INDIGO Interface Specification Low Level Design Document

**EIDIKOSYSTEMS INTEGRATORS**



Version: 1.0

Release Date: 20-08-2020

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

Document Classification: CONFIDENTIAL

Contents

[1 Introduction 5](#_Toc48319411)

[1.1 General Description 5](#_Toc48319412)

[2 Purpose 5](#_Toc48319413)

[3 Scope 5](#_Toc48319414)

[3.1 In Scope 5](#_Toc48319415)

[4 Channels Involved 5](#_Toc48319416)

[5 Interface Dependencies 6](#_Toc48319417)

[5.1 External Dependencies 6](#_Toc48319418)

[5.2 Internal Dependencies 6](#_Toc48319419)

[6 Business Process Summary 6](#_Toc48319420)

[6.1 Process Overview 6](#_Toc48319421)

[Figure: 1 Process flow Approach for Indigo. 7](#_Toc48319422)

[6.1.1 Indigo Steps 7](#_Toc48319423)

[7 IIB Implementation Process Flow 8](#_Toc48319424)

[7.1 Indigo Successful Request Response Flow 8](#_Toc48319425)

[7.1.1 Process Diagram 8](#_Toc48319426)

[7.1.2 Process Flow Steps 8](#_Toc48319427)

[7.2 Retry Process workflow 9](#_Toc48319428)

[7.2.1 Process Flow Diagram 9](#_Toc48319429)

[7.2.2 Process Flow steps 10](#_Toc48319430)

[7.3 RAW\_AUDIT Process flow 10](#_Toc48319431)

[7.3.1 Process Flow Diagram 10](#_Toc48319432)

[7.3.2 Process Flow Steps: 11](#_Toc48319433)

[7.4 TCPIP Router Process Flow 11](#_Toc48319434)

[7.4.1 Incoming Router Process Flow Diagram: 11](#_Toc48319435)

[7.4.2 Process Flow Steps: 12](#_Toc48319436)

[7.4.3 Integration Solution Details 12](#_Toc48319437)

[7.4.4 SB\_ESB\_PCODE Details 13](#_Toc48319438)

[7.4.5 Outgoing Router Process Flow Diagram 14](#_Toc48319439)

[7.4.6 Process Flow Steps: 14](#_Toc48319440)

[7.5 Error Audit Process Details: 15](#_Toc48319441)

[7.5.1 Process Flow Diagram: 15](#_Toc48319442)

[7.5.2 Process Flow Steps: 16](#_Toc48319443)

[8 Interface Definitions 16](#_Toc48319444)

[8.1 Request Message Definition 16](#_Toc48319445)

[8.1.1 Channel Request Message Structure /Schema 16](#_Toc48319446)

[8.1.2 Channel Request Message Details 16](#_Toc48319447)

[8.1.4 T24 Request Message Structure /Schema 26](#_Toc48319448)

[8.1.5 T24 Request Message Details 26](#_Toc48319449)

[8.1.2 Data Base Response Codes 32](#_Toc48319450)

[8.2 Response Message Definition 32](#_Toc48319451)

[8.2.1 T24 & Channel Response Message Structure/Schema 32](#_Toc48319452)

[8.2.2 T24 & Channel Response Message Details 33](#_Toc48319453)

[8.3 Integration Definition 42](#_Toc48319454)

[8.3.1 Integration Details: 42](#_Toc48319455)

[9 Queue Details 43](#_Toc48319456)

[9.1 Table for Queue Details 43](#_Toc48319457)

[10 Logging Mechanism 44](#_Toc48319458)

[10.3 File Based Logging 46](#_Toc48319459)

[10.3.1 Log4j 46](#_Toc48319460)

**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 Indigo ***(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 Indigo ***(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 channels

|  |  |
| --- | --- |
| **Item #** | **Channel Name** |
| 1 | Mobile Application |
| 2 | \*210# |
| 3 | Square World |
| 4 | Square for Business |
| 5 | Agent POS |
| 6 | Self Service |
| 7 | ATM |

# 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

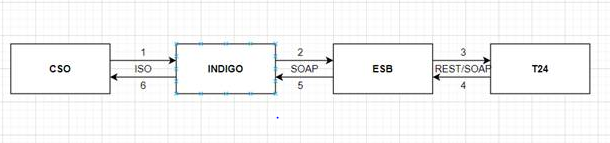
The following table lists interface specific internal requirements.

|  |  |
| --- | --- |
| **Item #** | **Internal Requirements** |
| 1 | TCPIP Router Interface (Gateway) – StewardBankTCPIPRouter |
| 2 | TCPIP Retry Interface (Subflow) - StewardBankT24Call |
| 3 | ISO8583\_Lib (Shared Library) |
| 4 | StewardBankCommonEsql (Shared Library) |
| 5 | DB Logging App (Audit Logging) |

# Business Process Summary

## Process Overview

The **Indigo** process primarily involves to



[If the Account is in T24-Request Type-0200]

### Figure: 1 Process flow Approach for Indigo.

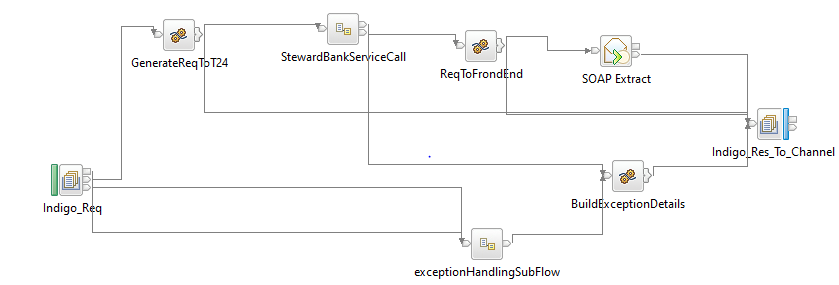
### Indigo Steps

|  |  |
| --- | --- |
| **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 T24 requires and send to the T24. |
| 4 | T24 will Respond back to the ESB with Response in SOAP. |
| 5 | The same T24 Response, ESB will send back to the channel (CSO). |
| 6 | On Successful call of T24 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

## Indigo Successful Request Response Flow

### Process Diagram



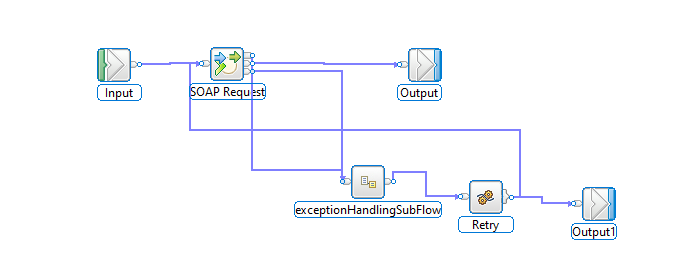
**Figure: 2 Indigo Flow**

### 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 (IndigoFlow) using Business Queue (INDIGO\_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 T24 server i.e T24 server. |
| 4 | After Framing the T24 Request is logged in RAW\_LOG\_SB\_REQ and is insert in Database. |
| 5 | If the T24 call is successful, then SOAP response is generated. |
| 6 | T24 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 T24 call same response is sent back to Channel (CSO) with the help of SOAP Router Application (SOAP OutgoingRouterFlow). |
| 8 | T24 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). |

## Retry Process workflow

### Process Flow Diagram



**Figure: 3 SOAP Retry for Indigo (T24 Call)**

### Process Flow steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1 | The StewardBankServiceCall Subflow will send it to T24 and then response from T24 contains SOAP message. |
| 2 | In case of failure and if the domain is SOAP then log it in database,log4j and send failure response to SOAP outgoing router in SOAP. |
| 3 | SOAP Retry call is used to call the backend service for multiple times. |
| 4 | On Successful call of backend service, response will be generated and send back to Business Application for further process. |
| 5 | On Unsuccessful call of backend service it Retry for 3 times and sends error message to Business Application for further Process. |

## RAW\_AUDIT Process flow

### Process Flow Diagram



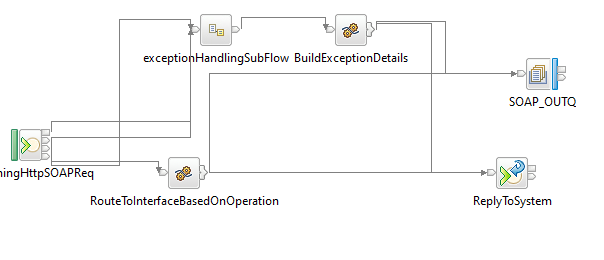
**Figure: 4 Raw\_Audit\_Logging Flow**

### 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. |

## SOAP Incomming Router Process Flow

### Incoming Router Process Flow Diagram:

****

**Figure: 5 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 INTEGRATION\_SOL\_DETAILS table and propagates it to the business flow. |
|  | It validates the Service Name and domain of input from INTEGRATION\_SOL\_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 Indigo Flow |

### Integration Solution Details

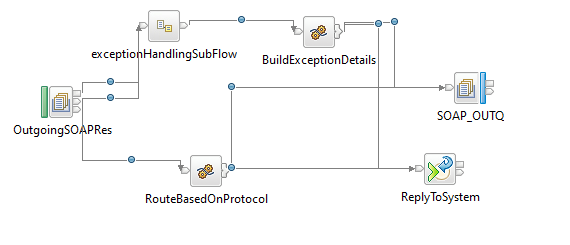
**Table Name: INTEGRATION\_SOL\_DETAILS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 Request Queue | Varchar | 20 | N | This Request Queue Name field is used for ISO Related Application |
| 3 | RES\_QUEUE | TCPIP 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 |

### SB\_ESB\_PCODE Details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | CHANNEL | Channel Name | Varchar | 20 | Y | From input request in 127.3 field. |
| 2 | POSTILION\_PCODE | Postilion Code Number | Varchar | 20 | N | The processing code from Request. |
| 3 | ESB\_PCODE | ESB processing Code | Varchar | 20 | N | The processing code which is sent to T24 |
| 4 | TRANS\_TYPE | Transaction type of channel. | Varchar | 20 | Y | The transaction type of channel.  (Indigo) |
| 5 | CHANNEL2 | 2nd Channel Name | Varchar | 20 | Y | From input request in 127.3 field. |
| 6 | CHANNEL3 | 3rd Channel Name | Varchar | 20 | Y | From input request in 127.3 field. |
| 7 | CHANNEL4 | 4th Channel Name | Varchar | 20 | Y | From input request in 127.3 field. |
| 8 | DESTINATION\_SINK | Destination Channel Name | Varchar | 20 | Y | From input request in 127.3 field. |

### SOAP Outgoing Router Process Flow Diagram



**Figure: 6 Outgoing Router Flow**

### Process Flow Steps:

|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The OutgoingRouterFlow brings the Channel Response from ESB application from common response (SOAP\_RES\_Q)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: 7 Exception Subflow**



**Figure: 8 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

### Channel Request Message Structure /Schema

**Request Type:** SOAP

### Channel Request Message Details

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:stew="http://temenos.com/stewardbank" xmlns:fun="http://temenos.com/FUNDSTRANSFERCARDCHARGE">

<soapenv:Header/>

<soapenv:Body>

<stew:CardCharges>

<WebRequestCommon>

<!--Optional:-->

<company>ZW0010001</company>

<password>T3m3nos@852</password>

<userName>TWSUSER01</userName>

</WebRequestCommon>

<OfsFunction>

<noOfAuth>0</noOfAuth>

</OfsFunction>

<FUNDSTRANSFERCARDCHARGEType id="">

<fun:DebitAccount>1000000047</fun:DebitAccount>

<fun:DebitCurrency>ZWL</fun:DebitCurrency>

<fun:DebitAmount>38.19</fun:DebitAmount>

<fun:CreditAccount>ZWL1409200010001</fun:CreditAccount>

</FUNDSTRANSFERCARDCHARGEType>

</stew:CardCharges>

</soapenv:Body>

</soapenv:Envelope>

### 8.1.4 T24 Request Message Structure /Schema

**Request Type:** SOAP

### T24 Request Message Details

**<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:stew="http://temenos.com/stewardbank" xmlns:fun="http://temenos.com/FUNDSTRANSFERCARDCHARGE">**

**<soapenv:Header/>**

**<soapenv:Body>**

**<stew:CardCharges>**

**<WebRequestCommon>**

**<!--Optional:-->**

**<company>ZW0010001</company>**

**<password>T3m3nos@852</password>**

**<userName>TWSUSER01</userName>**

**</WebRequestCommon>**

**<OfsFunction>**

**<noOfAuth>0</noOfAuth>**

**</OfsFunction>**

**<FUNDSTRANSFERCARDCHARGEType id="">**

**<fun:DebitAccount>1000000047</fun:DebitAccount>**

**<fun:DebitCurrency>ZWL</fun:DebitCurrency>**

**<fun:DebitAmount>38.19</fun:DebitAmount>**

**<fun:CreditAccount>ZWL1409200010001</fun:CreditAccount>**

**</FUNDSTRANSFERCARDCHARGEType>**

**</stew:CardCharges>**

**</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 & Channel Response Message Structure/Schema

**Response Type:** SOAP

### 8.2.2 T24 & Channel Response Message Details

<NS1:Envelope xmlns:NS1="http://schemas.xmlsoap.org/soap/envelope/">

<NS1:Body>

<ns9:CardChargesResponse xmlns:ns2="http://temenos.com/ACCOUNTCARDAMEND" xmlns:ns3="http://temenos.com/ACCOUNT" xmlns:ns4="http://temenos.com/GETACCTBAL" xmlns:ns5="http://temenos.com/FUNDSTRANSFERCARDCHARGEREV" xmlns:ns6="http://temenos.com/FUNDSTRANSFER" xmlns:ns7="http://temenos.com/FUNDSTRANSFERCARDCHARGE" xmlns:ns8="http://temenos.com/JSLCARDACCT" xmlns:ns9="http://temenos.com/stewardbank">

<Status>

<transactionId>FT20233D4CYT</transactionId>

<messageId/>

<successIndicator>Success</successIndicator>

<application>FUNDS.TRANSFER</application>

</Status>

<FUNDSTRANSFERType id="FT20233D4CYT">

<ns6:TRANSACTIONTYPE>AC</ns6:TRANSACTIONTYPE>

<ns6:DEBITACCTNO>1000000047</ns6:DEBITACCTNO>

<ns6:CURRENCYMKTDR>1</ns6:CURRENCYMKTDR>

<ns6:DEBITCURRENCY>ZWL</ns6:DEBITCURRENCY>

<ns6:DEBITAMOUNT>38.19</ns6:DEBITAMOUNT>

<ns6:DEBITVALUEDATE>20200820</ns6:DEBITVALUEDATE>

<ns6:CREDITACCTNO>ZWL1409200010001</ns6:CREDITACCTNO>

<ns6:CURRENCYMKTCR>1</ns6:CURRENCYMKTCR>

<ns6:CREDITCURRENCY>ZWL</ns6:CREDITCURRENCY>

<ns6:CREDITVALUEDATE>20200820</ns6:CREDITVALUEDATE>

<ns6:PROCESSINGDATE>20200820</ns6:PROCESSINGDATE>

<ns6:CHARGECOMDISPLAY>NO</ns6:CHARGECOMDISPLAY>

<ns6:COMMISSIONCODE>WAIVE</ns6:COMMISSIONCODE>

<ns6:CHARGECODE>WAIVE</ns6:CHARGECODE>

<ns6:PROFITCENTRECUST>100121</ns6:PROFITCENTRECUST>

<ns6:RETURNTODEPT>NO</ns6:RETURNTODEPT>

<ns6:FEDFUNDS>NO</ns6:FEDFUNDS>

<ns6:POSITIONTYPE>TR</ns6:POSITIONTYPE>

<ns6:AMOUNTDEBITED>ZWL38.19</ns6:AMOUNTDEBITED>

<ns6:AMOUNTCREDITED>ZWL38.19</ns6:AMOUNTCREDITED>

<ns6:CREDITCOMPCODE>ZW0010001</ns6:CREDITCOMPCODE>

<ns6:DEBITCOMPCODE>ZW0010001</ns6:DEBITCOMPCODE>

<ns6:LOCAMTDEBITED>38.19</ns6:LOCAMTDEBITED>

<ns6:LOCAMTCREDITED>38.19</ns6:LOCAMTCREDITED>

<ns6:CUSTGROUPLEVEL>99</ns6:CUSTGROUPLEVEL>

<ns6:DEBITCUSTOMER>100121</ns6:DEBITCUSTOMER>

<ns6:DRADVICEREQDYN>N</ns6:DRADVICEREQDYN>

<ns6:CRADVICEREQDYN>N</ns6:CRADVICEREQDYN>

<ns6:TOTRECCOMM>0</ns6:TOTRECCOMM>

<ns6:TOTRECCOMMLCL>0</ns6:TOTRECCOMMLCL>

<ns6:TOTRECCHG>0</ns6:TOTRECCHG>

<ns6:TOTRECCHGLCL>0</ns6:TOTRECCHGLCL>

<ns6:RATEFIXING>NO</ns6:RATEFIXING>

<ns6:TOTRECCHGCRCCY>0</ns6:TOTRECCHGCRCCY>

<ns6:TOTSNDCHGCRCCY>0.00</ns6:TOTSNDCHGCRCCY>

<ns6:AUTHDATE>20200820</ns6:AUTHDATE>

<ns6:ROUNDTYPE>NATURAL</ns6:ROUNDTYPE>

<ns6:gSTMTNOS>

<ns6:STMTNOS>192260627752949.00</ns6:STMTNOS>

<ns6:STMTNOS>1-2</ns6:STMTNOS>

</ns6:gSTMTNOS>

<ns6:CURRNO>1</ns6:CURRNO>

<ns6:gINPUTTER>

<ns6:INPUTTER>6277\_TWSUSER\_\_OFS\_TWS</ns6:INPUTTER>

</ns6:gINPUTTER>

<ns6:gDATETIME>

<ns6:DATETIME>2008201442</ns6:DATETIME>

</ns6:gDATETIME>

<ns6:AUTHORISER>6277\_TWSUSER\_OFS\_TWS</ns6:AUTHORISER>

<ns6:COCODE>ZW0010001</ns6:COCODE>

<ns6:DEPTCODE>1</ns6:DEPTCODE>

</FUNDSTRANSFERType>

</ns9:CardChargesResponse>

</NS1:Body>

</NS1:Envelope>

# Queue Details

## Table for Queue Details

|  |  |  |  |
| --- | --- | --- | --- |
| **QM Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| INDIGO\_SB\_SOAP\_REQ | Receives the Messages from Channel via Router to ESB IndigoFlow. | <Final Script Will be placed, after complete Unit Testing> |
| ALL\_SB\_TCP\_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. |  |
|  | EXC\_Q | This queue is used to store the exceptions occurred in the flows. |  |

# 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