Web-based offering Integration Integration Document v1.0

Document Summary: -

Item	Remarks
Organization	Synergistic Financial Networks Pvt. Ltd.
Document Name	Integration Document
Created By	Swapnil H Gawde

Document Revision and Review History: -

Date	Version	Prepared By	Reviewed By	Summary of Change
28-03-2024	v1.0	Swapnil H Gawde	Mr. Swapnil Shelke & Mr. Hatim Motorwala	

Contents

1. Introduction

• Purpose

This document is intended to guide the development team in integrating the necessary APIs for remote payment initiation using a web API mechanism with the Mosambee platform.
This document assumes that the team is familiar with and understand implementing the following:

- JSON API.
- HMAC using SHA512.
- Encryption and Decryption using AES GCM mechanism.

• Scope

This documentation covers the integration guidelines and will be facilitated to the clients who are interested in opting for web-based offering from Mosambee.

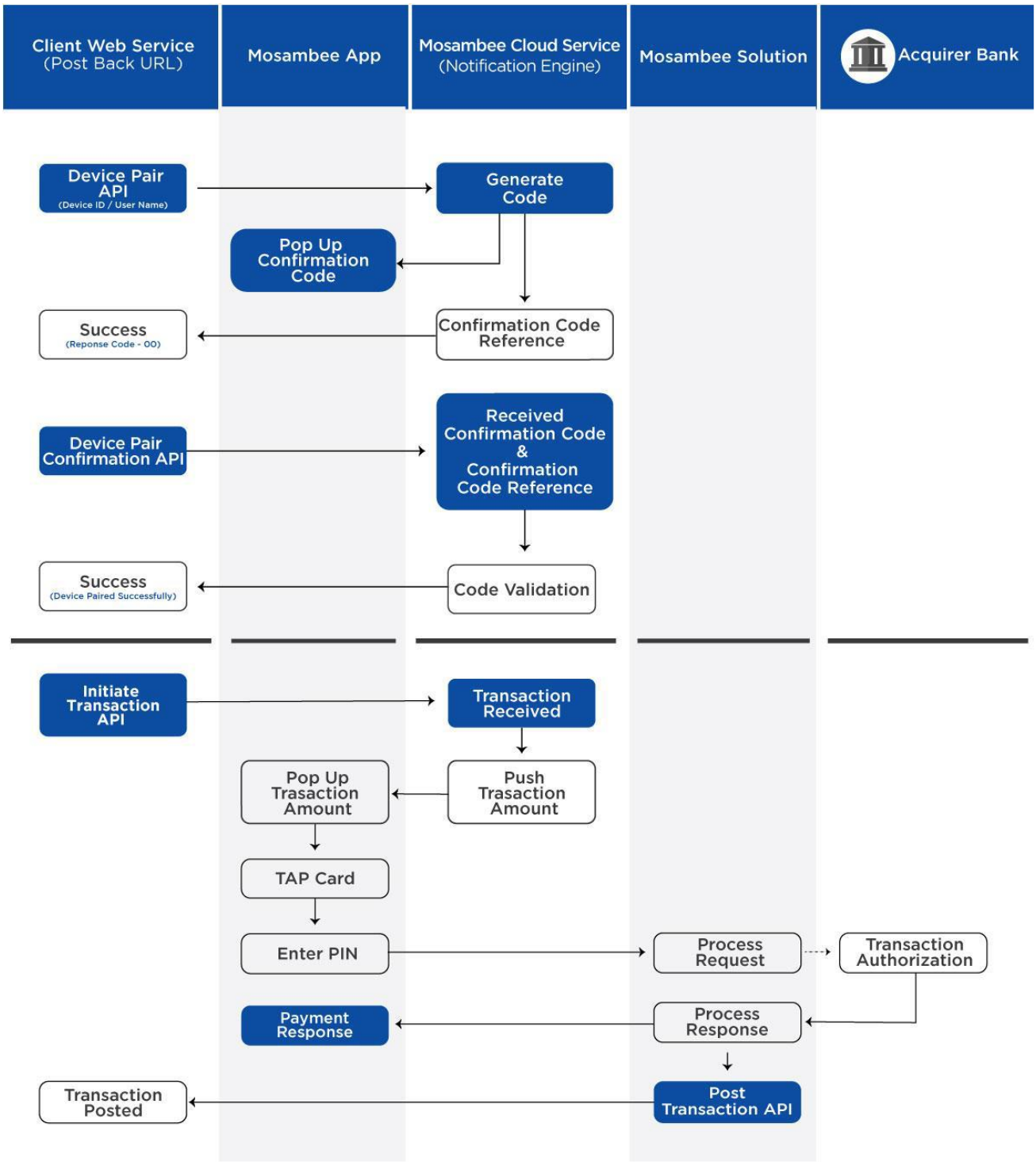
• Integration Overview

Below are the steps to be followed for end to end integration:

- The Mosambee server offers essential device pair/unpair APIs, crucial for pairing devices with users and establishing user identification.
- The Device Pair API serves the purpose of pairing a device, facilitating the connection between the device and a specific UserName or User ID.
- HMAC (Hash-based Message Authentication Code) will be applied for authentication, utilizing a hash function with SHA512 and a secret key on the data.
- Device pair confirmation API will override any existing paired of device identifier (UserName).
- Device Pair Confirmation API will be used to replace UserName or unpair / pair user with new device Serial number.
- Client server initiates a paring request through Device Pair API with parameters like deviceSerialNo, deviceIdentifier (UserName) which contains the HMAC with request payload and secret key for authentication.
- The Mosambee notification engine authenticates the key in the response, confirming the device pairing to the client-server.
- The client-server initiates a transaction through the Initiate Transaction API, including parameters such as userName, password, mobile number (optional), invoiceNo, amount, and session key (mandatory).
- The transaction, once received at the Mosambee notification engine, pushes the transaction amount to the Mosambee App.
- The Mosambee app will prompt the user to confirm the transaction and tap a card for processing.
- The Mosambee app will send a request to the Mosambee solution, which processes the request to the acquirer bank for authorization.
- The Mosambee solution receives a response from the acquirer bank and provides the transaction result in response to the Mosambee app.
- At the same time, the Mosambee solution will provide the transaction result to the client server by posting a transaction using the Post Transaction API.

2. System Architecture Overview

Below is the overview of the Web API integration:



3. API Specification:

This section outlines the request and response fields, along with a description of each API, including sample date.

API Parameter Specification:

The JSON API must include the following request header parameters as per the requirement. Any invalid parameters will be ignored, and the request may be discarded by the service.

Request Header:

Parameter	Type	Description	Mandatory
apiClientRef	String	API client reference.	Y
DateTime	Timestamp(22)	Timestamp including milliseconds.	Y
hmac	String (2048)	HMAC value of request payload.	Y

apiClientRef: Shared by Mosambee (Synergistic Financial Networks Pvt. Ltd) with client after successful onboarding.
dateTime: Timestamp in milliseconds. Ex – 1655780920129, if date time is Tuesday, June21, 2022 3:08:40.129 AM.
hmac: Client generate HMAC with shared secret key and request body payload, refer code sample section for generation of HMAC in Java.
algorithm: HmacSha512
hmac = encodeBase64(HMAC ('request payload', 'secret key'))

a. Device Pair API

This API allow client to pair device with device identifier (username).
HTTP Method: POST
API: https://uat.notifypro.in/api/device/v1/terminal/pair
Content Type: application/json
Accept: application/json

Request Parameter:

Parameter	Type	Description	Mandatory
deviceSerialNo	String (120)	Terminal device ID.	Y
deviceIdentifier	String (120)	Unique identifier for device mapping, generated once partner on-boarded successfully, *(username).*	Y
acquirerId	String	refers to the acquirer identifier reference	O
subAcquirerId	String	refers to the sub-acquirer identifier reference	O

Sample Request:

{
"deviceSerialNo":"132857",
"deviceIdentifier ":"7398949487"
"acquirerId": "388",
"subAcquirerId": "1144896"
}

Response Parameter:

Parameter	Type	Description	Mandatory
result	String (12)	Result of request, success or failed.	Y
message	String (1024)	Error message if any.	O
responseCode	Number (6)	Response code.	Y
confirmationCodeReference	Number (6)	Confirmation code reference.	C

Sample Response:

{
"result": "success",
"message": "confirmation code published successfully.",
"responseCode": "00",
"confirmationCodeReference": "254"
}

Device Pair API Curl request sample:

```
curl --location 'https://uat.notifypro.in/api/device/v1/terminal/pair' \
--header 'apiClientRef: MOSAMBEE' \
--header 'hmac:
f2t9kvntDSSsT1bnpI+cZwIQieoqRp5iblmTwN+yY77z4ZdKo/onP/djFD8hrlU/iOC9e8SScUv44qP/NV4EIQ==' \
--header 'Content-Type: application/json' \
--data '{"deviceSerialNo":"541265748","deviceIdentifier":"96369636912"}'
```

b. Device Pair Confirmation API

This API is used by client to complete device pairing using received confirmation code on device(terminal).
HTTP Method: POST
API: https://uat.notifypro.in/api/device/v1/terminal/pair/validate
Content Type: application/json
Accept: application/json

Request Parameter:

Parameter	Type	Description	Mandatory
confirmationCode	Number (6)	Received on device, 4 or 6-digits numeric value.	Y
confirmationCodeReference	Number (20)	Received in device pair API response.	Y
deviceSerialNo	String	Device serial number	Y
acquirerId	String	refers to the acquirer identifier reference	O
subAcquirerId	String	refers to the sub-acquirer identifier reference	O

Sample Request:

{
"confirmationCode": "5462",
"confirmationCodeReference": "254",
"deviceSerialNo":"14132014",
"acquirerId": "388",
"subAcquirerId": "1144896"
}

Response Parameter:

Parameter	Type	Description	Mandatory
result	String (12)	Result of request, success or failed.	Y
message	String (1024)	Error message if any.	O
responseCode	Number (6)	Response code.	Y

Sample Response:

{
"result": "success",
"message": "device paired successfully.",
"responseCode": "00"
}

Note: Device pair confirmation API will override any existing paired device identifier.

Device Pair Confirmation API Curl request Sample:

curl --location --request **POST** 'https://uat.notifypro.in/api/device/v1/terminal/pair/validate' \
–header 'apiClientRef: MOSAMBEE' \
–header 'hmac:
aJMTWkZfQ9kbC2pjeytAuWd8L9BuvGwuh/oTV9jZV+11sfGA+Fa6x7YaClznPToUypvY3GmCdg4raSC+KibWhQ
==' \
–header 'Content-Type: application/json' \
–data-raw '{ "confirmationCode": "6545", "confirmationCodeReference": "100490", "deviceSerialNo":
"541265748" }'

c. Initiate Transaction API

This API is used by client to initiate transaction on device (terminal).

HTTP Method: POST
API: https://uat.notifypro.in/api/transaction/v1/initiate
Content Type: application/json
Accept: application/json

Request Parameter:

Parameter	Type	Description	Mandatory
transactionDetails	String	Encrypted request data in HEX String format, refer following table for request fields.	Y

TransactionDetails fields:

Parameter	Type	Description	Mandatory
userName	String	Terminal Username.	Y
password	String	Terminal user pin / password.	Y
customerMobileNo	Number (10)	Customer mobile number (Mandatory for WALLET payment).	C
customerEmail	Email	Customer email address (required for SMSPay transaction if payment link need to send on email).	C
invoiceNo	String	Merchant reference number.	Y
transactionAmount	Decimal (8, 2)	Transaction amount.	Y
additionalAmount	Decimal (8, 2)	Tip Amount (mandatory for Sale+Tip transaction).	C
walletProvider	String	Wallet provider code.	C

transactionType	String	Transaction type.	Y
upiAddress	String	Customer VPA (mandatory for UPI collect based transaction).	C
payerName	String	Payer name (mandatory for cash or cheque transaction).	C
chequeNo	String	Cheque No (mandatory for cheque transaction)	C
autoInitiateFlag	String	Transaction auto initiate flag (True/False), by default - True	Y
acquirerId	String	refers to the acquirer identifier reference	O
subAcquirerId	String	refers to the sub-acquirer identifier reference	O

Sample Request:

{
"userName": "7666297333",
"password": "1321",
"customerMobileNo": "9923097475",
"invoiceNo": "123425345435",
"transactionAmount": "1000",
"transactionType": "22",
"upiAddress": "9923097475@ybl",
"payerName": "swapnil",
"autoInitiateFlag": "true",
"acquirerId": "388",
"subAcquirerId": "1144896"
}

Encrypted Sample Request:

{
"transactionDetails":"012C03B8F295CA6AC0AA519D949656356C6"
}

Encrypted Sample Request:

Encryption Algorithm: AES/GCM/NoPadding
Key Size: 256
GCM IV Length: 12
GCM Tag Length: 16

Response Parameter:

Parameter	Type	Description	Mandatory
result	String	Result of request success / failed.	Y
responseCode	String	Response code.	Y
message	String	Error message if any.	O
notificationDetails	String	Object	O

notificationDetails Parameter:

deviceCategory	String	Terminal or non terminal	O
deviceSerialNo	String	Device serial number	O
transactionId	String	Web transaction id	O

Sample Response:


```
{
   "result": "success",
   "message": "transaction details published successfully.",
   "responseCode": "00",
   "notificationDetails": [
      {
         "deviceCategory": "0",
         "deviceSerialNo": "141903232"
      }
   ],
   "transactionId": 3045
}
```


Initiate Transaction Curl request Sample:

curl --location --request POST 'https://uat.notifypro.in/api/transaction/v1/initiate'
–header 'apiClientRef: MOSAMBEE'
–header 'hmac:
W5XMXYnDfQqqKtZSnAtPfIF8LmN9XK8IUeOpfH0Wf1KjQCX2zIQEeCroDR0cQaBg8A25b86FG7kIxBKU6sO8Dg
=='
–header 'Content-Type: application/json'
–data-raw
'{ "transactionDetails":
"012E56EBB7AB4ED1EF17ED46D0060955BE0DB90AC0B1828EBDDDAF52C49808061A5E6F9CBB8C0ECA6B
B50DF780DE64C326E74FE6ED9999597F903741251C39BFED30C33510D7C3046129D30AC156719B4E6F8E7
601665137F9B69586CE79B202B27EA6F30776D28CB80603CBC8DF7065398D45F273FEFC0A1AAE9907E783
92E900B9DD55A84E68D6FB2D2BB6A55C7EE04B3039D82721F97DCE530EFC374F3CB7DBED2DBDFCFD690
009B835C899AADC42DDC20865F0B066245B6D797E2D3AD03817A61AC0D96420EAFFEBCB368F2C0272CB
4C0B02E31238EF64AF183A" }'

d. Cancel Transaction API

This API is used by client to cancel transaction on device (terminal).

HTTP Method: POST
API: https://uat.notifypro.in/api/transaction/v1/cancel
Content Type: application/json
Accept: application/json

Request Parameter:

Parameter	Type	Description	Mandatory
notificationIdentifier	String	Unique identifier for device mapping, generated once partner on-boarded successfully	Y
webTransactionId	String	this will be web transaction ref identifier which will be received in the web initiate api response	Y
transactionMessage	String	declined transactions request message	Y
acquirerId	String	refers to the acquirer identifier reference	O
subAcquirerId	String	refers to the sub-acquirer identifier reference	O

Sample Request:

```
{
   "notificationIdentifier": "141297816",
   "webTransactionId": "100003001",
   "transactionMessage": "declined",
   "acquirerId": "388",
   "subAcquirerId": "1144896"
}
```

Response Parameter:

Parameter	Type	Description	Mandatory
result	String (12)	Result of request, success or failed.	Y
message	String (1024)	Error message if any.	O
responseCode	Number (6)	Response code.	Y
notificationIdentifier	String (up to 120 characters)	Unique identifier for device mapping, generated once partner on-boarded successfully	Y
webTransactionId	String (up to 120 characters)	this will be web transaction ref identifier which will be received in the web initiate API response	Y

Sample Response:


```
{
   "result": "success",
   "message": "qr request cancelled.",
   "responseCode": "00",
   "notificationIdentifier": "141297816",
   "webTransactionId": "100003001"
}
```


Fail transaction sample response:


```
{
   "result": "failed",
   "message": "device not found or not active.",
   "responseCode": "105"
}
```


Cancel Transaction Curl request Sample:

```
  curl --location --request POST 'https://uat.notifypro.in/api/transaction/v1/cancel'
–header 'apiClientRef: MOSAMBEE'
–header 'hmac:
4a6Pr3kcLNsVF8wtvgJkjzylDyPHQWpf+gfeGZ2LAabz0qI/SZVFRB6hnMnrb3+Wi21ocoPawKvVXXPjvfE
qdw==' \
–header 'Content-Type: application/json'
–data-raw
'{ "notificationIdentifier": "141297816", "webTransactionId": "100003001", "transactionMessage":
"declined" }'
```

4. Code Samples

HMAC Java Sample Code

```
import java.security.GeneralSecurityException;
import java.util.Base64;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
/**
<p>Computes a Hash-based Message Authentication Code (HMAC) using the SHA512
hash function.
</p>
*/
public class HMACUtil {
private HMACUtil() {}
public static String computeHMAC(String message, String signatureKey)
throws GeneralSecurityException {
byte[] signatureKeyInByteArray = hexStringToByteArray(signatureKey);
SecretKeySpec secretKeySpec =
new SecretKeySpec(signatureKeyInByteArray, "HmacSHA512");
Mac mac = Mac.getInstance("HmacSHA512");
mac.init(secretKeySpec);
Mobilizing Transactions Transforming Business 10 return Base64.getEncoder()
.encodeToString(mac.doFinal(message.getBytes()));
}
```

```
private static byte[] hexStringToByteArray(String hexString) {

int len = hexString.length();
byte[] data = new byte[len / 2];
for (int i = 0; i < len; i += 2) {
data[i / 2] = (byte) ((Character.digit(hexString.charAt(i), 16) << 4)
+ Character.digit(hexString.charAt(i + 1), 16));
}
return data;
}
  }
```

AES GCM sample code

```
import java.security.GeneralSecurityException;
import javax.crypto.Cipher;
import javax.crypto.spec.GCMParameterSpec;
import javax.crypto.spec.SecretKeySpec;
/**
* <p>Crypto utility for AES GCM encryption and decryption.</p>
*/
public class AESGCMUtil {
public static final int AES_KEY_SIZE = 256;
public static final int GCM_IV_LENGTH = 12;
public static final int GCM_TAG_LENGTH = 16;
private AESGCMUtil() {}
public static String encrypt(String plainText, String secKey, String ivData)
throws GeneralSecurityException {
Cipher cipher = buildCipher(secKey, ivData, Cipher.ENCRYPT_MODE);
byte[] cipherText = cipher.doFinal(plainText.getBytes());
return byteToHexString(cipherText);
}
public static String decrypt(String encHexString, String secKey,
String ivData) throws GeneralSecurityException {
Cipher cipher = buildCipher(secKey, ivData, Cipher.DECRYPT_MODE);
byte[] decryptedText = cipher.doFinal(hexToByteArray(encHexString));
return new String(decryptedText);
}
private static Cipher buildCipher(String secretKey, String ivData,
int cipherMode) throws GeneralSecurityException {
SecretKeySpec keySpec =
new SecretKeySpec(DataConversionUtil.hexToByteArray(secretKey), "AES");
GCMParameterSpec gcmParameterSpec = new GCMParameterSpec(
GCM_TAG_LENGTH * 8, DataConversionUtil.hexToByteArray(ivData));
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
cipher.init(cipherMode, keySpec, gcmParameterSpec);
return cipher;
}
```

5. Response code details:

Response Code	Message
0	Request processed successfully; actual message depend on API.
100	message authentication failed.
101	invalid request.
102	invalid API client ref value.
105	device not found or not active
235	device already in un-paired state.
303	device already paired with {identifier}.
306	device is blocked.
307	device not active for pairing.
305	device not active for requested identifier.
700	QR not published.
706	qr cancel request unable to processed
00	qr request cancelled

Transaction Type:

Transaction Type	Discription
1	Sale
2	Void
22	UPI Id
24	BQR