NanoLoan Eligibility Check Interface Specification Low Level Design Document

**EIDIKOSYSTEMS INTEGRATORS**



Version: 1.0

Release Date: 31-08 -2020

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

Document Classification: CONFIDENTIAL

Contents

[1 Introduction 5](#_Toc50409452)

[1.1 General Description 5](#_Toc50409453)

[2 Purpose 5](#_Toc50409454)

[3 Scope 5](#_Toc50409455)

[3.1 In Scope 5](#_Toc50409456)

[4 Channels Involved 5](#_Toc50409457)

[5 Interface Dependencies 6](#_Toc50409458)

[5.1 External Dependencies 6](#_Toc50409459)

[5.2 Internal Dependencies 6](#_Toc50409460)

[6 Business Process Summary 7](#_Toc50409461)

[6.1 Process Overview 7](#_Toc50409462)

[Figure:1 Process flow Approach for NanoLoan Eligibility Check for \*236# Channel . 7](#_Toc50409463)

[Figure:2 Process flow Approach for Nano Loan Eligibility Check for Internet Banking Channel. 7](#_Toc50409464)

[6.2 Nano Loan Eligibility Check process Flow 8](#_Toc50409465)

[7 ESB Implementation Process Flow 8](#_Toc50409466)

[7.1 Funding Agent Lines Successful Request Response Flow 8](#_Toc50409467)

[7.1.1 Process Diagram 8](#_Toc50409468)

[7.1.2 Process Flow Steps 9](#_Toc50409469)

[7.2 Retry Process workflow 10](#_Toc50409470)

[7.2.1 Process Flow Diagram 10](#_Toc50409471)

[7.2.2 Process Flow steps 10](#_Toc50409472)

[8 Interface Definitions 15](#_Toc50409473)

[8.1 Request Message Definition 15](#_Toc50409474)

[8.1.1 Channel (LoanEngine/Internet Banking Backend) Request Message Structure /Schema 15](#_Toc50409475)

[8.1.2 LoanEngine/Internet Banking Backend Request Message Details 15](#_Toc50409476)

[8.1.3 LoanEngine/Internet Banking Backend Sample Source Messages 16](#_Toc50409477)

[8.1.4 DWH Request Message Structure /Schema 16](#_Toc50409478)

[8.1.5 DWH Request Message Details 16](#_Toc50409479)

[8.1.6 DWH Sample Source Messages 17](#_Toc50409480)

[8.1.7 Data Base Response Codes 17](#_Toc50409481)

[8.2 Response Message Definition 18](#_Toc50409482)

[8.2.1 LoanEngine/Internet Banking Backend Response Message Structure/Schema 18](#_Toc50409483)

[8.2.2 DWH Response Message Details 18](#_Toc50409484)

[8.2.3 DWH Sample Response Messages 18](#_Toc50409485)

[8.2.4 LoanEngine/Internet Banking Backend Response Message Details 19](#_Toc50409486)

[8.2.5 LoanEngine/Internet Banking Backend Sample Response Messages 19](#_Toc50409487)

[9 Queue Details 20](#_Toc50409488)

[9.1 Queue Details Table 20](#_Toc50409489)

[10 Logging Mechanism 20](#_Toc50409490)

[10.1 Insert into Raw\_Audit\_Table 20](#_Toc50409491)

[10.2 Insert into Error\_Audit\_Table 21](#_Toc50409492)

[10.3 File Based Logging 23](#_Toc50409493)

[10.3.1 Log4j 23](#_Toc50409494)

**Approval**

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

**Revision History**

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

# Introduction

## General Description

This Interface Design document outlines the integration requirements for the NanoLoan Eligibility Check ***(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 focuses on outlining the interface design for the NanoLoan Eligibility Check ***(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
* Data mapping between the source and the target
* Error handling and exception scenarios
* Validation and backup requirements

# Channels Involved

The following table lists Cash Out channels

|  |  |
| --- | --- |
| **Item #** | **Channel Name** |
| 1 | Internet Banking and \*236# |
| 2 | Mobile |

# Interface Dependencies

## External Dependencies

The following table lists interface specific External requirements.

|  |  |
| --- | --- |
| **Item #** | **External Requirements** |
| 1 | MS SQL Database - 2017 |
| 2 | Data Ware House |
| 3 | Loan Engine |

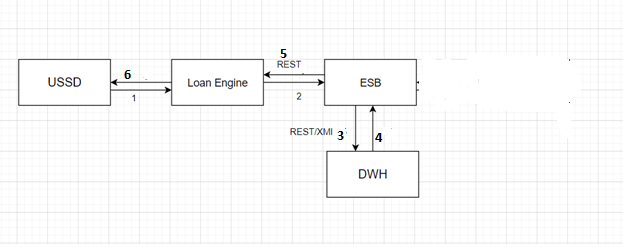
## Internal Dependencies

The following table lists interface specific internal requirements.

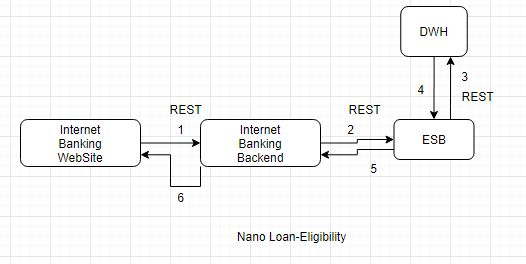
|  |  |
| --- | --- |
| **Item #** | **Internal Requirements** |
| 1 | HTTP Router Interface |
| 2 | HTTP Retry Interface (Subflow) |
| 3 | DB Logging App |
| 4 | StewardCommanEsql\_Lib |
| 5 | StewardExceptionHandler (Subflow) |

# Business Process Summary

## Process Overview



### Figure:1 Process flow Approach for NanoLoan Eligibility Check for \*236# Channel .



### Figure:2 Process flow Approach for Nano Loan Eligibility Check for Internet Banking Channel.

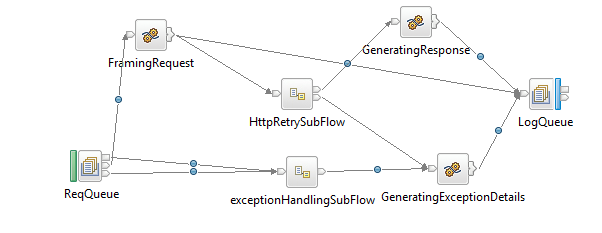
## Nano Loan Eligibility Check process Flow

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1 | ESB receives the request from channel (LoanEngine/Internet Banking Backend). |
| 2 | Through Http Router , the request comes from the channel |
| 3 | ESB frames the Request According to the DWH requires and send to the DWH. |
| 4 | DWH will Respond back to the ESB with Response |
| 5 | The same DWH Response, ESB will send back to the channel (LoanEngine/Internet Banking Backend) |
| 6 | On Successful calls of DWH and LoanEngine/Internet Banking Backend Request and Response will be placed in the RAW\_LOG\_SB\_REQ to log the request in Data Base (RAW\_AUDIT\_LOGGING) |
| 7 | On Unsuccessful calls of DWH and LoanEngine/Internet Banking Backend Request and Response will be placed in the ERROR\_LOG\_SB\_REQ to log the request in Data Base (ERR\_AUDIT\_LOGGING) |

# ESB Implementation Process Flow

## Funding Agent Lines Successful Request Response Flow

### Process Diagram



**Figure:3 Nano Loan Eligibility Check Flow**

### Process Flow Steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1. | ESB receives the request from the channel (LoanEngine/Internet Banking Backend) using the HTTP Router Application based on processingcode trigs the Business Application (NanoLoan Eligibility Check) using Business Queue (NANOELE\_SB\_HTTP\_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 Framing the DWH Request is logged in RAW\_LOG\_SB\_REQ and is insert in Data Base(RAW\_AUDIT\_LOGGING) |
| 5 | If the DWH call is successful, then JSON response will be generated. |
| 6 | DWH response in JSON format is logged in RAW\_LOG\_SB\_REQ and insert in Data Base (RAW\_AUDIT\_LOGGING) |
| 7 | The response from DWH is send back to the channel (LoanEngine/Internet Banking Backend) |
| 8 | Channel Response is logged in RAW\_LOG\_SB\_REQ and is insert in Data Base (RAW\_AUDIT\_LOGGING) |
| 9 | On Unsuccessful calls of DWH and Channel Request and 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 HTTP Retry for Cash Out**

### Process Flow steps

|  |  |
| --- | --- |
| **S. No** | **Activities** |
| 1 | HTTP Retry call is used to call the backend service for multiple times |
| 2 | On Successful call of backend service, response will be generated and send back to Business Application For further process |
| 3 | On Unsuccessful call of backend service it Retry for 3 times and sends error message to Business Application for further Process |

**7.3 Raw\_Audit Process Flow:**

**7.3.1 Process Flow Diagram:**



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

**7.3.2 Process Flow Steps:**

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

**7.4 HTTP Router Process Flow**

**7.4.1 Incoming Router Process Flow Diagram:**



**Figure:5 Incoming Router Flow**

**7.4.2 Process Flow Steps:**

|  |  |
| --- | --- |
| **S. No** | IIB receives the request from channel. |
|  | ESB receives the request from channel. |
|  | Request channel is the http request accepts the request from the channel |
|  | The request message is placed in the RAW\_LOG\_SB\_REQ to log the request in Data Base (RAW\_AUDIT\_LOGGING) |
|  | According to the Channel Request message fields such as channel and processing code 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 processingCode, Channel 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 Funding Agent Lines Flow |

**7.4.3 StewardBank ESB Transaction Details**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **DB Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory(Yes/No)** | **Comments** |
| 1 | APPLICATION\_NAME | Transaction Name | Varchar | 50 | N | This will indicate transaction name |
| 2 | SERVICE\_NAME | Service Name | Varchar | 30 | N | This will indicate transaction name |
| 3 | PROC\_CODE | Processing Code | Varchar | 20 | Y | From input request |
| 4 | TCP\_REQ\_QNAME | TCPIP Request Queue | Varchar | 20 | N | This Request Queue Name field is used for ISO Related Application |
| 5 | TCP\_RES\_QNAME | TCPIP Response Queue | Varchar | 20 | N | This Response Queue Name field is used for ISO Related Application |
| 6 | HTTP\_REQ\_QNAME | HTTP Request Queue | Varchar | 20 | Y | This Request Queue Name field is used for REST Related Application |
| 7 | HTTP\_RES\_QNAME | HTTP Response Queue | Varchar | 20 | Y | This Response Queue Name field is used for REST Related Application |
| 8 | CHANNEL | Channel | Varchar | 20 | Y | Channel Name |
| 9 | MSGDOMAIN | Message Domain | Varchar | 20 | Y | Request Message Format |

**7.4.4 Outgoing Router Process Flow Diagram**



**Figure:6 Outgoing Router Flow**

**7.4.5 Process Flow Steps:**

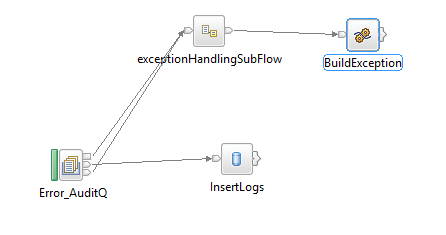
|  |  |
| --- | --- |
| **S. No** | **Activities** |
|  | The OutgoingHttpRes brings the Channel Response |
|  | Channel Response will be logged in RAW\_LOG\_SB\_REQ and insert in Data Base. |
|  | Finally Channel Response will be send to the Front End. |

**7.5 Error\_Audit Process Flow**

**7.5.1 Process Flow Diagram:**



**Figure:7** **Exception Subflow**



**Figure:8** **Error\_Audit\_Logging Flow**

**7.5.2 Process Flow Steps:**

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

# Interface Definitions

## Request Message Definition

### Channel (LoanEngine/Internet Banking Backend) Request Message Structure /Schema

**Request Type:** HTTP

**Request Format:** JSON

**Request URL:** <http://IPAddress:Port/v1/stewardBank/>

### 8.1.2 LoanEngine/Internet Banking Backend Request Message Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item # | Tag Name/ Field Name | Data Type | Mandatory(Yes/No) | Comments |
| 1 | Channel | String | Y | Channel name |
| 2 | Processingcode | String | Y | Processing code |
| 3 | Mobile Number | String | Y | Mobile number to check for the loan eligibility |

### 8.1.3 LoanEngine/Internet Banking Backend Sample Source Messages

|  |
| --- |
| {  "channel": "mobileapp",  "processingcode":"800010",  "mobilenumber":"263771222746"  } |

### 8.1.4 DWH Request Message Structure /Schema

**Request Type:** HTTP

**Request Format:** JSON

**Request URL:**  http://IPAddress:Port/jigsaw/serviceRequest/FundingAgentLines

### 8.1.5 DWH Request Message Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item # | Tag Name/ Field Name | Data Type | Mandatory(Yes/No) | Comments |
| 1 | Channel | String | Y | Channel name |
| 2 | Processingcode | String | Y | Processing code |
| 3 | Mobile Number | String | Y | Mobile number to check for the loan eligibility |

### 8.1.6 DWH Sample Source Messages

|  |
| --- |
| {  "channel": "mobileapp",  "processingcode":"800010",  "mobilenumber":"263771222746"  } |

### 8.1.7 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

### LoanEngine/Internet Banking Backend Response Message Structure/Schema

**Response Type:** JSON

### DWH Response Message Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Tag Name/Field Name** | **Data Type** | **Mandatory(Yes/No)** | **Comments** |
| 1 | NationalId | String | Y | Main Tag |
| 2 | LimitAmount | String | Y | Available limit |
| 3 | Account Numbers | String | Y | List of account numbers |

### 8.2.3 DWH Sample Response Messages

|  |
| --- |
| {  "nationalid": "22235190J22",  "limitamount": "600",  "accountnumbers": [  "1010743768",  "1002544397"  ]  } |

### 8.2.4 LoanEngine/Internet Banking Backend Response Message Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Tag Name/Field Name** | **Data Type** | **Mandatory(Yes/No)** | **Comments** |
| 1 | NationalId | String | Y | Main Tag |
| 2 | LimitAmount | String | Y | Available limit |
| 3 | Account Numbers | String | Y | List of account numbers |

### 8.2.5 LoanEngine/Internet Banking Backend Sample Response Messages

|  |
| --- |
| {  "nationalid": "22235190J22",  "limitamount": "600",  "accountnumbers": [  "1010743768",  "1002544397"  ]  } |

# 

# Queue Details

## Queue Details Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Queue Objects** | **Script** | | |
| Queues | Name | Purpose | Script Combined |
| NANOELE\_SB\_HTTP\_REQ | Receives the Messages from Channel through HTTP Router to ESB NanoLoanEligibilityCheck. | <Final Script Will be placed, after complete Unit Testing> |
| ALL\_SB\_HTTP\_RES | This is the Queue used to send to response back to the channel through Httprouter. |
| 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

## 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\_NUMBER | Retrieval Reference Number | Varchar | 20 | Y |  |

## 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 |
| 8 | TIME\_LOCAL\_TRANSACTION | Transaction Time | Varchar | 20 | Y | Time of the Transaction |
| 9 | DATE\_LOCAL\_TRANSACTION | Transaction Date | Varchar | 20 | Y | Date of the Transaction |
| 9 | RETRIEVAL\_REFERENCE\_NUMBER | Retrieval Reference Number | Varchar | 20 | Y | Retrieval Reference Number |
| 10 | BORKER\_ERROR\_CODE | Broker Error Code | Varchar |  | Y | Error code generate by Broker |
| 11 | USER\_DEFINE\_ERROR\_CODE | User Defined Error Code | Varchar |  | Y | User Defined Code |

## File Based Logging

### 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.