Small World Remittance Open Payments Interface Specification Low Level Design Document

**EIDIKOSYSTEMS INTEGRATORS**



Version: 1.0

Release Date: 16-09-2020

|  |  |  |
| --- | --- | --- |
| Prepared By | Reviewed By | Approved By |
| EIDIKO |  |  |

Document Classification: CONFIDENTIAL

Contents

[1 Introduction 5](#_Toc50368892)

[1.1 General Description 5](#_Toc50368893)

[2 Purpose 5](#_Toc50368894)

[3 Scope 5](#_Toc50368895)

[3.1 In Scope 5](#_Toc50368896)

[4 Channels Involved 5](#_Toc50368897)

[5 Interface Dependencies 6](#_Toc50368898)

[5.1 External Dependencies 6](#_Toc50368899)

[5.2 Internal Dependencies 6](#_Toc50368900)

[6 Business Process Summary 6](#_Toc50368901)

[6.1 Process Overview 6](#_Toc50368902)

[Figure: 1 Process flow Approach for Indigo Account Balance. 7](#_Toc50368903)

[6.1.1 Account Balance Steps 7](#_Toc50368904)

[7 IIB Implementation Process Flow 7](#_Toc50368905)

[7.1 Account Balance Successful Request Response Flow 7](#_Toc50368906)

[7.1.1 Process Diagram 7](#_Toc50368907)

[7.1.2 Process Flow Steps 8](#_Toc50368908)

[7.2 Retry Process workflow 9](#_Toc50368909)

[7.2.1 Process Flow Diagram 9](#_Toc50368910)

[7.2.2 Process Flow steps 9](#_Toc50368911)

[7.3 RAW\_AUDIT Process flow 10](#_Toc50368912)

[7.3.1 Process Flow Diagram 10](#_Toc50368913)

[7.3.2 Process Flow Steps: 10](#_Toc50368914)

[7.4 SOAP Incomming Router Process Flow 11](#_Toc50368915)

[7.4.1 Incoming Router Process Flow Diagram: 11](#_Toc50368916)

[7.4.2 Process Flow Steps: 11](#_Toc50368917)

[7.4.3 STWB\_ESB\_TRAN\_DETAILS Details 12](#_Toc50368918)

[7.4.4 SB\_ESB\_PCODE Details 13](#_Toc50368919)

[7.4.5 SOAP Outgoing Router Process Flow Diagram 13](#_Toc50368920)

[7.4.6 Process Flow Steps: 14](#_Toc50368921)

[7.5 Error Audit Process Details: 14](#_Toc50368922)

[7.5.1 Process Flow Diagram: 14](#_Toc50368923)

[7.5.2 Process Flow Steps: 15](#_Toc50368924)

[8 Interface Definitions 16](#_Toc50368925)

[8.1 Request Message Definition 16](#_Toc50368926)

[8.1.1 Channel Request Message Structure /Schema 16](#_Toc50368927)

[8.1.2 Channel Request Message Details 16](#_Toc50368928)

[8.1.4 T24 Request Message Structure /Schema 17](#_Toc50368929)

[8.1.5 T24 Request Message Details 17](#_Toc50368930)

[8.1.3 Data Base Response Codes 18](#_Toc50368931)

[8.2 Response Message Definition 19](#_Toc50368932)

[8.2.1 T24 & Channel Response Message Structure/Schema 19](#_Toc50368933)

[8.2.2 T24 & Channel Response Message Details 19](#_Toc50368934)

[9 Queue Details 21](#_Toc50368935)

[9.1 Table for Queue Details 21](#_Toc50368936)

[10 Logging Mechanism 22](#_Toc50368937)

[10.3 File Based Logging 24](#_Toc50368938)

[10.3.1 Log4j 24](#_Toc50368939)

**Approval**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Role | Approver | Company | E-mail | Sign-Off Date |
|  |  | Eidiko |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Version | Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## General Description

This Interface Design document outlines the integration requirements for the Small World Open Payments ***(Request and Response)***interface used as banking operations. It summarizes the business processes, which use this interface. It also covers error handling and exception scenarios.

# Purpose

The purpose of this document is to capture events that trigger the interface, main steps within the interface and the integration architecture. This document is intended for use by the developers of the applications identified, the integration development team, and by the test organizations responsible for the testing of these applications.

# Scope

## In Scope

This document focuseson outlining the interface design for the Small World Open Payments ***(Request and Response)*** interface. Central to this document are the following:

* Overview of the business process that drives the need for the interface
* Proposed integration approach
* Trigger events and business dependencies on this interface
* Sequence /Flow Chart diagram of the interface
* Error handling and exception scenarios
* Validation and backup requirements

# Channels Involved

The following table lists Indigo Account Balance channels

|  |  |
| --- | --- |
| **Item #** | **Channel Name** |
| 1 | Remittance |

# Interface Dependencies

## External Dependencies

The following table lists interface specific External requirements.

|  |  |
| --- | --- |
| **Item #** | **External Requirements** |
| 1 | ORADSN Database |
| 2 | Core Banking System(Temenos-T24) |
| 3 | CSO System |

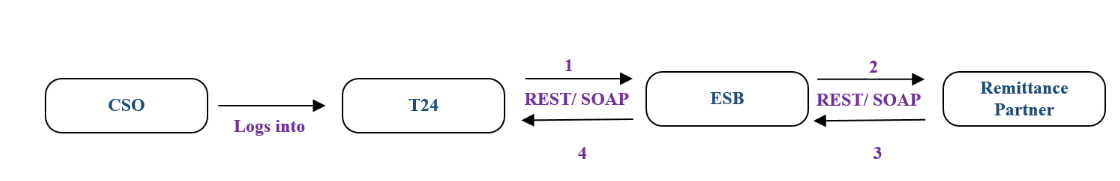
## Internal Dependencies

The following table lists interface specific internal requirements.

|  |  |
| --- | --- |
| **Item #** | **Internal Requirements** |
| 1 | SOAP Router Interface (Gateway) – StewardBankSoapRouter |
| 3 | ISO8583\_Lib (Shared Library) |
| 4 | StewardBankCommonEsql (Shared Library) |
| 5 | DB Logging App (Audit Logging) |

# Business Process Summary

## Process Overview



### Figure: 1 Process flow Small World Remittance Open Payments.

### Small World Remittance Open Payments Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1 | ESB receives the request from channel (CSO). |
| 2 | Request channel is the SOAP request accepts the request from the channel. |
| 3 | ESB frames the Request According to the Remittance Partner (SmallWorld) requires and send to the Remittance Partner (SmallWorld). |
| 4 | Remittance Partner (SmallWorld) will Respond back to the ESB with Response in SOAP. |
| 5 | The same Remittance Partner (SmallWorld) Response, ESB will send back to the channel (CSO). |
| 6 | On Successful call of Remittance Partner (SmallWorld) Response will be placed in the (RAW\_LOG\_SB\_REQ)Queue to log the response in Database (RAW\_AUDIT\_LOGGING) |
| 7 | On Unsuccessful call of T24 Response will be placed in the (ERROR\_LOG\_SB\_REQ) Queue to log the response in Database (ERR\_AUDIT\_LOGGING). |

# IIB Implementation Process Flow

## Small World Remittance Open Payments Successful Request Response Flow

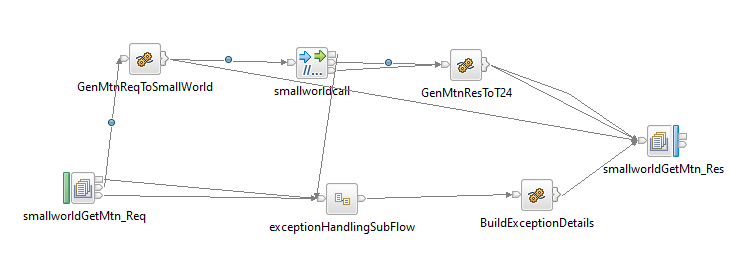
The **Small World Remittance Open Payments** process primarily involves 3 services

1. getMtnByFolio

2. getTransactionExportByMtn

3. payoutPickupAnywhereTransaction

### getMtnByFolio Process Diagram

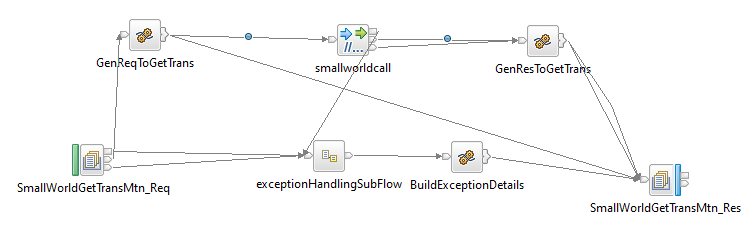


**Figure: 2 SmallWorld getMtnByFolio** **Flow**

### getMtnByFolio Process Flow Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1. | ESB receives the request from the channel using the SOAP Router Application based on Service Name trigs the Business Application (StewardBankRemittanceOpenpayments) of flow (SmallWorldGetMtnbyFolio) using Business Queue (SWGM\_SB\_SOAP\_REQ) i.e Request from channel. |
| 2 | Initial request and response are logged in the RAW\_LOG\_SB\_REQ and is insert in Database (RAW\_AUDIT\_LOGGING). |
| 3 | After logging of request ESB frames the Request to Remittance Partner (SmallWorld) server in the Business flow (StewardBankGetMtnbyFolio) i.e T24 server. |
| 4 | After Framing the Remittance Partner (SmallWorld) Request is logged in RAW\_LOG\_SB\_REQ and is insert in Database. |
| 5 | If the Remittance Partner (SmallWorld) call is successful, then SOAP response is generated. |
| 6 | Remittance Partner (SmallWorld) Response in SOAP Format is logged in RAW\_LOG\_SB\_REQ and insert in Database (RAW\_AUDIT\_LOGGING). |
| 7 | After the Response is generated by Remittance Partner (SmallWorld) call same response is sent back to Channel (CSO) with the help of SOAP Router Application (HTTP OutgoingRouterFlow). |
| 8 | Remittance Partner(SmallWorld)Response is logged in RAW\_LOG\_SB\_REQ and is insert in Database(RAW\_AUDIT\_LOGGING). |
| 9 | On Unsuccessful call T24 Response will be placed in the ERROR\_LOG\_SB\_REQ to log the request in Data Base(ERR\_AUDIT\_LOGGING). |

### getTransactionExportbyMtn Process Diagram

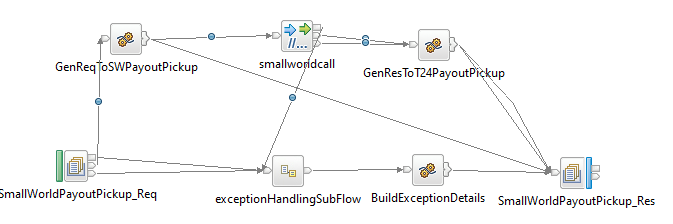


**Figure: 3 SmallWorld getTransactionExportbyMtn** **Flow**

### getTransactionExportbyMtn Process Flow Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1. | ESB receives the request from the channel using the SOAP Router Application based on Service Name trigs the Business Application (StewardBankRemittanceOpenpayments) of flow(SmallWorldGetTransactionExportbyMtn) using Business Queue (SWGTM\_SB\_SOAP\_REQ) i.e Request from channel. |
| 2 | Initial request and response are logged in the RAW\_LOG\_SB\_REQ and is insert in Database (RAW\_AUDIT\_LOGGING). |
| 3 | After logging of request ESB frames the Request to Remittance Partner (SmallWorld)server in the Business flow(SmallWorldGetTransactionExportbyMtn) i.e T24 server. |
| 4 | After Framing the Remittance Partner (SmallWorld)Request is logged in RAW\_LOG\_SB\_REQ and is insert in Database. |
| 5 | If the Remittance Partner (SmallWorld) call is successful, then SOAP response is generated. |
| 6 | Remittance Partner (SmallWorld) Response in SOAP Format is logged in RAW\_LOG\_SB\_REQ and insert in Database (RAW\_AUDIT\_LOGGING). |
| 7 | After the Response is generated by Remittance Partner (SmallWorld) call same response is sent back to Channel (CSO) with the help of SOAP Router Application (HTTP OutgoingRouterFlow). |
| 8 | Small Remittance Partner (SmallWorld) World Response is logged in RAW\_LOG\_SB\_REQ and is insert in Database(RAW\_AUDIT\_LOGGING). |
| 9 | On Unsuccessful call T24 Response will be placed in the ERROR\_LOG\_SB\_REQ to log the request in Data Base(ERR\_AUDIT\_LOGGING). |

### PayoutPickupAnywhereTransaction Process Diagram



**Figure: 4 SmallWorld PayoutPickupAnywhereTransaction** **Flow**

### PayoutPickupAnywhereTransaction Process Flow Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1. | ESB receives the request from the channel using the SOAP Router Application based on Service Name trigs the Business Application (StewardBankRemittanceOpenpayments) of flow(SmallWorldPayoutPickupAnywhereTransaction) using Business Queue (SWPP\_SB\_SOAP\_REQ) i.e Request from channel. |
| 2 | Initial request and response are logged in the RAW\_LOG\_SB\_REQ and is insert in Database (RAW\_AUDIT\_LOGGING). |
| 3 | After logging of request ESB frames the Request to Remittance Partner(SmallWorld) server in the Business flow(SmallWorldPayoutPickupAnywhereTransaction) i.e T24 server. |
| 4 | After Framing the Remittance Partner (SmallWorld) Request is logged in RAW\_LOG\_SB\_REQ and is insert in Database. |
| 5 | If the Rem ittance Partner (SmallWorld)call is successful, then SOAP response is generated. |
| 6 | Remittance Partner (SmallWorld) Response in SOAP Format is logged in RAW\_LOG\_SB\_REQ and insert in Database (RAW\_AUDIT\_LOGGING). |
| 7 | After the Response is generated by Remittance Partner (SmallWorld) call same response is sent back to Channel (CSO) with the help of SOAP Router Application (HTTP OutgoingRouterFlow). |
| 8 | Remittance Partner (SmallWorld) Response is logged in RAW\_LOG\_SB\_REQ and is insert in Database(RAW\_AUDIT\_LOGGING). |
| 9 | On Unsuccessful call T24 Response will be placed in the ERROR\_LOG\_SB\_REQ to log the request in Data Base(ERR\_AUDIT\_LOGGING). |

## RAW\_AUDIT Process flow

### Process Flow Diagram



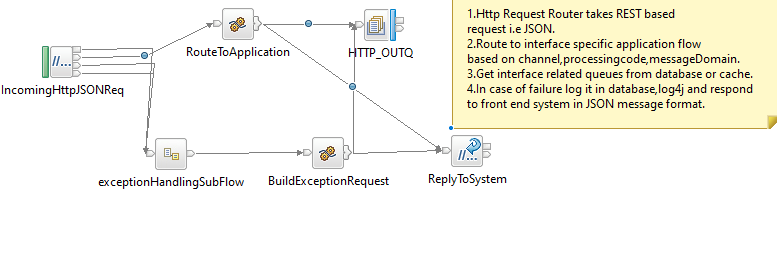
**Figure: 5 Raw\_Audit\_Logging Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The RAW\_AUDIT locks the input Request and Response |
|  | After the query is executed the record is inserted into the Database. |
|  | Final Response is inserted into the Database and then Response is logged into the RAW\_LOG\_SB\_REQ. |

## HTTP Incomming Router Process Flow

### Incoming Router Process Flow Diagram:

****

**Figure: 7 Incoming Router Flow**

### Process Flow Steps:

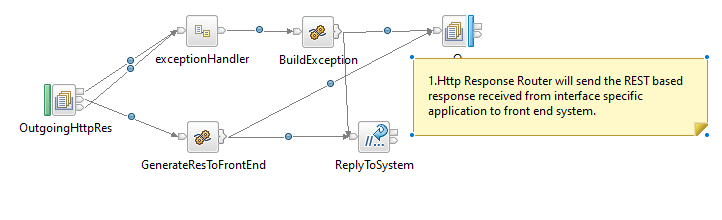
|  |  |
| --- | --- |
| **S. No** | IIB receives the request from channel. |
|  | ESB receives the request from channel. |
|  | Request channel is the SOAP request accepts the request from the channel |
|  | The request message is placed in the RAW\_LOG\_SB\_REQ to log the request in Database(RAW\_AUDIT\_LOGGING) |
|  | According to the Channel Request message fields such as Service Name it picks the queue name of the business flow from STWB\_ESB\_TRAN\_DETAILS table and propagates it to the business flow. |
|  | It validates the Service Name and domain of input from STWB\_ESB\_TRAN\_DETAILS if it fails to validate the error response is logged in the logs and in Data Base (ERROR\_AUDIT\_LOG) |
|  | After the validation Request message is passed through ESB. If the queue name presents in MQ manager the ESB business flow triggers otherwise the error response is sent to the front end as Unable to open queue. |
|  | If it is success or failure response will store them in RAW\_AUDIT\_LOG table or ERROR\_AUDIT\_LOG and logging into log file. |
|  | The request from channel is sent to ESB Respective Flow. |

### STWB\_ESB\_TRAN\_DETAILS Details

**Table Name: STWB\_ESB\_TRAN\_DETAILS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | PROC\_CODE | Processing Code | Varchar | 20 | Y | From input request |
| 2 | REQ\_QUEUE | TCPIP/SOAP Request Queue | Varchar | 20 | N | This Request Queue Name field is used for ISO Related Application |
| 3 | RES\_QUEUE | TCPIP/SOAP Response Queue | Varchar | 20 | N | This Response Queue Name field is used for ISO Related Application |
| 4 | HTTP\_REQ | HTTP Request Queue | Varchar | 20 | Y | This Request Queue Name field is used for REST Related Application |
| 5 | HTTP\_RES | HTTP Response Queue | Varchar | 20 | Y | This Response Queue Name field is used for REST Related Application |
| 6 | CHANNEL | Channel | Varchar | 20 | Y | Channel Name |
| 7 | MSGDOMAIN | Message Domain | Varchar | 20 | Y | Request Message Format |

### HTTP Outgoing Router Process Flow Diagram



**Figure: 8 Outgoing Router Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The OutgoingRouterFlow brings the Channel Response from ESB application from common response (ALL\_SB\_HTTP\_RES)queue. |
|  | Channel Response will be logged in RAW\_LOG\_SB\_REQ and insert in Data Base. |
|  | Finally Channel Response will be send to the Channel(CSO). |

## Error Audit Process Details:

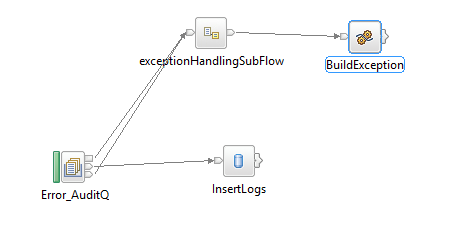
### Process Flow Diagram:



**Figure: 9 Exception Subflow**



**Figure: 10 Error\_Audit\_Logging Flow**



### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The ERROR\_AUDIT\_Q locks the input Request. |
|  | After the query is executed the record is inserted into the Data Base. |
|  | Final Response is inserted into the Database and then Response is logged into the ERR\_LOG\_SB\_REQ. |

# Interface Definitions

## Request Message Definition

### T24 Request Message Structure /Schema for GetMtnbyFolio Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:**[http://IPAddress:Port/v1/stewardBank/](http://IPAddress:Port/stewardbank/services/)

### T24 Request Message Details for GetMtnbyFolio Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getMtnByFolio>  <!--Optional:-->  <arg0>1010210836724</arg0>  <!--Optional:-->  <arg1>api101191</arg1>  <!--Optional:-->  <arg2>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg2>  </web:getMtnByFolio>  </soapenv:Body>  </soapenv:Envelope> |

### 8.1.3 Small World(Remittance Partner) Request Message Structure/Schema for GetMtnbyFolio Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:**<https://ua.oswfs.com/ImportExport/SimpleTransactionExportService>

### 8.1.4 Small World(Remittance Partner) Request Message Details for GetMtnbyFolio Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getTransactionExportByMtn>  <arg0>210836724</arg0>  <!--Optional:-->  <arg1></arg1>  <!--Optional:-->  <arg2></arg2>  <!--Optional:-->  <arg3></arg3>  <!--Optional:-->  <arg4>api101191</arg4>  <!--Optional:-->  <arg5>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg5>  </web:getTransactionExportByMtn>  </soapenv:Body>  </soapenv:Envelope> |

### 8.1.5 T24 Request Message Structure /Schema for GetTransactionExportByMtn Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:**[http://IPAddress:Port/v1/stewardBank/](http://IPAddress:Port/stewardbank/services/)

### 8.1.6 T24 Request Message Details for GetTransactionExportByMtn Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getTransactionExportByMtn>  <arg0>210836724</arg0>  <!--Optional:-->  <arg1></arg1>  <!--Optional:-->  <arg2></arg2>  <!--Optional:-->  <arg3></arg3>  <!--Optional:-->  <arg4>api101191</arg4>  <!--Optional:-->  <arg5>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg5>  </web:getTransactionExportByMtn>  </soapenv:Body>  </soapenv:Envelope> |

### 8.1.7 Small World(Remittance Partner) Request Message Structure/Schema for GetTransactionExportByMtn Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:** <https://ua.oswfs.com/ImportExport/SimpleTransactionExportService>

### 8.1.8 Small World(Remittance Partner) Request Message Details for GetTransactionExportByMtn Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getTransactionExportByMtn>  <arg0>210836724</arg0>  <!--Optional:-->  <arg1></arg1>  <!--Optional:-->  <arg2></arg2>  <!--Optional:-->  <arg3></arg3>  <!--Optional:-->  <arg4>api101191</arg4>  <!--Optional:-->  <arg5>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg5>  </web:getTransactionExportByMtn>  </soapenv:Body>  </soapenv:Envelope> |

### 8.1.9 T24 Request Message Structure /Schema for payoutPickupAnywhereTransaction Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:**[http://IPAddress:Port/v1/stewardBank/](http://IPAddress:Port/stewardbank/services/)

### 8.1.10 T24 Request Message Details for payoutPickupAnywhereTransaction Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getTransactionExportByMtn>  <arg0>210836724</arg0>  <!--Optional:-->  <arg1></arg1>  <!--Optional:-->  <arg2></arg2>  <!--Optional:-->  <arg3></arg3>  <!--Optional:-->  <arg4>api101191</arg4>  <!--Optional:-->  <arg5>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg5>  </web:getTransactionExportByMtn>  </soapenv:Body>  </soapenv:Envelope> |

### 8.1.11 Small World(Remittance Partner) Request Message Structure/Schema for payoutPickupAnywhereTransaction Service

**Request Type:** SOAP

**Request Format:** XMLNSC

**Request URL:** <https://ua.oswfs.com/ImportExport/SimpleTransactionExportService>

### 8.1.12 Small World (Remittance Partner) Request Message Details for payoutPickupAnywhereTransaction Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservice.ie.smallworldfs.com/">  <soapenv:Header/>  <soapenv:Body>  <web:getTransactionExportByMtn>  <arg0>210836724</arg0>  <!--Optional:-->  <arg1></arg1>  <!--Optional:-->  <arg2></arg2>  <!--Optional:-->  <arg3></arg3>  <!--Optional:-->  <arg4>api101191</arg4>  <!--Optional:-->  <arg5>d51bdc96dec6b778f3b07172d253faa25b0a75ae</arg5>  </web:getTransactionExportByMtn>  </soapenv:Body>  </soapenv:Envelope> |

### Data Base Response Codes

001 = "Fatal Exception";

002 = "Recoverable Exception";

003 = "Configuration Exception";

004 = "Security Exception";

005 = "Parser Exception";

006 = "Conversion Exception";

007 = "Data Base Exception";

008 = "User Exception";

009 = "Cast Exception";

010 = "Message Exception";

011 = "SQL Exception";

012 = "Socket Exception";

013 = "Socket Timeout Exception";

014 = "Unknown Exception";

015 = "Failure";

## Response Message Definition

### T24 & Small World(Remittance Partner) Response Message Structure/Schema for GetMtnbyFolio Service

**Response Type:** SOAP

### T24 & Small World(Remittance Partner) Response Message Details for GetMtnbyFolio Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  <soapenv:Body>  <ns2:getMtnByFolioResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">  <return>210836724</return>  </ns2:getMtnByFolioResponse>  </soapenv:Body>  </soapenv:Envelope> |

### T24 & Small World(Remittance Partner) Response Message Structure/Schema for GetTransactionExportByMtn Service

**Response Type:** SOAP

### T24 & Small World(Remittance Partner) Response Message Details for GetTransactionExportByMtn Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  <soapenv:Body>  <ns2:getMtnByFolioResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">  <return>210836724</return>  </ns2:getMtnByFolioResponse>  </soapenv:Body>  </soapenv:Envelope> |

### T24 & Small World(Remittance Partner) Response Message Structure/Schema for payoutPickupAnywhereTransaction Service

**Response Type:** SOAP

### T24 & Small World(Remittance Partner) Response Message Details for payoutPickupAnywhereTransaction Service

|  |
| --- |
| <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  <soapenv:Body>  <ns2:getMtnByFolioResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">  <return>210836724</return>  </ns2:getMtnByFolioResponse>  </soapenv:Body>  </soapenv:Envelope> |

# Queue Details

## Table for Queue Details

**Queue Objects**

**Queue Objects**

|  |  |  |  |
| --- | --- | --- | --- |
| **Queue Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| SWGM\_SB\_SOAP\_REQ | Receives the Messages from Channel via Router to ESB SmallWorldGetMtnByFolio Flow. | <Final Script Will be placed, after complete Unit Testing> |
| SWGTM\_SB\_SOAP\_REQ | Receives the Messages from Channel via Router to ESB SmallWorldGetTransactionExportByMtn Flow. |
| SWPP\_SB\_SOAP\_REQ | Receives the Messages from Channel via Router to ESB SmallWorldPayoutPickupAnywhereTransaction Flow. |
| ALL\_SB\_HTTP\_RES | This is the Queue used to send to response back to the channel via router. |
| RAW\_LOG\_SB\_REQ | This queue is used to DB Log the successful transactions into the database. |
|  | ERR\_LOG\_SB\_REQ | This queue is used to DB Log the error/unsuccessful transactions into the database. |  |
|  | SB\_EXCQ | The common flow ‘StewardBankExceptionHandling’ uses this queue to store the exception details. |  |

# Logging Mechanism

**10.1 Insert into Raw\_Audit\_Table**

**Table Name: RAW\_AUDIT\_TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | MSGID | MessageID | Varchar | 100 | Y | From input request |
| 2 | LOGGING\_TIME | Logging Time | TimeStamp | 6 | Y | Name |
| 3 | MESSAGE | Message(Request/Response) | Clob | - | Y | Request from channel |
| 4 | MESSAGETYPE | Type of Message(Request/Response) | Varchar | 100 | Y | Type of message either Request or Response |
| 5 | APPNAME | Application Name | Varchar | 100 | Y | Name of the Application |
| 6 | BROKER | Broker Name | Varchar | 100 | Y | Broker Name |
| 7 | TIME\_LOCAL\_TRANSACTION | Transaction Time | Varchar | 20 | Y | Time of the Transaction |
| 8 | DATE\_LOCAL\_TRANSACTION | Transaction Date | Varchar | 20 | Y | Date of the Transaction |
| 9 | RETRIEVAL\_REFERENCE\_N | Retrieval Reference Number | Varchar | 20 | Y |  |

**10.2 Insert into Error\_Audit\_Table**

**Table Name: ERROR\_AUDIT\_TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | MSGID | MessageID | Varchar | 50 | Y | From input request |
| 2 | LOGGING\_TIME | Application Name | Timestamp | 50 | Y | Time of the logging |
| 3 | MESSAGE | Message Type(Request/Response) | Clob | 50 | Y | Request of the Message |
| 4 | MESSAGETYPE | Message Type | Varchar | 50 | Y | Type of message either Request or Response |
| 5 | APPNAME | Application Name | Varchar | 4000 | Y | Name of the Application |
| 6 | BROKER | Broker Name | Varchar | - | Y | Name of the Broker |
| 7 | ERRORDESCRIPTION | Exception | Clob | 4000 | Y | Exception Information |
|  | TIME\_LOCAL\_TRANSACTION | Transaction Time | Varchar | 20 | Y | Time of the Transaction |
|  | DATE\_LOCAL\_TRANSACTION | Transaction Date | Varchar | 20 | Y | Date of the Transaction |
|  | RETRIEVAL\_REFERENCE\_NUMBER | Retrieval Reference Number | Varchar | 20 | Y | Retrieval Reference Number |
|  | BORKER\_ERROR\_CODE | Broker Error Code | Varchar |  | Y | Error code generate by Broker |
|  | USER\_DEFINE\_ERROR\_CODE | User Defined Error Code | Varchar |  | Y | User Defined Code |

## 10.3 File Based Logging

### 10.3.1 Log4j

As part of auditing the request we have to use Log4j mechanism in the application to log the request in the file.

File Path: To be specified, while deploying the application