About the Company

IME Digital Solution Ltd. is a subsidiary of IME Group established under the payment and settlement bylaws 2072, with an aim to provide digital financial services to customers in partnerships with banks/FIs as well as telecom operators and aggregators, deployed through self-service as well as agent-based delivery channel.

IME Digital intends to introduce mobile money service, with name IME Pay, under its digital financial services initiative and offer a wide range of payment services to remittance customers existent with money transfer associate company IME Ltd. and also cater to the vast majority of unbanked and under-banked customers of Nepal. IME Pay can be accessed via mobile and can be serviced through any of the 20000 plus strong agent network of IME and beyond.

Introduction

This document describes the process for successfully integrating IME Pay into your transaction process, explaining how to facilitate communication between your mobile app and the IME pay.

Payment via IME Pay SDK completes in merely 4 steps:

1. Customer selects IME Pay as the payment method in merchant's app.
2. Customer Enters wallet number and PIN.
3. Customer verifies the OTP
4. Customer completes the transaction.

Note: This document helps to integrate IME Pay SDK to your mobile app (Android, iOS & Flutter)

Prerequisites

Merchants need to develop the following APIs to integrate the IME Pay Merchant SDK into their app.

1.Merchant Recording API:

Merchants who accept payments directly from a mobile app are considered special merchants. They must create a transaction-recording API and provide its URL as a parameter when calling the SDK. The SDK will internally post data to this API using the specified parameters. Based on the response from this API, the SDK will determine whether to proceed with the payment.

Request Params posted by SDK:

```
{
"MerchantCode": "M0001",
 "ReferenceId": "Ref-9901",
"TokenId": "201612151220041022",
"Amount": "1035"
}
```

Response Paramenters:

```
{
"ResponseCode":"0",
"ReferenceId":"Ref-9901",
"ResponseDescription": "Payment Request Recorded"
}
```

The merchant's payment recording URL should capture and store the values received from the SDK request. To achieve this, the merchant must maintain a database table to log each request from the SDK. This table should include the required columns in the specified format. Storing transaction records in this manner enables the merchant to review and resolve any failed transactions, which may occur for various reasons. This process ensures failed transactions can be

verified and completed. The SDK will proceed with the payment only if the response is successful and returns a ‘ResponseCode’ of 0.

Below is the table structure to store the parameters from the recording URL request, along with other parameters that will be posted in later stages of the integration.

Column Name	Data Type	Description
MerchantCod	VARCHAR(10)	**Will be provided by IME Pay**
TranAmount	MONEY	**Amount involved in the transaction**
RefId	VARCHAR(20)	**Unique id generated by merchant’s system**
TokenId	VARCHAR(20	**Will be provided to SDK by IME Pay at the end of Step 2**
TransactionId	VARCHAR(20)	**IME Pay generated unique TransactionId**
Msisdn	VARCHAR(20)	**IME Pay Wallet ID**
TranStatus	TINYINT	**Txn Status at IME Pay End**
StatusMessage	VARCHAR(50)	**Txn Status Message**
RequestDate	DATETIME	**Txn Requested DateTime (Time of Recording)**
ResponseDate	DATETIME	**Txn Response received DateTime (Time of Txn Response)**

2. Delivery Url

This API must be developed by the merchant. It serves as a webhook URL triggered by the IME Pay server upon transaction completion, regardless of success or failure. The API should not require any authorization restrictions. We strongly recommend merchants rely on the data posted through this API to determine the payment status.

Request JSON parameters will be as below:


```
{
"MerchantCode":"M0001",
"RefId":"Ref-9901",
"TokenId":"201612151220041022",
"ResponseCode":"0",
"Msisdn":"9800007788"
"TransactionId":"201609271720522269"
"ResponseDescription": "Success"
}
```


Response JSON values will be as below:


```
{
"MerchantCode":"M0001",
"RefId":"Ref-9901",
"TokenId":"201612151220041022",
"ResponseCode":"0"
}
```


ResponseCode:

- 0 → Success
- 1 → Not found or failed

Next, update the database record created earlier (identified by **RefId**) with the following:

- **TransactionId**
- **Msisdn**
- **TranStatus** using the **ResponseCode**
- **StatusMessage** using the **Response Description**.

Once you are ready with the Merchant Recording API & Delivery API, Let's dive into the SDK implementation part.

Step 1: Import SDK to your project

Android: *implementation 'np.com.imepay:payment-sdk:1.0.1'*

iOS: *pod 'IMEPay'*

Flutter: *imepay_merchant_sdk*

Step 2: Initialize SDK

For Android:

Using the instance of IME Pay, call the method performpaymentV1().

```
IMEPayment imePayment = new IMEPayment(activity.this, ENVIRONMENT.LIVE);
imePayment.performPaymentV1("MERCHANT_CODE", "MERCHANT_NAME" ,
"RECORDING_SERVICE_URL" , "DELIVERY_SERVICE_URL", "AMOUNT",
"REFERENCE_ID", "MODULE" , "USERNAME", "PASSWORD", new
IMEPaymentCallback() {

@Override public void onSuccess(int responseCode, String transactionId, String
msisdn, String amount, String refId) {}

@Override public void onTransactionCancelled(String refID){}
});
```

For iOS

```
let manager = IMPPaymentManager(environment: Live) // For production

let manager = IMPPaymentManager(environment: Test) // For Test
manager?.pay(withUsername: "username" , password: "password", merchantCode:
"merchantCode", merchantName: "merchantName", merchantUrl: "merchantUrl", amount:
"amount", referenceId: "referenceId", module: "module",

 success: { (transactionInfo) in
transactionInfo.responseCode
transactionInfo.responseDescription
transactionInfo.transactionId
transctionInfo.customerMsisdn
transctionInfo.amount
transactionInfo.referenceId
},

failure: { (transactionInfo, errorMessage) in
})
```

For Flutter

```
var result =await StartSdk.callSdk(context, merchantCode: MERCHANT_CODE,
merchantName: MERCHANT_NAME, merchantUrl: MERCHANT_URL, amount: AMOUNT,
refId: REF_ID, module: MODULE, user: USER, password: PASSWORD, deliveryUrl:
DELIVERY_URL, buildType: BuildType.STAGE);
```

Note:

** [Username, Password, Module, MerchantCode, MerchantName] will be provided by IME pay at the time of registration.*

** The TokenId should be valid for a single use only, with a new TokenId generated for each transaction. This identifier should be created by the merchant and act as a unique reference for the transaction from the merchant's perspective.

Step 3:

After Step 2, the SDK launches and prompts the user to enter their IME Pay wallet number and PIN. Upon authorization, the user is required to input the OTP sent to the provided mobile number. Once the OTP is successfully verified, the SDK triggers the payment API to process the transaction. Upon completion, the SDK

sends a callback to the mobile app with the transaction status, indicating either success or failure.

- onSuccess :

Once the final payment transaction is initiated by the SDK after completing all authentication steps, this method is triggered by the SDK, regardless of whether the transaction succeeds or fails, using the parameters outlined below-:

Response Code 100 : Transaction successful.
Response Code 101 : Transaction failed.
TransactionId : Unique ID generated from IME Pay system (if successful)
Amount: Amount paid by Customer
RefID: Reference ID pass at the time of initilizing SDK
Msisdn : IME Pay Wallet Number (Mobile number entered by user)

- onTransactionCancelled :

Before the final payment transaction, if the user cancel the payment process by clicking the back button or anything else, this method gets invoked by the SDK with the parameter described above.

In the SDK's OnSuccess callback, it is strongly advised that the merchant should not treat this callback as the final transaction outcome. Instead, the merchant's mobile app should send transaction completion information, including the reference ID and other details, to the merchant server. The merchant server should then either verify the transaction status using delivery API data or perform the recheck process (Step 4) and respond accordingly to the mobile app. The mobile app should display the final outcome, confirming the success or failure of the purchase, only after receiving this response.

Step 4:

If the IME Pay system fails to trigger the merchant’s delivery URL for any reason, the merchant must verify the transaction status by calling the recheck API using

the parameters saved from the delivery URL. This API should be implemented on the merchant's server, not within the mobile app.

API URL to recheck confirmation with type = ‘POST’

Test Env: *https://stg.imepay.com.np:7979/api/Web/Recheck*

Live Env: https://payment.imepay.com.np:7979/api/Web/Recheck

Also Authorization header and Module* should be sent to API for authentication purpose. The same username and password provided by IME Pay team should be used in header.

Authorization = Basic {username:password} *

Module = {Module} **

NOTE:

* Values should be encoded using Base64.

** Module will be provided by IME pay.

Request JSON parameters will be as below:

```
{
"MerchantCode":"M0001",
"RefId":"Ref-9901",
"TokenId":"201612151220041022"
}
```

Response JSON values will be as below:

```
{
"ResponseCode":"0",
"Msisdn":"9800007788"
"TransactionId":"201609271720522269"
"ResponseDescription": "Success",
}

ResponeCode: 0 -> Success
ResponeCode: 1 -> Not found or failed
```

Now, update the database table record created above identified by RefId with TransactionId, Msisdn, and TranStatus as response code and StatusDetail as Response Description.

Test Data:

Merchant Code: DEMOIMEP

Merchant Name: IME Pay Demo

Merchant Url: https://stg.imepay.com.np:7979/api/sdk/recordTransaction

Delivery Url: http://172.20.22.11:1717/api/sdk/deliveryService

Merchant Module: DEMOIMEP

Username: demoimepay

Password: IMEPay@123

During the integration process, the merchant will need to enter an IME Pay test number. To obtain this, the merchant should contact the IME Pay representative listed below to register one of their mobile numbers as an IME Pay test number. This will enable them to receive OTP and simulate the entire transaction process through the SDK.

Contact Person:

Bibek Chaudhary

9801190834

bibek.chaudhary@imepay.com.np