

Functional System Definition Of E-Pay SDK

August 2024

|  |  |
| --- | --- |
| **Document Control Sheet** | |
| **Current Version** | 1.1 |
| **Project Code** | NS\_SBI\_04\_08\_2024 |
| **Project Name** | SBI ePay Implementation |
| **Document Type** | ePay SDK |
| **Author** | Sagar |
| **First Reviewed By** | Neeraj Durgapal |
| **Frequency of Document Review** | Next Revision of Standard Documents |
| **Document Creation Date** | 19-08-2024 |
| **Last Updated Date** | 24-08-2024 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Version** | **Date** | **Name** | **Comments** |
| 1.0 | 22-08-2024 | Sagar | Draft |
| 1.1 | 24-08-2024 | Neeraj Durgapal | Approved |

[Document Purpose 3](#_Toc245526869)

[Scope 3](#_Toc1807032229)

[Functional Flow: 4](#_Toc850046997)

[1. FS\_SDK 1 - Token Generation 5](#_Toc1357829679)

[1.1 – Objective: 5](#_Toc333387708)

[1.2 - Token Generation: 5](#_Toc1932809297)

[1.3 - Token Generation Validations: 5](#_Toc1899955829)

[2. FS\_SDK 2 – Merchant’s Customer Creation: 6](#_Toc554490933)

[2.1 – Objective: 6](#_Toc172817142)

[2.3 - Functional Flow: 7](#_Toc1335181730)

[2.4 - Customer Creation Validations: 7](#_Toc463388595)

[3. FS\_SDK 3 – Order Creation: 8](#_Toc1259269358)

[3.1 – Objective: 8](#_Toc589400278)

[3.2 - Order Creation: 8](#_Toc1276849891)

[3.3 - Functional Flow: 8](#_Toc680542451)

[3.4 - Create Order Validations: 9](#_Toc1258201076)

[4. FS\_SDK 4 – Transaction Initiation: 10](#_Toc337103990)

[4.1 – Objective: 10](#_Toc1038007643)

[4.2 - Transaction Process: 10](#_Toc105636910)

[4.3 - Functional Flow: 10](#_Toc1610116092)

[5. FS\_SDK 5 – Transaction Status: 12](#_Toc1154167980)

[5.1 – Objective: 12](#_Toc1407260006)

[5.2 - Payment Status: 12](#_Toc526845397)

[5.3 - Functional Flow: 12](#_Toc1387345564)

[6. FS\_SDK 6 – Frontend SDK Payment Initiation: 12](#_Toc792330005)

[6.1 – Objective: 13](#_Toc1096767578)

[6.2 - Payment Initiation: 13](#_Toc870249060)

[6.2 - Validations: 13](#_Toc2104427431)

[6.3 - Functional Flow: 14](#_Toc520466076)

# Document Purpose

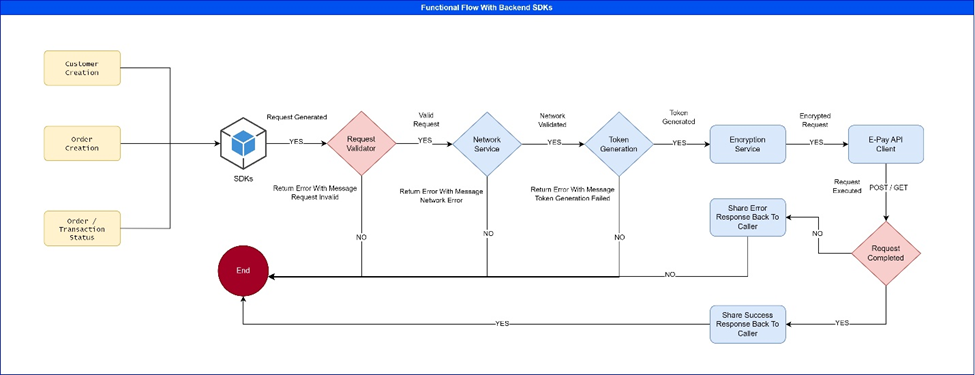
This document details the implementation of Mobile and Java SDKs for Merchant and Payment integration in the SBI ePay application.

# Scope

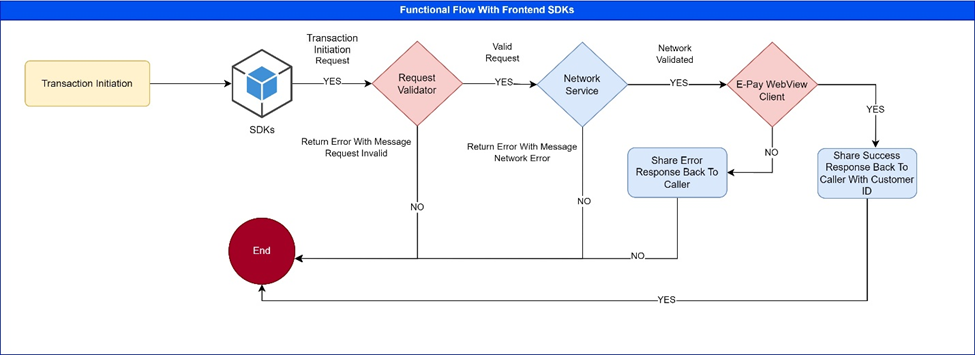
The core responsibilities

* Backend SDK
  + **Merchant’s Customer Creation –** The SDK will implement functionality for creating customers under a merchant MID.
  + **Encryption & Decryption –** The SDK will have capability to encrypt and decrypt the request as per AES256-GCM logic.
  + **Checksum –** The SDK will also have capability for checksum as per with SHA512 logic.
  + **Order Creation -** The SDK will provide the capability for merchants and their customers to create orders for transactions on the SBIePay platform.
  + **Transaction Initialization -** The SDK will enable the initiation of transactions for merchants and customers within the SBIePay platform.
  + **Order and Transaction Status –** The SDK will offer functionality to retrieve the status of orders using the Order Reference Number and ATRN Number.
  + **Cancellation & Refund with Status –** The SDK will offer functionality to initiate the cancellation and refund and retrieve the status of refund / cancellation using the ATRN Number.
  + **Supported Frameworks for Backend SDK**
    - **Phase 1**
      * Java
    - **Phase 2**
      * .Net
      * Python
      * Golang
      * PHP
* Frontend SDK
  + **Payment Checkout**
    - The SDK will enable the launching of Checkout Page as a WebView for the Merchant Customer to make the payment.
    - SDK will capture the following events during payment process:
      * Payment is marked as Successful
      * Payment is marked as Failed
      * Payment is marked as Cancelled
      * Browser Closed
        + Bank Page
      * Epay Webview
        + Browser Refresh
        + Bank Page
  + **Supported Frameworks for Mobile/Web SDK – Web view rendering**
  + Native Android – Kotlin & Java
  + Native IOS - Swift
  + Flutter – Dart
  + React Native

### Functional Flow with Backend SDKs:



Functional Flow with FrontEnd SDKs:



1. FS\_SDK 1 - Token Generation

#### – Objective:

* JWT Token will be generated via token generation API of Transaction Service for Request Type
  + Merchant Customer Creation
  + Merchant Customer Order Creation
  + Merchant Order Transaction

#### 1.2 - Token Generation:

* Customer / Order Creation
  + To generate a JWT token for Customer and Order creation within the SDK, we need to provide the merchant secret and API keys with request type in header.
* Transaction / Status
  + To generate a token for transaction and status, we need the hash value of Order Reference Number and MID, and request type.

#### \1.3 - Token Generation Validations:

* For Customer and Order Creation Token generation following parameters is required:

|  |  |  |
| --- | --- | --- |
| Parameter | Mandatory | Description |
| Merchant API Key | Yes | Merchant API Key id which was generated during onboarding and shared with Merchant |
| Merchant Secret Key | Yes | Hash Value of Merchant API Key secret which was generated during onboarding and shared with Merchant |
| Function Type | Yes | Customer, Order |

* For Transaction and Status token request should have the below parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Mandatory | Description |
| Order Ref Number + MID | Yes | Hash Value of combination of Order Ref Number and MID |
| Function Type | Yes | Transaction, Payment |

### 2. FS\_SDK 2 – Merchant’s Customer Creation:

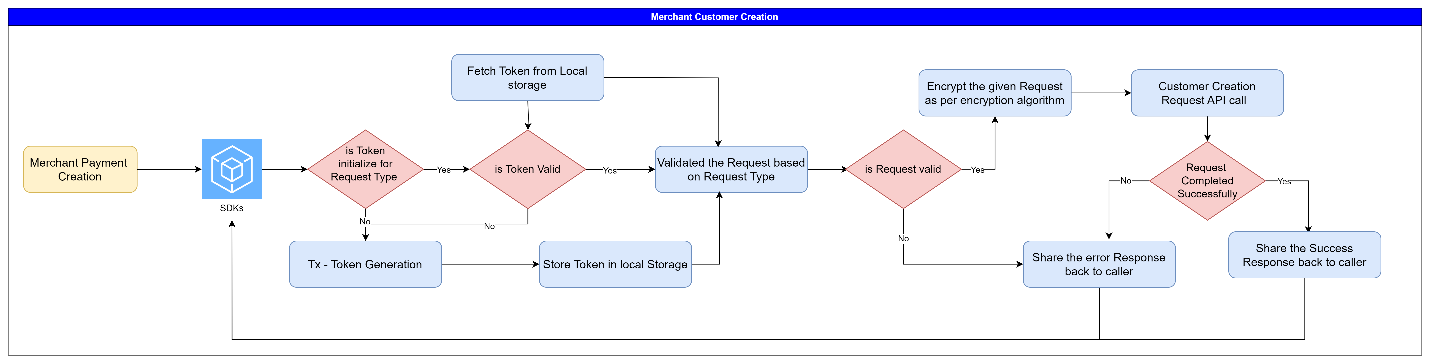
#### 2.1 – Objective:

The SDK will create a merchant customer using the created customer Transaction Service API.

*2.2 - Customer Creation:*

* The SDK first check is the merchant has been initialized and having the valid token in local storage.
* If a valid token is not present, then Transaction Token Generation API needs to call for getting the valid token for customer request Type.
* Customer Creation requests will be validated at SDK level.
* Valid Customer Creation requests will be encrypted.
* Transaction Service Customer Creation API will be called with valid Token and encrypted request.
* API response will be passed on to the caller.

#### 2.3 - Functional Flow:



#### 2.4 - Customer Creation Validations:

* For creating customer following parameters is required in encrypted request:

### 3. FS\_SDK 3 – Order Creation:

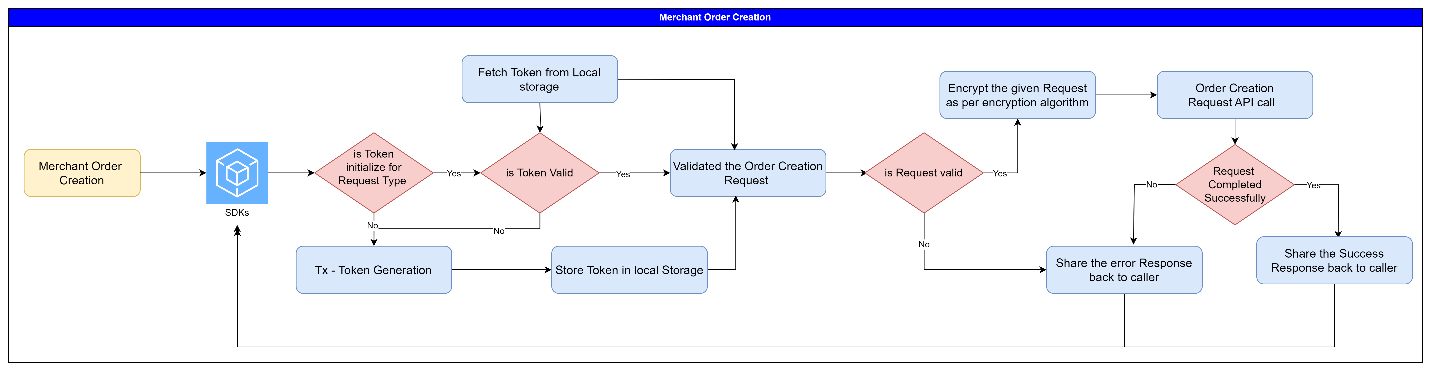
#### 3.1 – Objective:

The SDK will create an Order for merchant customers using the Order creation API of Transaction Service.

#### 3.2 - Order Creation:

* The SDK first check is the merchant has been initialized and having the valid token in local storage.
* If a valid token is not present, then the Token Generation API of Transaction Service needs to call for getting the valid token for Order request Type.
* Order Creation requests will be validated at SDK level.
* Valid Order Creation requests will be encrypted.
* Transaction Service Order Creation API will be called with valid Token and encrypted request.
* API response will be passed on to the caller.

#### 3.3 - Functional Flow:



#### 3.4 - Create Order Validations:

* For creating order following parameters is required in encrypted request:

### 4. FS\_SDK 4 – Transaction Initiation:

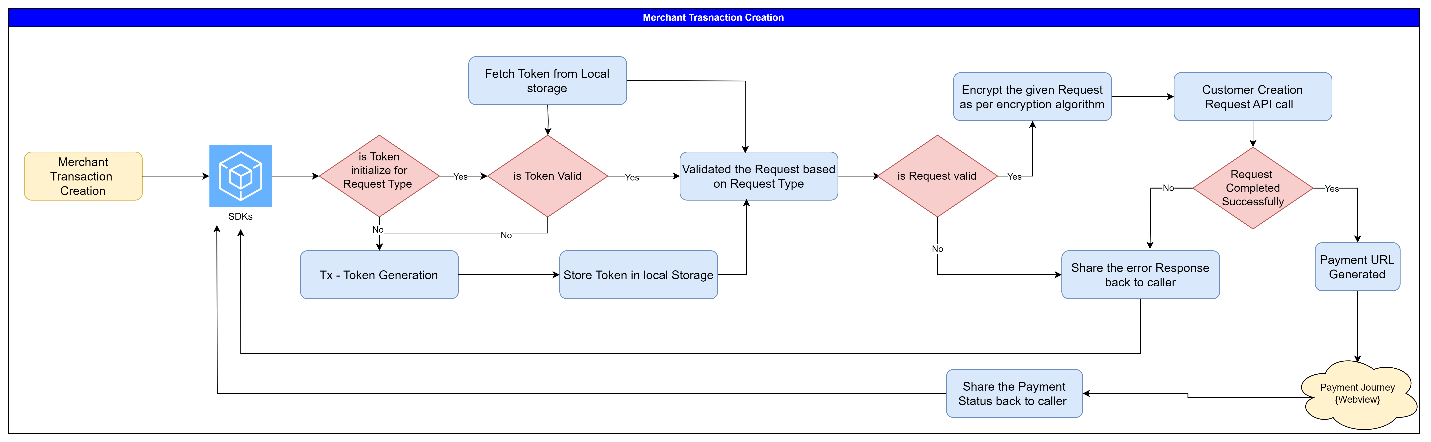
#### 4.1 – Objective:

The SDK will initiate the Transaction for merchant customers using the Transaction Creation API of Transaction Service.

#### 4.2 - Transaction Process:

* The SDK is designed to retrieve and Transaction the Order status.
* The SDK first check is the merchant has been initialized and having the valid token in local storage or payment URL is directly provided, then sdk will validate the request and proceed accordingly.
* If a valid token is not present, then Transaction Token Generation API needs to call for getting the valid token for Transaction request Type.
* Transaction Creation requests will be validated at SDK level.
* Valid Transaction Creation requests will be encrypted.
* Transaction Service Order Creation API will be called with valid Token and encrypted request.
* API response will be passed on to the caller.

#### 4.3 - Functional Flow:



### 5. FS\_SDK 5 – Transaction Status:

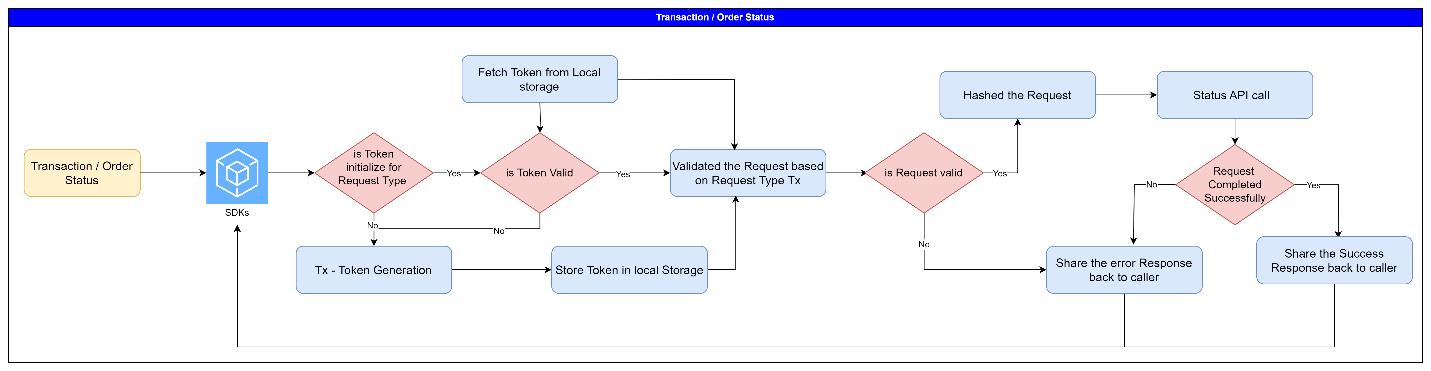
#### 5.1 – Objective:

The SDK will retrieve the Transaction/ Order Status for merchant customers using the Transaction Status API of Transaction Service.

#### 5.2 - Payment Status:

* The SDK is designed to retrieve and Transaction the Order status.
* The SDK first check is when the merchant has been initialized and having the valid token in local storage or request URL is directly provided, then SDK will validate the request and proceed accordingly.
* If a valid token is not present, then Transaction Token Generation API needs to call for getting the valid token for Transaction request Type.
* Transaction/ Order Status requests will be validated at SDK level.
* Valid Transaction/ Order Status requests will be encrypted.
* Transaction/Order Status API will be called with valid Token and encrypted request.
* For UPI Payment channel, Intent will be called and display all UPI payment apps to user.
* Frontend Mobile SDK have a provision to auto read SMS functionality through which OTP will be auto fetch and fill for payment channels.
* API response will be passed on to the caller.

#### 5.3 - Functional Flow:



### 6. FS\_SDK 6 – Frontend SDK Payment Initiation:

#### 6.1 – Objective:

The SDK will initiate the Payment for merchant’s customers using the valid Payment URL responded by Order creation.

#### 6.2 - Payment Initiation:

* The SDK first checks for the valid payment URL.
* If the payment URL is not valid, then an error response is sent back to the caller.
* If the payment URL is valid then SDK will open this URL in webview chrome client and loads the checkout page
* Users can choose the payment mode as per their choice and complete the payment.
* SDK will capture the following events and accordingly send the response to Merchant
  + Payment Success
  + Payment Failed
* Event listeners need to be implemented in Checkout Page if initiated by SDK.

#### 6.2 - Validations:

|  |  |  |
| --- | --- | --- |
| Parameter | Mandatory | Description |
| URL | Yes | A Valid Uniform Resource Identifier for Epay Domain |
| Hash | Yes | A valid hash token |
| Language | No | Language which merchant sends, defaults to English |

#### 6.3 - Functional Flow:

