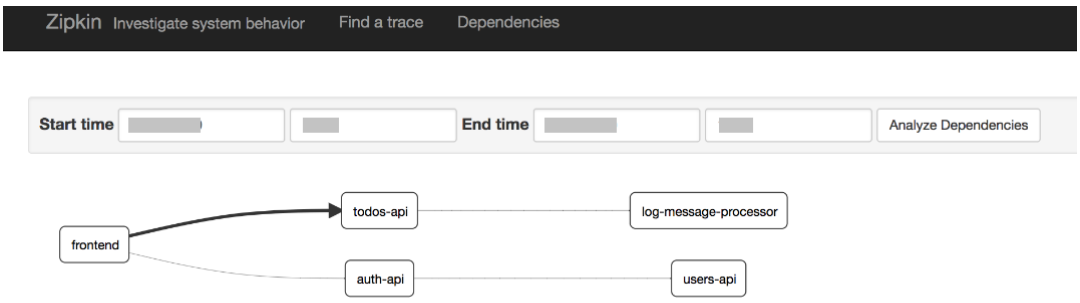
Figure 1-10 Details of Error



> **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** to get the dependency graph information between micro-services.

Figure 1-11 Sample Dependency Graph



1.5 Troubleshooting Logs using ELK Stack

This topic describes about Troubleshooting Logs using ELK Stack.

This topic contains the following subtopics:

- Set Up ELK
  This topic provides the links to setup ELK.

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

1.5.1 Set Up ELK

This topic provides the links to setup ELK.

1. Download the Elastic search from https://www.elastic.co/downloads/elasticsearch**https://www.elastic.co/downloads/elasticsearch**.
2. Download the Kibana from https://www.elastic.co/downloads/kibana**https://www.elastic.co/downloads/kibana**.
3. Download the Logstash from https://www.elastic.co/downloads/logstash**https://www.elastic.co/downloads/logstash**.

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

Step to run ELK:

4. 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 &**
5. Configure the Kibana to point the running instance of elastic search in the `kibana.yml` file.

Figure 1-12 Logstash Configuration

```
# 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
```

6. Follow the below steps to configure the Logstash.

a. **Input**: This configuration is required to provide the log file location for the Logstash to read from.

b. **Filter**: Filters in Logstash is basically used to control or format the read operation (Line by line or Bulk read).

c. **Output**: This provides the running elastic search instance to send the data for persisting.

Figure 1-13 Kibana

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 1-14 Kibana

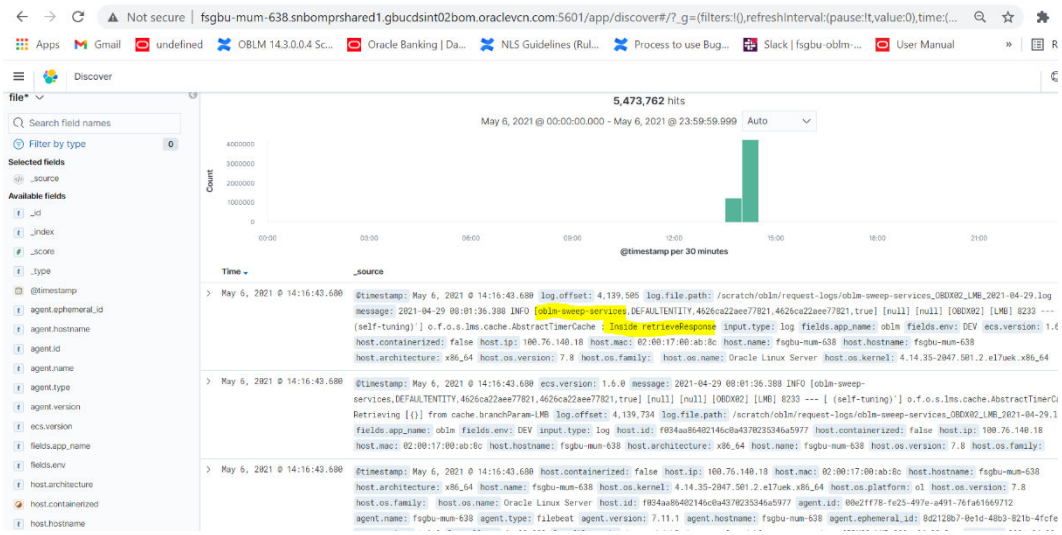
```
  #Grokking Spring Boot's default log format
  grok {
    match => [ "message", "%{TIMESTAMP_ISO8601:timestamp}\s+%{LOGLEVEL:severity}\s+\[%{DATA:service},%{DATA:trace},%{DATA:span}
  }


  #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** you should be able to see the Kibana console UI. Kibana needs elasticsearch to be UP as it creates indexes & fetches logs from it.
5. Start the nohup bin/kibana &

Figure 1-15 Kibana



1.6 Check if Kafka is Running

This topic provides information about Kafka is Running.

1. Run the cmd $ **netstat –tlnp | grep :9092**.

> **Note:**
>
> 9092 is default port of kafka.

Possible issue while starting kafka

2. Kafka is not starting may be because zookeeper is not yet started.
   - Run the cmd $ **netstat –tlnp | grep :2181**.

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

if any services is not running on this port means, zookeeper is down.

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

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

> **Note:**
>
> Some references that can be useful https://docs.cloudera.com/documentation/kafka/latest/topics/kafka_faq.html

1.7 Troubleshooting Environmental Issues

This topic describes about the troubleshooting environmental issues.

This topic contains the following subtopics:

- Possible Issues While Deploying Services
  This topic describes the possible issues that may occur in the environment.
- Possible Issues While Logging in and Launching Screen
  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.

This subsection describes the possible issues that may occur in the environment.

Service deployment is failing due to flyway

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.

You may check **flyway_schema_history** table of the respective schema for finding the flyway script entries.

Other possible issues

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-5.3.0.war`.

1.7.2 Possible Issues While Logging in and Launching Screen

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

Login Page is not Launching

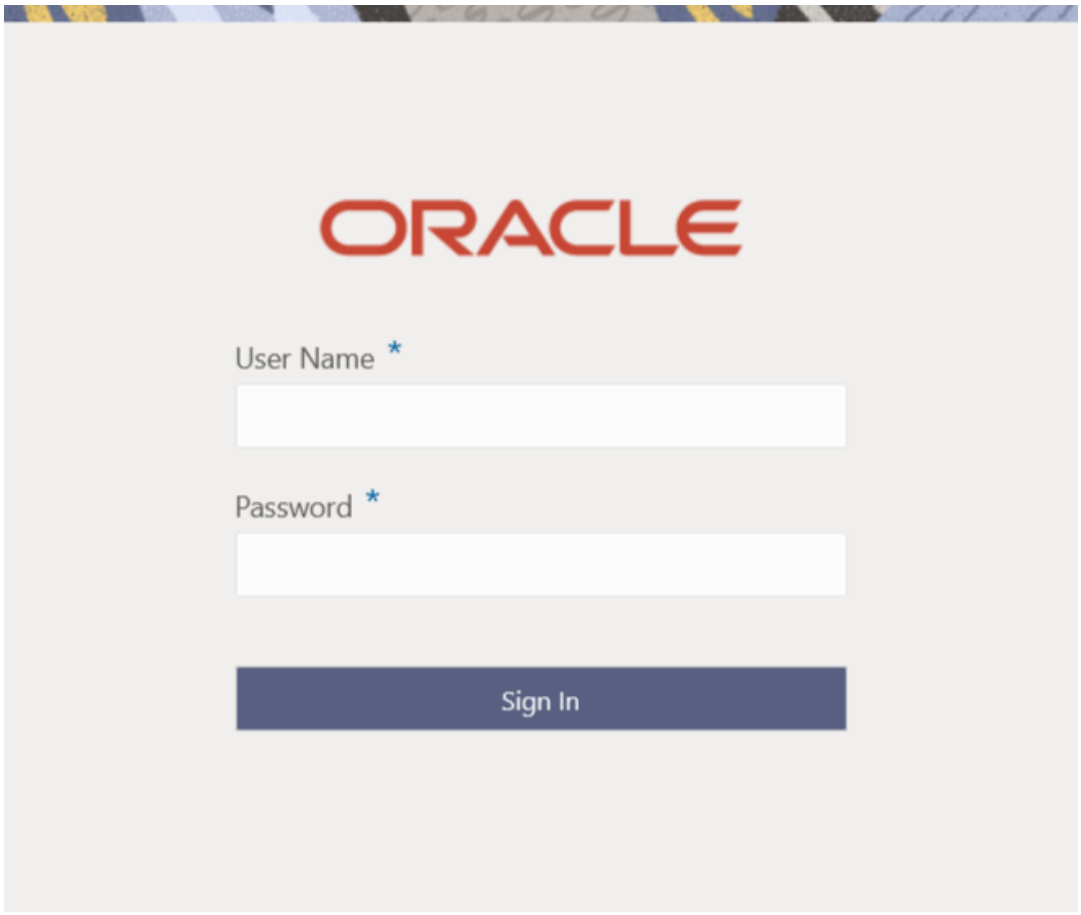
Perform the following checks if the login page is not launching.

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, obvam-component-server etc are also deployed along with the app-shell.

   Example: `http://<ip-address>:<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 1-16 Sign In

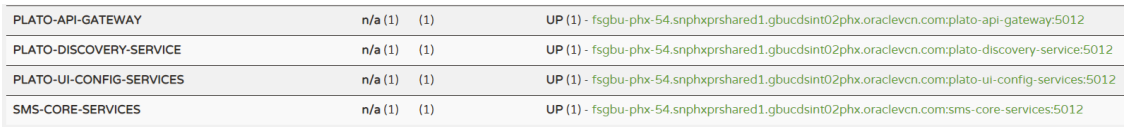


Unable to login after launching the application

Perform the following check if you are 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 1-17 Services



Unable to login after restarting the services

Perform the following check if you are 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.

Retail Banking menus are not displayed after logging in

After you log in, if the Retail Banking menus are not displayed, 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.

Screens are not launching after logging in

If you are not able to launch the screens 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:

2.1 WebLogic

This topic describes about the Weblogic details.

This topic contains the following subtopics:

2.2 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**.
2. On the **Summary of JDBC Data Sources** screen, click **New** and add the data source providing the required details.

Figure 2-2 Summary of JDBC Data Sources

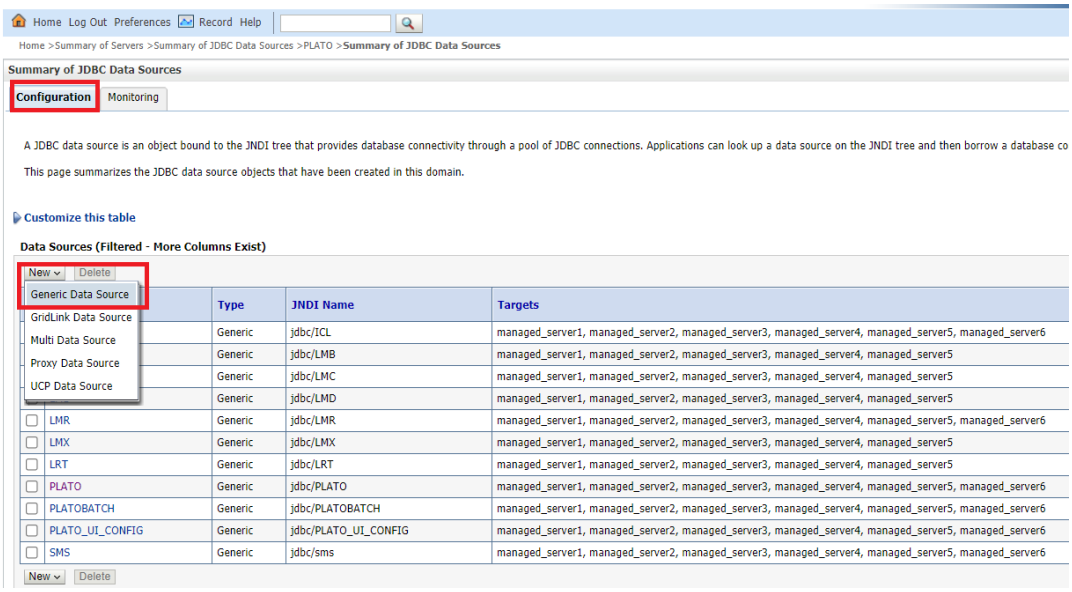
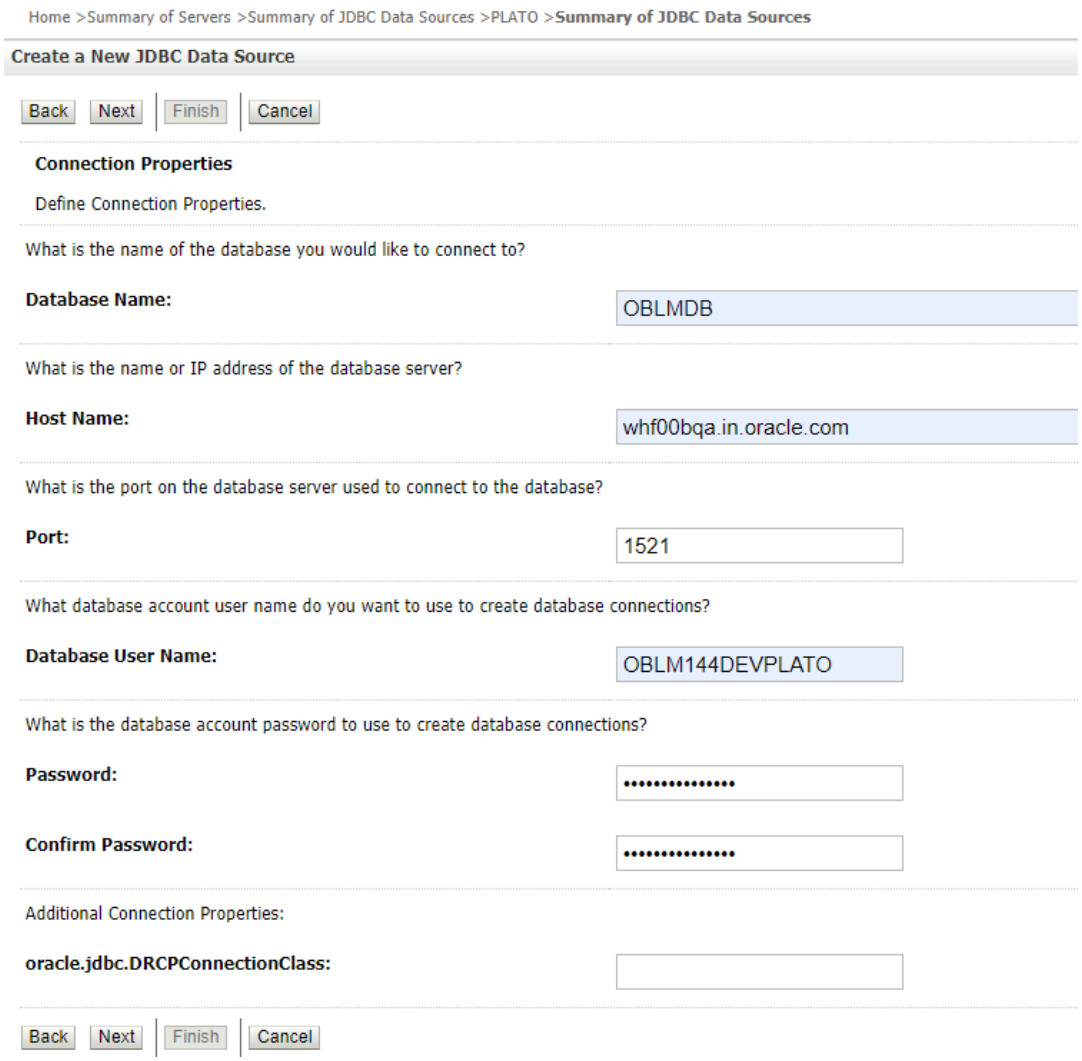


Figure 2-3 Create a New JDBC Data Source

Figure 2-4 Create a New JDBC Data Source



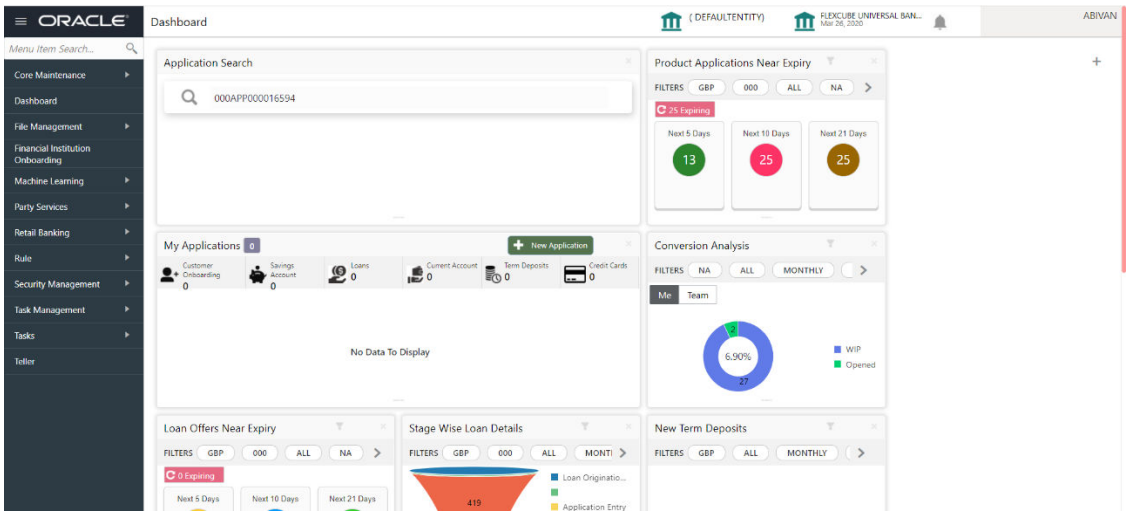
3

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



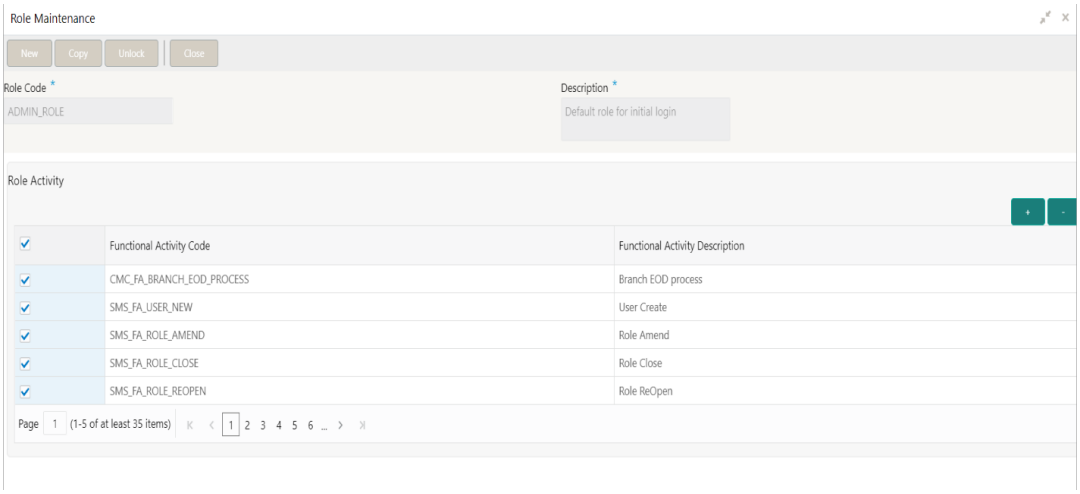
User Role Issues

Role Profile includes access rights to the functional activities that are common to a group of users. A user can be linked to a Role Profile by which you give the user access rights to all the functional activities in the Role Profile.

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

- On **Security Management**, click **Role** screen.

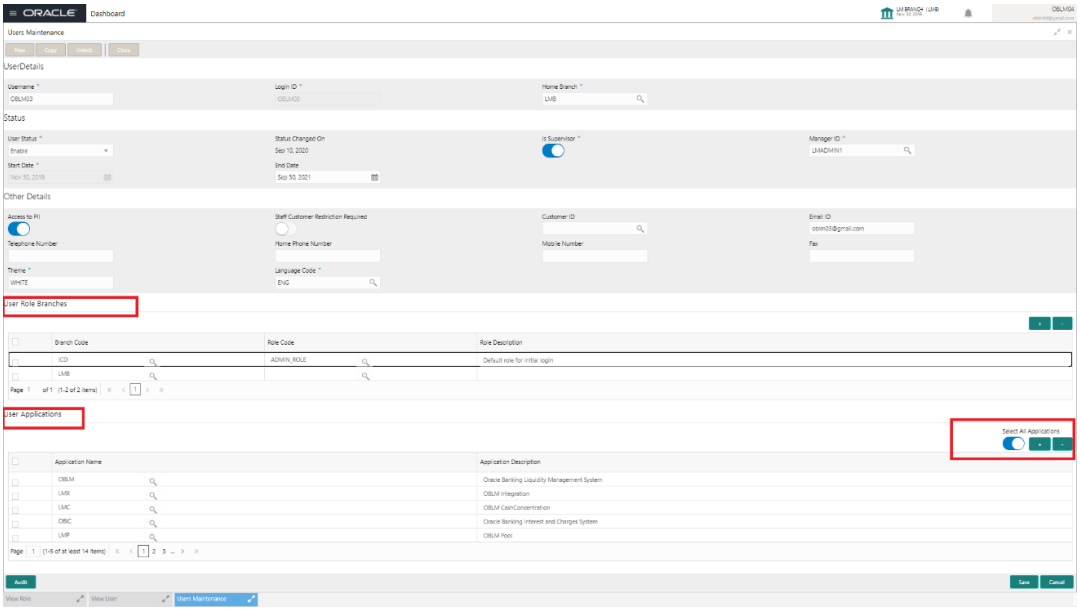
Figure 3-2 Role Maintenance



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

- On **Security Management**, click **User** screen.

Figure 3-3 Users Maintenance



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

- First level issues
  This topic provides information about the first level issues.
- 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

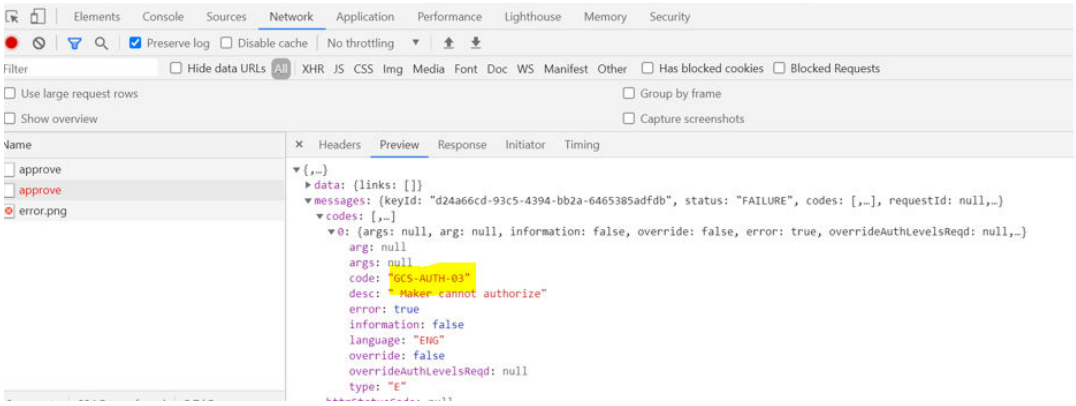
This topic provides information about the first level issues.

Error Message not Shown

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

1. Press **F12** to open the **Networks**.
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

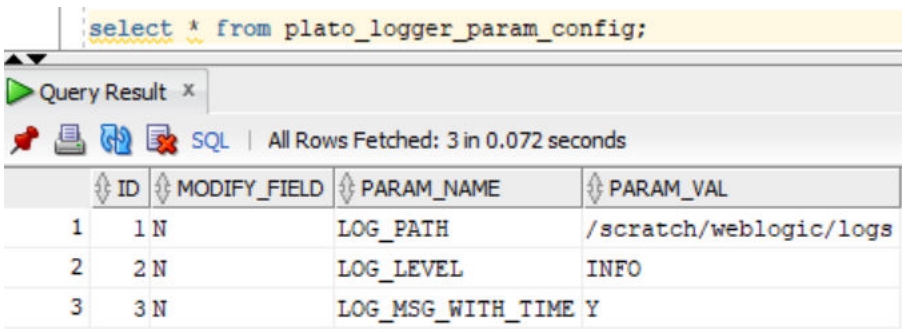


Setting Log File Path

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



	ID	MODIFY_FIELD	PARAM_NAME	PARAM_VAL
1	1	N	LOG_PATH	/scratch/weblogic/logs
2	2	N	LOG_LEVEL	INFO
3	3	N	LOG_MSG_WITH_TIME	Y

Dynamic Log Generation Issues

For generating dynamic service logs, insert the data to **PLATO_DEBUG_USERS** table.

Figure 3-6 Dynamic Log Generation Issues

```
select * from plato_debug_users;
```

Query Result

SQL | All Rows Fetched: 33 in 0.111 seconds

	ID	DEBUG_ENABLED	SERVICE_CODE	USER_ID
1	95	Y	plato-orch-service	ABIVAN
2	96	Y	plato-orch-service	ABIVAN2
3	97	Y	plato-o	ABIVAN
4	98	Y	plato-o	ABIVAN2
5	99	Y	plato-alerts-management-services	ABIVAN
6	100	Y	plato-alerts-management-services	ABIVAN2

Query: Select * from PLATO_DEBUG_USERS;

> **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

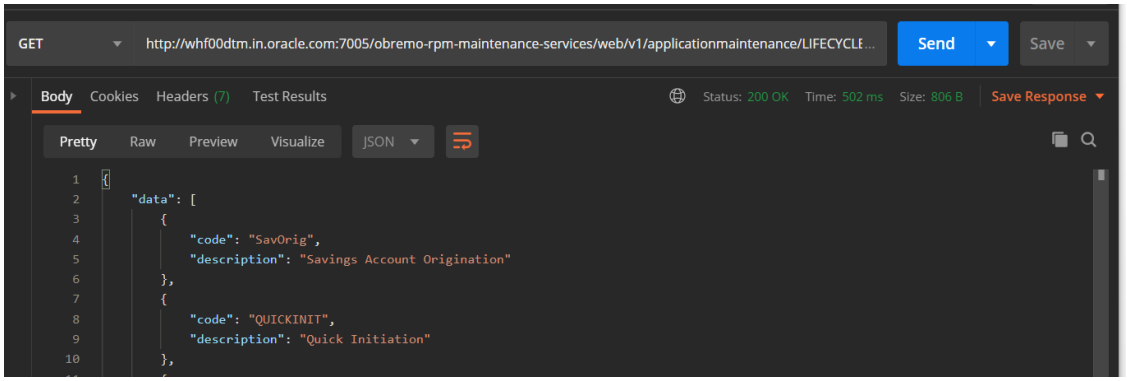
/scratch/weblogic/logs/

Name	Size	Changed
obremo-rpm-projection-services_ABIVAN_000_2021-05-07.log	173 KB	5/10/2021 11:37:10 AM
obremo-rpm-cmn-applicantservices_ABIVAN_000_2021-05-07.log	96 KB	5/10/2021 11:37:09 AM
obremo-rpm-maintenance-services_ABIVAN_000_2021-05-07.log	285 KB	5/10/2021 11:37:08 AM
obremo-rpm-maintenance-services_ABIVAN_000_2021-05-10.log	69 KB	5/10/2021 11:37:07 AM
cmc-transactioncontroller-services_ABIVAN_000_2021-05-10.log	130 KB	5/10/2021 10:21:51 AM

Call is Failing in Gateway

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

Figure 3-8 Call is Failing in Gateway



> **Note:**
>
> Restart the specific services if required.

Code error in GCS side

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

404 error

The possible causes for 404 error are as follows:

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

500 internal error

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:

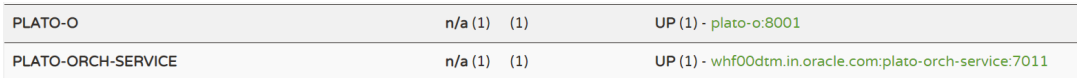
- In the IN request and OUT response, verify that all the field data is going to service side.
- If there is any error related to SMS, check for the availability of SMS entries.
- Validate the endpoints and data.
- Validate the request headers passed during the API call.

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

Apply Now is Failing in Product Catalogue

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.



- Check whether the INITIATION workflow DSL is imported.
  - Front-End Menu: Tasks Business Process Maintenance Search for INITIATION workflow
- 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.

> **Note:**
>
> Refer **Preliminary Check for UI** topic to see if any API call is failing

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 subsections:

Existing Customer Details Fetch is failing

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

If in **Customer Information** data-segment, the existing customer details is not fetching, follow the below steps:

> **Note:**
>
> Refer to **Preliminary Check for UI** to see if any Party API is failing.

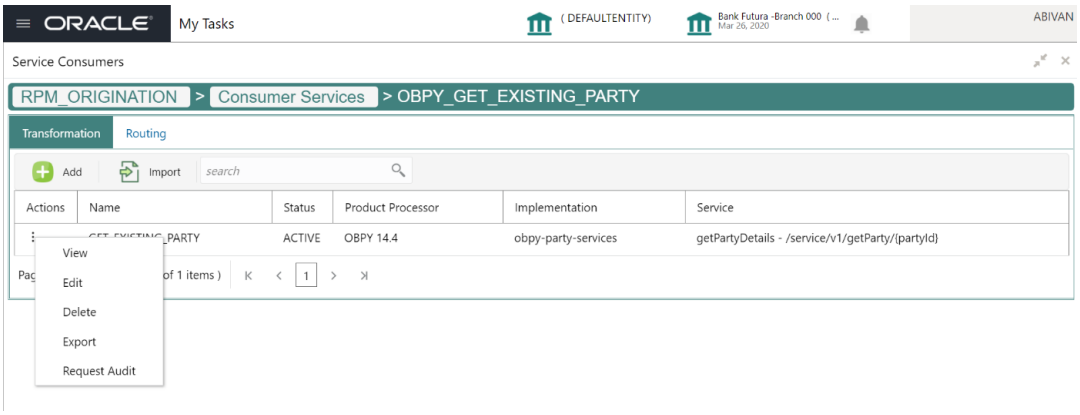
1. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Party Module has failed.
   a. On **Home** screen, click **Core Maintenance**. Under **Core Maintenance**, click **Routing**.
   b. Under **Routing**, click **Service Consumers**. Under **Service Consumers**, Click **RPM_ORIGINATION**.

c. Under **RPM_ORIGINATION**, click **Consumer Services**. Under **Consumer Services**, click **OBPY_GET_EXISTING_PARTY**

> **Note:**
>
> If you do not find any Oracle Banking Routing Hub configuration named **OBPY_GET_EXISTING_PARTY**, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.

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

Figure 3-9 Service Consumers



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.

Figure 3-10 Request Audit

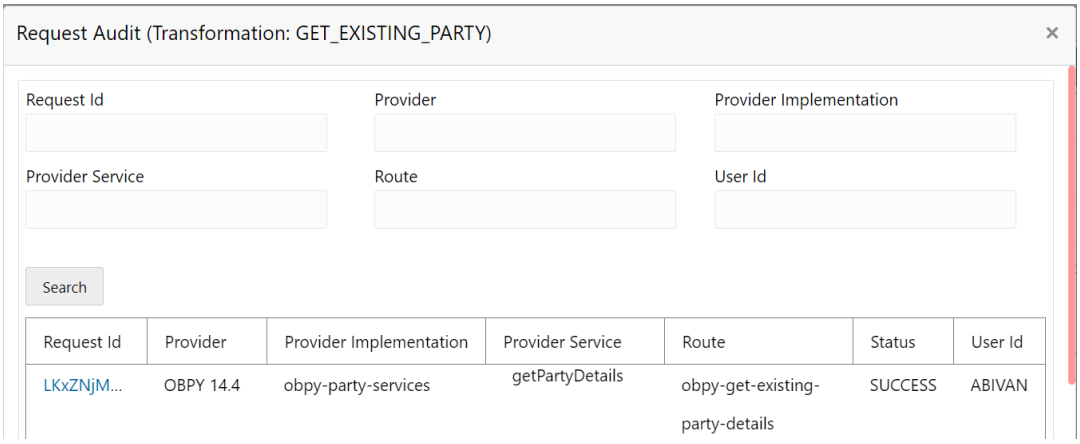
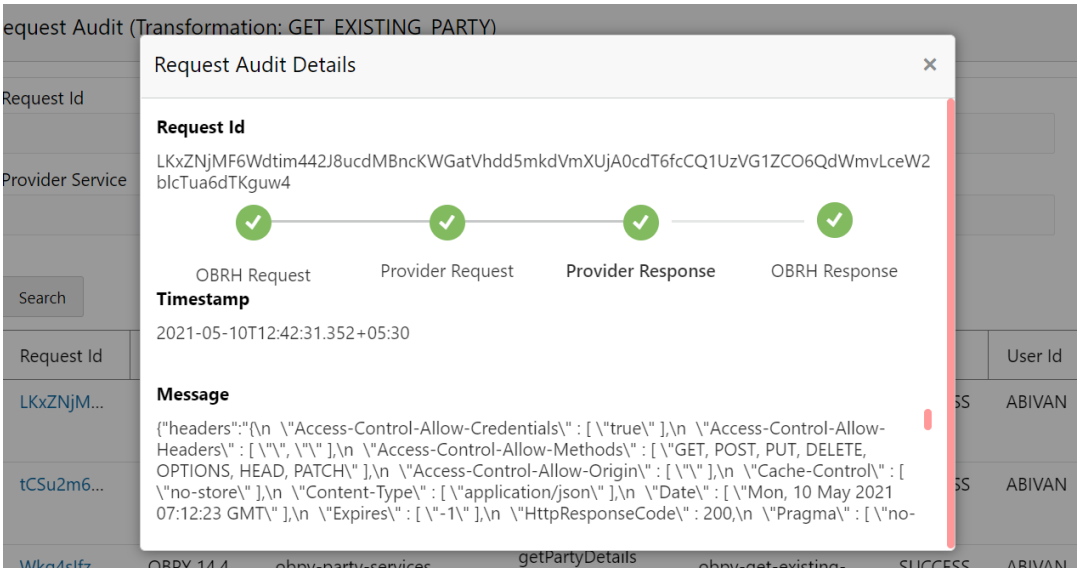


Figure 3-11 Request Audit Details



Customer Information Data-segment Drop-downs not Fetching

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

If in **Customer Information** data-segment, the existing customer details is not fetching, follow the below steps:

> **Note:**
>
> Refer to **Preliminary Check for UI** to see 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**. Under **Service Consumers**, Click **RPM_ORIGINATION**.
   c. Under **RPM_ORIGINATION**, click **Consumer Services**. Under **Consumer Services**, click **OBPY_GET_EXISTING_PARTY**

   > **Note:**
   >
   > If you do not find any Oracle Banking Routing Hub configuration named **OBPY_GET_EXISTING_PARTY**, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.

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

3.4 FLEXCUBE Host Integration Troubleshooting

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

The possible issues and causes are described in the following subsections:

Host Calls Failing

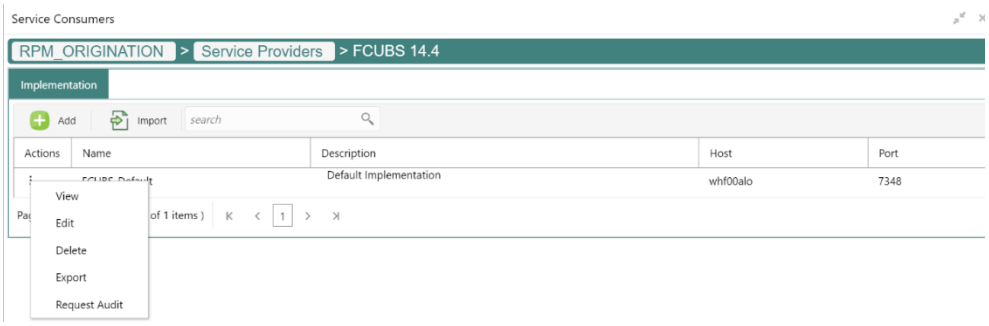
Host call failure may be due to various reasons ranging from improper Oracle Banking Routing Hub configuration to absence of maintenance in the Oracle FLEXCUBE Universal Banking environment. Host call may fail during Business Product Host Product listing, Interest or Charge Details data-segment fetch 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**. Under **Service Consumers**, Click **RPM_ORIGINATION**.
   c. Under **RPM_ORIGINATION**, click **FCUBS**.

> **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.

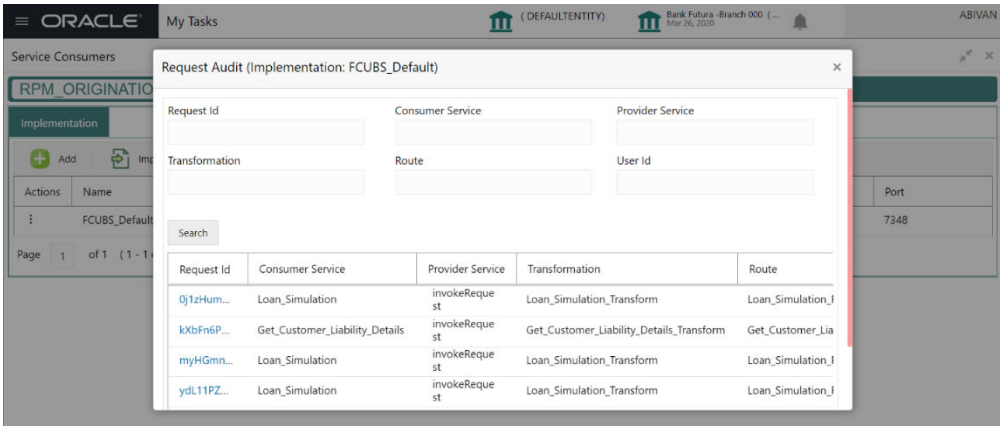
   d. From the **Actions**, click on **Request Audit**.

Figure 3-12 Service Consumer



   e. Check the latest Transformation for which you have performed the operation.

Figure 3-13 Request Audit



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

2. If there is no Oracle Banking Routing Hub call but, still Host call is failing (especially for Account Creation), then failure might be in the workflow task level. In order to debug this scenario, follow the below steps:

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

   API Url: http://whf00dtm.in.example.com:7011/plato-orch-service/api/v1/extn/custom-actions/queries/tasks?offset=0&limit=100

Figure 3-14 Body

```
{
    "q": "applicationNumber eq 000APP000006967",
        "queryType": [
        "ACQUIRED",
        "AVAILABLE",
        "HOLD",
        "COMPLETED"
    ]
}
```

Headers:

Content-Type:application/json

userId:

appId:platoorch

branchCode:

entityId:DEFAULTENTITY

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

Figure 3-15 subWorkflowId

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

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

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

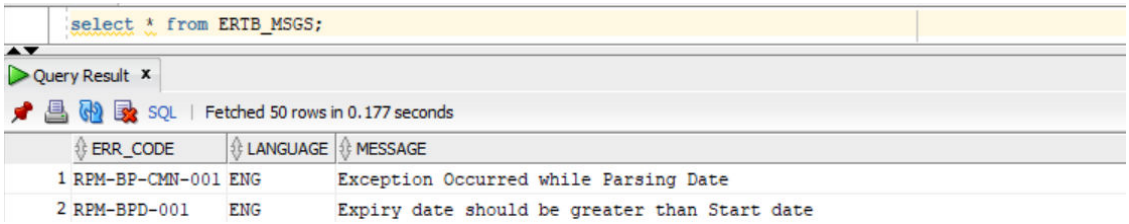
4
Business Error Codes

This topic provides information about business error codes.

The list of overrides/information/error codes that might be faced during usage of the application can be found in the table `ERTB_MSGS` of the corresponding service schema being operated on.

For example, if you face an error in Business Product maintenance screen and you want to see the error code in the table, you should connect to your Business Product schema and search for that particular error code in the ERTB_MSGS table.

Figure 4-1 Error Codes and Messages



Index