Developer Guide

Payment API integration with ESB

[Flow Diagram](#_3e503iyi8fgh)

[Property Configurations](#_o1qzkv2ndj8e)

[Configuring ESBSwitch](#_udjlxgb2m195)

[Configuring ESB Endpoint in Hub](#_jh054lusf41b)

[Artifacts](#_z622m9sd8luh)

[CAPP](#_h10vzubysn1d)

[LimitCheck Mediator](#_j62reexl1gk5)

[Configuring the ESB Pack](#_ugqcy1u1b68q)

[Cleaning UserInfo Database](#_82q13zmgg725)

[Endpoint Configurations](#_ckjmbmqk44f4)

[Production Tips](#_r7oqcz5q2yaj)

[Cleaning the Database](#_dm37exu8k473)

[Renaming Datasource](#_xownmi4ck9pl)

# Flow Diagram

TelcoPaymentESB (1).png

# Property Configurations

## Configuring *isUserInfoEnabled*

All payment charge requests will be forwarded to ESB. But we can configure whether to check for userInfo(check whether prepaid or postpaid) by setting the <IsUserInfoEnabled>true</IsUserInfoEnabled> property for each operator in *mifeEventSpendLimit.xml* configuration file(in Hub).

## Configuring ESB Endpoint in Hub

In order to change the ESB endpoint for the Payment API, change

esbEndpoint=<endpoint url here>

In repository/conf/**axiataMediator\_conf.properties** file in the **Hub**. Hub will be sending the request to ESB in following format.

*ESB\_url?oprendpoint=xxxx&operator=yyyyy&consumerKey=zzzzz&isUserInfoEnabled=<true|false>*

# Artifacts

All ESB artifacts are in the following repository under PaymentChargeESB directory.

<https://github.com/fazlan-nazeem/customerPOC/>

## 

## 

## CAPP

Following two maven projects contain the Capp artifacts.

* [PaymentCharge](https://github.com/fazlan-nazeem/customerPOC/tree/master/PaymentChargeESB/PaymentCharge)
* [PaymentChargeCapp](https://github.com/fazlan-nazeem/customerPOC/tree/master/PaymentChargeESB/PaymentChargeCapp)

All synapse artifacts are in PaymentCharge project.

Do the changes to the PaymentCharge and build the Capp using PaymentChargeCapp project. (use dev studio).

Copy the capp to deployment/server/carbonapps directory or upload it via management console to the ESB.

## LimitCheck Mediator

Following maven project contains LimitCheck mediator files.

[CheckLimit](https://github.com/fazlan-nazeem/customerPOC/tree/master/PaymentChargeESB/CheckLimit)

Do the necessary changes and build the mediator. Once done copy the jar file to repository/components/lib directory in ESB.

# Configuring the ESB Pack

* Download an ESB 5.0 pack.
* Copy the class mediator(checklimit-1.0.jar) to repository/components/lib directory.
* Copy Dbutils-1.0-SNAPSHOT.jar to repository/components/lib directory.
* Add the Capp(PaymentChargeCapp\_1.0.0.car) to respository/deployment/server carbonapps directory.
* configure mifeEventSpendLimit.xml with daily and monthly limits for each operator and place it in repository/conf
* Add following datasources to repository/conf/datasources/master-datasources.xml

1. AXIATA\_MIFE\_DB
2. telco\_ds

* Populate the *AXIATA\_MIFE\_DB* with the axiata db tables
* Execute the following two db scripts into the same database which is defined in telco\_ds

Userinfo\_tokenfo.sql

This script will create two tables namely **userinfo** and **tokeninfo**

# Cleaning UserInfo Database

A specific time interval should be decided on how often we clean up the userinfo local database. A cron job could be setup to do this task.

A shell script and an sql script has been created for this task.

* cleanup.sql
* Database\_clean.sh

# Endpoint Configurations

Following is the *CelcomUserInfoEndpoin*t userinfo endpoint configuration.

<endpoint xmlns="http://ws.apache.org/ns/synapse" name="CelcomUserInfoEndpoint">

<http uri-template="http://localhost:8286/oneapi/queryprofile/v1/{uri.var.operatorEndpoint}/subscriberType" method="get">

<suspendOnFailure>

<errorCodes>101500,101501,101506,101507,101508</errorCodes>

<initialDuration>1000</initialDuration>

<progressionFactor>2.0</progressionFactor>

<maximumDuration>60000</maximumDuration>

</suspendOnFailure>

<markForSuspension>

<errorCodes>101504,101505</errorCodes>

<retriesBeforeSuspension>3</retriesBeforeSuspension>

<retryDelay>1</retryDelay>

</markForSuspension>

<timeout>

<duration>1000</duration>

<responseAction>fault</responseAction>

</timeout>

</http>

</endpoint>

According to the above timeout configuration, when userinfo API fails to respond before 1 second, the call will be timed out. This is configured in following XML tag.

<duration>1000</duration>

Once timed out this endpoint will receive 3 more calls before it is suspended. If one of those 3 calls succeed then endpoint will be marked active. If all those 3 calls fail, then the endpoint will be suspended until <initialDuration> is passed.

For more info on handling endpoint configurations check [this](https://docs.wso2.com/display/ESB500/Endpoint+Error+Handling#EndpointErrorHandling-Timeout) doc.

In case if this endpoint is suspended, a different thread will be spawned to query the same API with a larger timeout.

There are 3 endpoints configured by default.

* CelcomUserInfoEndpoint
* CelcomDBUserInfoEndpoint
* CelcomTokenEndpoint

# Production Tips

1. Change Endpoint URLs of Userinfo API and Timeouts with suitable values in following two files.

* CelcomDBUserInfoEndpoint.xml
* CelcomUserInfoEndpoint.xml

1. Make sure the following two data sources are configured in ESB.

* jdbc/AXIATA\_MIFE\_DB
* jdbc/telco\_ds

Also make sure the db **AutoReconnect is switched off for telco\_ds**, because it will degrade performance when the db is down.

1. Make sure following two configuration file is present in ESB and Hub with identical configurations .

* mifeEventSpendLimit.xml

1. We need to edit the Capp to change the endpoint timeouts etc.
2. Remove <header name="sandbox" scope="transport" value="admin"/> from chargeRequestSeq.xml before production. This was added for sandbox testing purposes.

## Cleaning the Database

In production the database which the “telco\_ds” datasource is pointing to should be cleaned up regularly. This should be done for two reasons.

1. To avoid the database growing indefinitely
2. To propagate the change of a user from prepaid to postpaid or vice versa

Dbscripts have been added in [dbscripts](https://github.com/fazlan-nazeem/customerPOC/tree/master/PaymentChargeESB/dbscripts). Edit the number of days interval in cleanup.sql to delete records older than that amount of days.

## Renaming Datasource

In order to rename “telco\_ds” datasource to another name, the datasource configurations in the Capp has to be changed in the following files.

* dbLookupSeq.xml
* dbAsyncInsertSeq.xml