

EPOS API Integration Document

v1.0.2

Contents

Contents	2
1. Introduction	4
2. Inter-application communication	5
3. Pre-requisites	6
4. Request – Header Information	6
5. Response – Header Information	6
6. API Details	8
6.1 DoTransaction	8
7. API integration Process for Billing App	19
8. Glossary	21
9. Security Requirement	22
10. Pine Labs validation testing process	23

REVISION HISTORY

Version	Modified By	Description	Date
1.0	Pine Labs	Initial document creation	26 th December 2018
1.0.1	Pine Labs	Updated Constants in Section 8	5 th March 2019
1.0.2	Pine Labs	Updated Security Req.	5 th July 2019

1. Introduction

This integration guide will help with follows:

- Carry-out sale transactions using multiple payment modes – credit-card / debit-card via SMS link, Wallets, Bharat QR, UPI.

2. Inter-application communication

Billing application will communicate with *EPOS APIs* for transactions. For this communication, it will use *Messenger over Bound Service*.

In this process, the service defines a *Handler* that responds to different types of *Message* objects. This *Handler* is the basis for a *Messenger* that shares an *IBinder* with the client, allowing the client to send commands to the service using *Message* objects. Additionally, the client defines a *Messenger* of its own to send messages back. This technique allows the apps to perform Inter-Process Communication (IPC).

3. Pre-requisites

EPOS App must be logged-in and activated before calling any API mentioned in this document. If it is not in logged-in state during API call, it will display login screen to accept user credentials. On successful login, the API request will be processed.

4. Request – Header Information

Below are the parameters of *Header* which will be common for all API requests.

Parameter Name	Description	Data Type	Is Mandatory
ApplicationId	Unique application Id issued by Pine Labs	String (100)	Yes
UserId	Billing app user-Id/name	String (100)	No
MethodId	Unique Method Id. Refer Method List Sec 9.2	String (10)	Yes
VersionNo	API version number. For e.g. "1.0"	String (10)	Yes

Sample Request

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1001",
    "VersionNo": "1.0"
  }
}
```

5. Response – Header Information

Below are the parameters of *Header* data which will be common for all API responses.

Parameter Name	Description	Data Type
ApplicationId	Unique application Id issued by Pine Labs	String (100)
UserId	Billing app user-Id/name	String (100)
MethodId	Unique Method Id. Refer Method List Sec 9.2	String (10)
VersionNo	API version number. For e.g. "1.0"	String (10)

Below are the parameters of *Response* data which will be common for the entire APIs response.

Parameter Name	Description	Data Type
ResponseCode	Response code	String (10)

ResponseMsg	Response message	String (255)
-------------	------------------	--------------

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1001",
    "VersionNo": "1.0"
  },
  "Response": {
    "ResponseCode": "00",
    "ResponseMsg": "Success"
  }
}
```

6. API Details

6.1 DoTransaction

This API will be called when the Billing App completes the product selection and is ready to accept payment from the customer. Billing App will add all required tender options in its App, and call this API with specific tender such as Sale, Prepaid redeem etc.

This API can also be used for Load, Activation, Void transaction, etc. refer to complete list of transactions supported in **Section 9.3**.

Request:

S. No.	Tag Name	Description	Data Type	Is Mandatory
1	TransactionType	The type of payment transaction to be processed. Refer Sec 9.3 for all possible values.	Long	Yes
2	BillingRefNo	Transaction reference number from external application. EPOS will use this value for printing on chargeslip.	String(10)	No
3	PaymentAmount	Amount to be charged to customer – expressed as a whole number in lowest currency unit (i.e. in paise)	Long	Yes
4	BankCode	The acquirer bank code / Host Type to which transaction will be routed. Optional in case of sale transaction if Automatic Acquirer Selection is chosen	String(2)	No
5	CardNumber	Mobile number field. If empty then App will ask for mobile number input.	String(76)	No
6	EntryMode	Mobile Number – 01 Barcode – 02 Manual Entry - 03	String(37)	No
7	InvoiceNo	If independent transaction, then it is not required. Else in case of dependent transaction, it is the Invoice number of parent transition.	String(6)	No
8	IsSwipe	Set as FALSE	Boolean	Yes
9	Field0	Multiple Usage field	String	No
10	Field1	Multiple Usage field	String	No
11	Field2	Multiple Usage field	String	No

12	Field3	Multiple Usage field	String	No
13	BatchNo	If independent transaction, then it is not required. Else in case of dependent transaction, it is the Batch Id of parent transition.	Integer(9001-99999)	No
14	Roc	If independent transaction, then it is not required. Else in case of dependent transaction, it is the Roc of parent transition.	Integer(101-999)	No
15	TransactionLogId	If PG@POS transaction, then Mobile number where link will be sent. Else in case of dependent transaction, it is the Transaction log of parent transaction.	Long	No
16	RewardAmount	Amount to be paid by reward points amount or in cash in paise (or in lowest currency)	Long	No
17	CustomerMobileNumber	Customer mobile number if required to be captured. Can be used for sending SMS for charge slip. If there are more than one value pipe separated format can be used.	String(100)	No
18	CustomerEmailId	Customer email Id if required to be captured. Can be used for sending SMS for charge slip. If there are more than one value pipe separated format can be used.	String(500)	No
19	MerchantMobileNumber	Merchant mobile number if required to be captured. Can be used for Number(s) sending SMS for charge slip. If there are more than one value pipe separated format can be used.	String(100)	No
20	MerchantEmailId	Merchant email Id if required to be captured. Can be used for sending SMS for charge slip. If there are more than one value pipe separated format can be used.	String(500)	No
21	ConsentCustomerMobile	By default this is FALSE. If this parameter is set as TRUE, it is assumed that the merchant	Boolean	No

		has taken consent from customer for sending charge slip on his/her mobile number(s).		
22	ConsentCustomerEmailId	By default this is FALSE. If this parameter is set as TRUE, it is assumed that the merchant has taken consent from customer for sending charge slip on his/her email id(s).	Boolean	No
23	ConsentMerchantMobile	By default this is FALSE. If this parameter is set as TRUE, it is assumed that the merchant gives consent for sending charge slip on his/her mobile number.	Boolean	No
24	ConsentMerchantEmailId	By default this is FALSE. If this parameter is set as TRUE, it is assumed that the merchant gives consent for sending charge slip on his/her email id(s).	Boolean	No
25	WalletProgramId	This ID will be assigned by Pine labs to each wallet program type. While performing any Wallet transaction this field needs to set to identify wallet host.	Long	
26	CurrencyId	International Currency Code. Default is ("INR")	String(3)	No
27	Products	This tag will store list of products	Object List	No
	ProductId	Product Id or Code in Billing App	String(100)	
	ProductName	Product Name	String(100)	
	Quantity	Quantity of Product sold	Long	
	ListPrice	List Price of product in smallest unit	Long	
	Amount	Price of Product sold (List Price x Quantity) in smallest unit	Long	
	Discount	Discount of Product sold (List Price x Quantity) in smallest unit	Long	
	DiscountedAmount	Amount after discount, in smallest unit	Long	
	ProductInfo	Reserved for Future use	Array[]	No

	Key	Key Name	String(10)	
	Value	Value Text	String(100)	
28	AdditionalInfo	Reserved for Future use	Array[]	No
	Key	Key Name	String(10)	
	Value	Value Text	String(100)	

Sample JSON

For HDFC UPI Sale Transaction of amount of Rs 14.11

```
{
  "Detail": {
    "BillingRefNo": "155237781",
    "BankCode": 2, //HDFC UPI
    "PaymentAmount": 1411,
    "TransactionType": 5120
  },
  "Header": {
    "ApplicationId": "1001",
    "MethodId": "1001",
    "UserId": "UserId",
    "VersionNo": "1.0"
  }
}
```

For HDFC BQR Sale Transaction of amount of Rs 14.11

```
{
  "Detail": {
    "BillingRefNo": "155237781",
    "BankCode": 1, //HDFC BQR
    "PaymentAmount": 1411,
    "TransactionType": 5123 //BQR Sale
  },
  "Header": {
    "ApplicationId": "1001",
    "MethodId": "1001",
    "UserId": "UserId",
    "VersionNo": "1.0"
  }
}
```

For PhonePe Wallet Sale for Rs. 100

```
{
  "Detail": {
    "BillingRefNo": "TX12345678",
    "PaymentAmount": 10000,
    "BankCode": 105, //PhonePe
    "TransactionType": 5102
  },
  "Header": {
    "ApplicationId": "1001",

```

```
"MethodId": "1001",  
"UserId": "UserId",  
"VersionNo": "1.0"  
}  
}
```

For Freecharge Wallet Sale for Rs. 100

```
{  
  "Detail": {  
    "BillingRefNo": "TX12345678",  
    "PaymentAmount": 10000,  
    "BankCode": 103, //Freecharge  
    "TransactionType": 5102  
  },  
  "Header": {  
    "ApplicationId": "1001",  
    "MethodId": "1001",  
    "UserId": "UserId",  
    "VersionNo": "1.0"  
  }  
}
```

For Airtel Bank Sale for Rs. 100

```
{  
  "Detail": {  
    "BillingRefNo": "TX12345678",  
    "PaymentAmount": 10000,  
    "TransactionType": 5127  
  },  
  "Header": {  
    "ApplicationId": "1001",  
    "MethodId": "1001",  
    "UserId": "UserId",  
    "VersionNo": "1.0"  
  }  
}
```

PG @ POS SALE CHARGE REQUEST

Sample request strings:

If billing application just triggers the transaction using card

```
{  
  "Detail": {  
    "BillingRefNo": "1552377816904",  
    "PaymentAmount": 100,  
    "TransactionType": 5561  
  }  
}
```

```
"BankCode" : 1,  
"TransactionLogId": 9999888809    // Optional  
,  
"Header": {  
  "ApplicationId": "abcdefgh",  
  "MethodId": "1001",  
  "UserId": "UserId",  
  "VersionNo": "1.0"  
}  
}
```

If billing application just triggers the transaction for Bank EMI

```
{  
  "Detail": {  
    "BillingRefNo": "1552377816904",  
    "PaymentAmount": 100,  
    "TransactionType": 5566  
    "BankCode" : 1,  
    "TransactionLogId": 9999888809    // Optional  
  },  
  "Header": {  
    "ApplicationId": "abcdefgh",  
    "MethodId": "1001",  
    "UserId": "UserId",  
    "VersionNo": "1.0"  
  }  
}
```

If billing application just triggers the transaction for Brand EMI

```
{  
  "Detail": {  
    "BillingRefNo": "1552377816904",  
    "PaymentAmount": 100,  
    "TransactionType": 5567  
    "BankCode" : 1,  
    "TransactionLogId": 9999888809    // Optional  
  },  
  "Header": {  
    "ApplicationId": "abcdefgh",  
    "MethodId": "1001",  
    "UserId": "UserId",  
    "VersionNo": "1.0"  
  }  
}
```

Response:

Tag Name	Description	Data Type
Payments	Array of payments object	Object Array
BillingRefNo	Transaction reference number from external application. EPOS will only use this value for printing on chargeslip.	String(10)
ApprovalCode	Authorization code, if transaction was approved. Otherwise empty string. Presence of non-zero length approval code string indicates successful authorization of transaction.	String(6)
HostResponse	Response string if a response for transaction was received from bank switch. Otherwise, if any error occurs before response is received, this is an empty string.	String(50)
CardNumber	Mobile Number / Empty string.	String(19)
ExpiryDate	Empty String	String(4)
CardholderName	Empty string	String(25)
CardType	Empty string	String(12)
InvoiceNumber	EFT transaction invoice number, if transaction authorized. Otherwise, 0/EDC ROC (the same is printed on chargeslip)	Long
BatchNumber	EFT transaction batch number, if transaction authorized. Otherwise, 0/EDC Batch ID (in case of Reward transaction)	Long
TerminalId	EFT TID, if transaction authorized. Otherwise, empty string	String(8)
LoyaltyPointsAwarded	Loyalty point awarded, if any.	Long
Remark	Description of error, if an error occurs. Otherwise, empty string. An empty string in this field DOES NOT imply successful transaction authorization	String(100)
AcquirerName	Name of acquirer to which transaction was routed. E.g. "ICICI BANK"	String(48)
MerchantId	EFT ME ID, if transaction authorized. Otherwise, empty string	String(15)
RetrievalReferenceNumber	EFT RRN, if transaction authorized. Otherwise, empty string	String(12)
CardEntryMode	Not applicable	Integer
PrintCardholderName	Not applicable	Integer
MerchantName	Merchant name, if transaction authorized. Otherwise, empty string	String(23)
MerchantAddress	Merchant address line, if transaction authorized. Otherwise, empty string	String(23)

MerchantCity	Merchant city line, if transaction authorized. Otherwise, empty string	String(23)
PlutusVersion	App Version	String(40)
AcquiringBankCode	Code for bank used for processing transaction	Integer
RewardRedeemedAmount	Redeemed Amount in Paise	Long
RewardRedeemedPoints	Redeemed Points	Double
RewardBalanceAmount	Balance Amount	String(10)
RewardBalancePoints	Balance Points	Double
Field0	Multiple Usage Field	String
CouponCode	Processing Fee in Rs. (decimal) Or Coupon Code. Coupon code is the value which will be coming as a response to voucher redemption. This field will be present in case of voucher redemption. OR Loyalty number fetched if the transaction type is 4301.	String(23)
AmountProcessed	Amount will be in paise or lowest currency.	String(99)
Field3	Multiple Usage Field	String
Field4	Multiple Usage Field	String
TransactionDate	Date of the Transaction as per acquiring host. Date to be printed on chargeslip. In MMDDYYYY Format.	String(8)
TransactionTime	Time of the Transaction as per acquiring host. Time to be printed on chargeslip. HHMMSS where HH in 24 hour format.	String(6)
PineLabsClientId	Unique ID assigned to Pine Labs EDC.	Integer
PineLabsBatchId	Batch ID of Pine Labs EDC	Integer
PineLabsRoc	ROC of Pine Labs EDC	Integer
AdditionalInfo	Reserved for Future use	Array[]
Key	Key Name	String(10)
Value	Value Text	String(100)

Sample JSON

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1001",
    "VersionNo": "1.0"
  },
  "Response": {
    "ResponseCode": "00",
    "ResponseMsg": "Success"
  }
}
```

```
},
"Detail": {
  "Payments": [
    {
      "BillingRefNo": "105",
      "ApprovalCode": "7261A9",
      "HostResponse": "APPROVED",
      "CardNumber": "438624*****2802",
      "ExpiryDate": "0406",
      "CardholderName": "AMITMOHAN",
      "CardType": "UPI",
      "InvoiceNumber": 11,
      "BatchNumber": 2,
      "TerminalId": "30000001",
      "LoyaltyPointsAwarded": 1,
      "Remark": "PROCESSED",
      "AcquirerName": "AXIS UPI",
      "MerchantId": "000100090015607",
      "RetrievalReferenceNumber": "624615343002",
      "CardEntryMode": 1,
      "PrintCardholderName": 1,
      "MerchantName": "HPCL Area 18",
      "MerchantAddress": "Kamala Mills",
      "MerchantCity": "Noida",
      "PlutusVersion": "1.51 ICICI BANK",
      "AcquiringBankCode": 2,
      "TransactionDate": "02012011",
      "TransactionTime": "210403",
      "PineLabsClientId": 12345,
      "PineLabsBatchId": 9002,
      "PineLabsRoc": 105
    }
  ]
}
```

Sample Response for PG@POS

Sample Response String:

There will be 2 cases:

CASE 1: Initiated Sale charge Txn

```
{
  "Response": {
    "AppVersion": "109",
    "ResponseMsg": "TRANSACTION INITIATED, CHECK GET STATUS",
    "ResponseCode": 0
  },
  "Detail": {
    "InvoiceNumber": 118,
    "MerchantName": "Tatastarbucks Test",
    "CardType": "WALLET",
```



```

"ExpiryDate": " ",
"PlutusVersion": "v1.0.0",
"AcquiringBankCode": 103,
"AmountProcessed": "200",
"ApprovalCode": "",
"TerminalId": "20001397",
"TransactionDate": "03042019",
"PineLabsBatchId": 200,
"PineLabsClientId": 0,
"CardNumber": "*****3310",
"TransactionTime": "115634",
"HostResponse": "TRANSACTION INITIATED, CHECK GET STATUS",
"RetrievalReferenceNumber": "",
"MerchantId": "1",
"PineLabsRoc": 5102,
"BillingRefNo": "TX12345678",
"AcquirerName": "PG@POS",
"MerchantAddress": "Pine Labs Pvt Ltd",
"MerchantCity": "DIBRUGARH",
"BatchNumber": 9066,
"PrintCardholderName": 0
},
"Header": {
  "UserId": "UserId",
  "ApplicationId": "1001",
  "VersionNo": "1.0",
  "MethodId": "1001"
}
}

```

CASE 2: Sale charge is complete

```

"1245","005436","APPROVED ","*****6400","
","","WALLET","231","9102","ABCD123466","","","PG@POS","M2306160483220675579140","","","0
","varunstore","jaipur","ADILABAD","v1.0.0","105","","","","","","2500","","","04112017","161835
"

```

```

{
  "Response": {
    "AppVersion": "109",
    "ResponseMsg": "APPROVED",
    "ResponseCode": 0
  },
  "Detail": {
    "InvoiceNumber": 118,
    "MerchantName": "Tatastarbucks Test",
    "CardType": "WALLET",
    "ExpiryDate": " ",
    "PlutusVersion": "v1.0.0",
    "AcquiringBankCode": 103,

```

```
"AmountProcessed": "200",
"ApprovalCode": "005436",
"TerminalId": "20001397",
"TransactionDate": "03042019",
"PineLabsBatchId": 200,
"PineLabsClientId": 0,
"CardNumber": "*****3310",
"TransactionTime": "115634",
"HostResponse": "APPROVED",
"RetrievalReferenceNumber": "20001397000010",
"MerchantId": "1",
"PineLabsRoc": 5102,
"BillingRefNo": "TX12345678",
"AcquirerName": "PG@POS",
"MerchantAddress": "Pine Labs Pvt Ltd",
"MerchantCity": "DIBRUGARH",
"BatchNumber": 9066,
"PrintCardholderName": 0
},
"Header": {
  "UserId": "UserId",
  "ApplicationId": "1001",
  "VersionNo": "1.0",
  "MethodId": "1001"
}
}
```

7. API integration Process for Billing App

Messenger usage flow:

1. **EPOS** will host a service that will implement a *Handler* for receiving call-back from Billing App.
2. This handler will create a *Messenger* object which further creates an *IBinder* object which EPOS service returns to Billing App.
3. Billing App will use the *IBinder* object to create a *Messenger* object to send *Messages*.
4. The service running in **EPOS** will receive each *Message* in JSON string format in its *Handler* and corresponding API action is performed.
5. After processing the API request, the service will respond back in JSON string format to Billing App using *Messenger*.

Sample Code for calling EPOS API from Billing App

1. Billing App will bind to the **EPOS** service Handler

```
Intent intent = new Intent();
intent.setAction(EPOS_ACTION);

intent.setPackage(EPOS_PACKAGE);

bindService(intent, connection, BIND_AUTO_CREATE);
```

2. After successful binding, the Service will respond to the *ServiceConnection* by returning to *onServiceConnected()*. A new messenger will be created using returned *IBinder*.

```
private ServiceConnection connection = new ServiceConnection() {

    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {

        mServerMessenger = new Messenger(service);

        isBound = true;
    }

    @Override
    public void onServiceDisconnected(ComponentName name) {

        mServerMessenger = null;

        isBound = false;
    }
}
```

3. A message will be created and sent using the above *mServerMessenger*. This message will contain the API request information.

```

Message message = Message.obtain(null, MESSAGE_CODE);

Bundle data = new Bundle();

String value = { "Header": { "ApplicationId": "abcdefgh", "UserId":
"user1234", "MethodId": "1004", "VersionNo": "1.0"} }"; // sample json request

data.putString(BILLING_REQUEST_TAG, value);

message.setData(data);

try {

    message.replyTo = new Messenger(new IncomingHandler());

    mServerMessenger.send(message);

} catch (RemoteException e) {

    e.printStackTrace();

}

```

4. On receiving the response back from Service, Billing App will process the response in *IncomingHandler*.

```

private class IncomingHandler extends Handler {

    @Override
    public void handleMessage(Message msg) {

        Bundle bundle = msg.getData();

        String value = bundle.getString(BILLING_RESPONSE_TAG);
        // process the response Json as required.

    }

}

```

List of Constants:

Name	Value
EPOS_PACKAGE	com.pinelabs.epos
EPOS_ACTION	com.pinelabs.epos.SERVER
MESSAGE_CODE	1001
BILLING_REQUEST_TAG	MASTERAPPREQUEST
BILLING_RESPONSE_TAG	MASTERAPPRESPONSE

8. Glossary

8.1 Response Codes

For all API successful responses, Response Code will be set to zero.

Code	Message
1	App not activated
2	Already activated
3	Invalid Method Id
4	Invalid User/Pin
5	User blocked for max attempt
6	Permission denied for this user
7	Invalid data-format

More error-codes will be added as per specific scenarios.

8.2 Method Ids

Code	Method Name
1001	Do Transaction

8.3 Transaction Types

	Transaction Description	Transaction Type Value
1.	Wallet Sale	5102
2.	Wallet Load	5103
3.	Wallet Void	5104
4.	UPI Sale	5120
5.	UPI / BQR Void	5121
6.	UPI / BQR Get Status	5122
7.	Bharat QR Sale	5123
8.	Airtel Bank Sale	5127
9.	PG@POS Bank EMI Sale	5566
10.	PG@POS Brand EMI Sale	5567
11.	PG@POS Credit/Debit Card Sale	5561
12.	PG@POS Void	5562
13.	PG@POS Get Status	5563
14.	PG@POS Resend SMS	5564

8.4 Wallet IDs – Bank Code

	Wallet Name	Bank Code - ID
1.	FreeCharge	103
2.	Phone Pe	105

8.5 UPI / BQR IDs – Bank Code

	Host Name	Bank Code - ID
--	-----------	----------------

1.	HDFC Bharat QR	1
2.	HDFC UPI	2

9. Security Requirement

Following minimum requirements should be ensured for all existing or new Third party service providers:

A. Due diligence should be performed on vendor financial stability before an agreement is entered with Third party service provider. Pine Labs and Third party may enter in escrow agreement in case of any issues

B. Non-disclosure agreement (NDA) to be executed between Pine Labs and Third party service provider

Pine Labs defined NDA shall be executed with Third party service provider but not limited to liabilities, security and confidentiality requirements

C. Application Security Assessment document:

a. Application Secure Code Review Report: Application secure code review report shall be obtained from vendor which would have been tested against OWASP and SANS and the same shall be shared with Information Security group before testing

b. Application Pentest Report: Application penetration testing report and assessment report should be obtained from Third party service provider which is tested against OWASP & SANS requirement. The same shall be shared with Information Security group before testing

D. Business should review change management process in place with Third party service provider. Business should review processes in place by Third party service provider to periodically share patches for applications to minimize risks to Pine Labs

F. Following technical controls should be evaluated by Business such as:

- a. Application interface authentication
- b. Data masking techniques
- c. Rest full API's

G. Security Assessment Ownership and Requisite

Security assessment on Third party application will be under ownership of Pine Labs information security team. QA team shall provide following details for Security Assessment:

#	Question	Response
1	Company Name	3rd Party Name
2	Application Details	Complete use case of Application
3	Request and Response Parameter	
4	Application Login Account username and password	

H. Security Assessment by Pine Labs

Post Compliance to above mentioned requirements from A to F, Information security group at Pine Labs will conduct an assessment on the application to identify security-related weaknesses in the application as per OWASP & SANS guidelines.

I. Internal Approval

Pine Labs Security team also requires approval from Business Head and Technology Head before initiating any signing or security Assessment of application

10. Pine Labs validation testing process

- Module Lead assign testing to QA team for 3 rounds of Independent Individual Validation Testing (IIVT).
- IIVT round can take more than 2 weeks for this session. It is dependent on successful execution of all the test cases.
- After executing all the test cases independently by the team, they would share their observations & feedback to Pine Labs Module Lead/Manager.
- On the basis of these observations submitted by the team, Module Lead/Manager would then decide to submit the same for further process of Validation Testing.
- If all the test cases have been executed successfully, then it goes to the final round for next course of action.
- Pine Labs team would test the basic flow of billing application and execute all the test cases thoroughly post which they would submit their observation to the Pine Labs Module Lead/Manager and the Merchant
- After successful completion of this round, team will submit their final observation and provide end to end training to our Operation Team and Customer Care for further support.
- After this entire activity has been completed, we would sign off for Pilot only on a few identified Billing POS for 2 weeks. Merchant will share the process note of the complete billing application flow.
- Final exercise before we roll out: Risk Management team will continuously monitor and follow up with the merchant and monitor the transactions status and provide feedback during Pilot and submit their report for final roll out.