Pine Labs

PLUTUS SMART/APOS INTEGRATION GUIDE

Version – 2.1

Contents

Disclaimer & Legal Notice

Every care has been taken in the preparation of this document to ensure that the information provided is accurate, factual and correct to the best of our knowledge. Pine Labs accepts no liability for any loss or damage or unforeseen consequential loss or damage arising from the use of the information contained within this document.

Revision History

Version	Modified	Description	Date
1.0	Pine Labs	Initial document creation	26-Dec-18
1.0.1	Pine Labs	Updated Constants in Section 8	05-Mar-19
1.0.2	Pine Labs	Updated Security Requirements	04-Jun19
1.0.3	Pine Labs	Added Scan Code API	23-Jul-19
1.0.4	Pine Labs	Added Upload Invoice API	05-Aug-19
1.0.5	Pine Labs	Addition of Payer Name in Request	25-Feb-20
2.0	Pine Labs	Addition of Billing App API Support for Hybrid Applications	04-May-20
2.1	Pine Labs	Removal of Payer Name in Request	05-May-20

1. Introduction

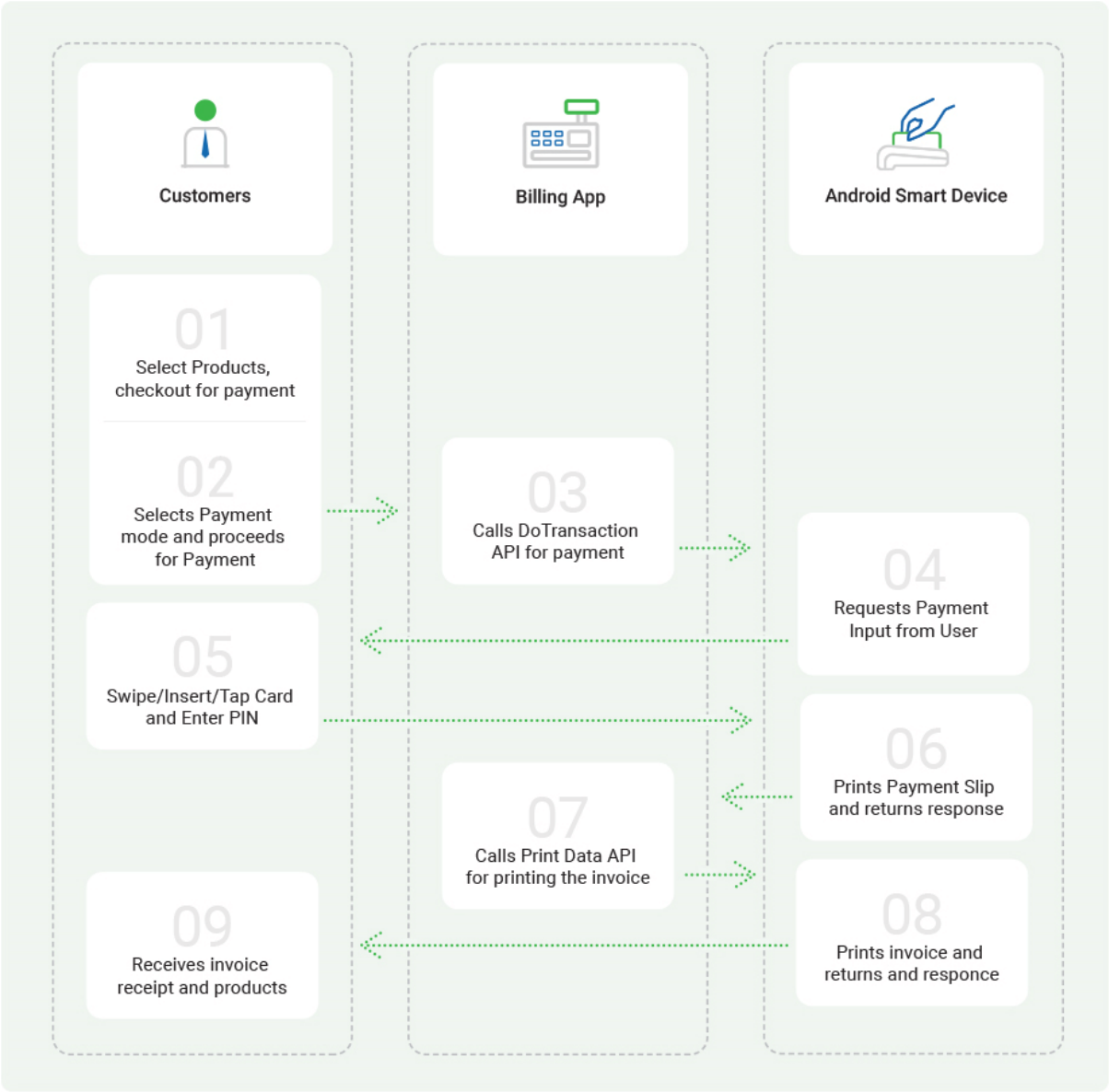
Plutus Smart or APOS (Android POS) is an Android-based smart transaction machine with ergonomic design, fast transaction speed and intuitive interface. Integrate it with your existing billing system for faster transaction speed.

Plutus Smart Integration allows you to access the following features:

- Carry-out sale transactions using multiple payment modes - credit card, debit card, wallet etc.
- Printing service to carry out bill, coupon, promotion printing using device printer
- Batch settlement
- View/get terminal profile info

2. Sample Use Case

- User selects products and checkout for payment.
- User selects payment mode and proceeds for payment.
- Billing App calls **DoTransaction** API with payment-amount.
- Plutus Smart processes the payment and prints chargeslip.
- On receiving success response, Billing App calls **PrintData** API with invoice details.
- Plutus Smart prints the invoice and returns response.
- User receives invoice receipt.



3. Inter-application Communication

Billing application will communicate with Plutus Smart APIs for transactional and other Plutus-enabled features. 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).

4. Getting Started On Developer Portal

To integrate your application(s) with the Plutus Smart API on the Android Smart device, following steps need to be followed:

1. Firstly, you need to create a developer account with Pine Labs using Signup option on https://developer.pinelabs.com/ .
2. Once signed-up, sign-in to your developer account to proceed.
3. After signing-in, you need to create a billing application profile.
4. Once the billing application profile is created, you can now proceed with the code integration and use test key for unit testing with the simulator application.
5. Unit-testing using Simulator includes the below mentioned steps:
    a. First, you need to add your mobile device by using the Add device option, ignore if already done.
    b. You need to map the device with the billing application.
    c. Next, you need to download the simulator application available on the GitHub link .
    d. Now, install the simulator application on your mobile and activate it. Post activation, the application is all set for testing.
    e. To proceed further, you will need to add the test-key in your billing application, provided for application testing available on the portal, to initiate transaction request. This test-key is unique for every application.
    f. For reference, you can download a sample billing application along with source code available using this GitHub link .
    g. During unit-testing phase, you can also download and execute sample test cases using this GitHub link .
6. Post development and unit-testing, you need to go for Certification on the Simulator App.
7. You need to create test plan for the billing application and run test steps to complete all the test cases under this test plan as guided on the portal.
8. Once you run the test cases, the logs will be uploaded through Simulator App for validation, and execution will be verified.
9. Post completion of test cases execution, you will get an option to proceed for the application Certification.
10. You will be given Production key for the application and you will be required to upload application APK with this key and other details.
11. Your application will be reviewed by Pine Labs' team and you will be informed about the application status by emails.

5. API Integration Process

5.1 API Integration Process for Native App's

Messenger usage flow:

1. **Plutus Smart** 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 Plutus Smart service returns to Billing App.
3. Billing App will use the *IBinder* object to create a *Messenger* object to send *Message*s.
4. The service running in **Plutus Smart** 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 Plutus Smart API from Billing App

1. Billing App will bind to the **Plutus Smart** service Handler

```
Intent intent = new Intent();
intent.setAction(PLUTUS_SMART_ACTION);
intent.setPackage(PLUTUS_SMART_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
PLUTUS_SMART_PACKAGE	com.pinelabs.masterapp
PLUTUS_SMART_ACTION	com.pinelabs.masterapp.SERVER
MESSAGE_CODE	1001
BILLING _REQUEST_TAG	MASTERAPPREQUEST
BILLING_RESPONSE_TAG	MASTERAPPRESPONSE

5.2 API Integration Process for Hybrid App's

Plutus Smart now supports communication with apps built on hybrid platform. For this communication to happen it will use Intents. The hybrid framework used to build the billing application can provide this functionality natively or can be achieved via help of a third-party plugin.

In this process, the hybrid application defines an intent with action as "com.pinelabs.masterapp.HYBRID_REQUEST" and the request JSON as extra payload. When adding the payload as extra, the client has to use the key as "REQUEST_DATA" and then use

startActivityForResult() method to launch the intent. This intent is received by the server which internally binds to the service and process the request send by the client. The result is returned via the same calling intent, the client has to define onActivityResult() to receive the result from the server and use “RESPONSE_DATA” as the key to extract data from the intent.

Example of Intent object for DoTransactionRequest in a Hybrid Application:

```
String doTransactionPayload =
"{\"Detail\"{\"BillingRefNo\":\"TX12345678\",\"PaymentAmount\":9900,\"TransactionType\":4001},
\"Header\"{\"ApplicationId\":\"1001\",\"MethodId\":\"1001\",\"UserId\":\"userId\",\"VersionNo\":\
"1.0\"}}";

val intent = Intent(“com.pinelabs.masterapp.HYBRID_REQUEST”);
intent.putExtra(“REQUEST_DATA”,doTransactionPayload);
startActivityForResult(intent);
```

Note: The Request and the Response parameters will remain the same as in case of a native request the only change is instead of using a messenger, we are passing data via extra in an Intent object.

6. 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 Plutus System	String (100)	Yes
UserId	Billing app user-Id/name	String (100)	No
MethodId	Unique Method Id	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"
  }
}
```

7. 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 Plutus System	String (100)
UserId	Billing app user-Id/name	String (100)
MethodId	Unique Method Id	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)

Sample Request

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

}
```

8. API Details

8.1 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 Transaction Types for complete list of transactions supported.

8.1.1 Request Parameter

S. No.	Tag Name	Description	Data Type	Is Mandatory
1	TransactionType	The type of payment transaction to be processed by Plutus Smart. Refer Transaction Types for all possible values.	Long	Yes
2	BillingRefNo	Transaction reference number from external application. Plutus will use this value for printing on charge slip.	String (10)	No
3	PaymentAmount	Amount to be charged to card – expressed as a whole number in lowest currency unit (i.e. in paise)	Long	Yes
4	BankCode	The acquirer bank code to which transaction will be routed. Optional in case of sale transaction if Automatic Acquirer Selection is chosen	String (2)	No
5	CardNumber	Track 1 data of the card or Card number if manual entry. If empty, then App will ask for card input.	String (76)	No

6	Expiry	Track 2 data of the card or Expiry date if manual entry. If empty, then Plutus will ask for card input. Expiry date is in YYMM format. In case of Pine 360, if Track 1 consists of mobile or GV number, this field will indicate the card entry mode. 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	Specifies if Swipe needs to be disabled on Plutus. By default, it is TRUE.	Boolean	Yes
9	Field0	Multiple Usage field	String	No
10	Field1	Multiple Usage field	String	No
11	Field2	Multiple Usage field	String	No
12	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
13	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
14	TransactionLogId	If independent transaction, then it is not required. Else in case of dependent transaction, it is the	Long	No

		Transaction log of parent transaction.		
15	RewardAmount	Amount to be paid by reward points amount or in cash in paise (or in lowest currency)	Long	No
16	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
17	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
18	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
19	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
20	ConsentCustomerMobile	By default, this is FALSE. If this parameter is set as TRUE, it is assumed that the merchant has taken consent	Boolean	No

		from customer for sending charge slip on his/her mobile number(s).		
21	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
22	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
23	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
24	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	
25	AdditionalInfo	Map of key value pairs to capture additional transactional data. MaxLength allowed is 10 elements. Allowed for sale transaction type (4001) only.	HashMap<String,String>	No

	Key	Key Name	String (25)	
	Value	Value Text	String (25)	
26	MobileNumberForEChargeSlip	Mobile Number for printing E-chargeslip	String (10)	

Sample JSON Request

8.1.1.1 Card Sale

For Sale Transaction of amount of Rs 99990.00 with Additional Info Details

```
{
  "Header": {
          "ApplicationId": "abcdefgh",
          "UserId": "user1234",
          "MethodId": "1001",
          "VersionNo": "1.0"
  },
  "Detail": {
          "TransactionType": "4001",
          "BillingRefNo": "TXN12345678",
          "PaymentAmount": "9999000",
          "MobileNumberForEChargeSlip": "9876543210",
          "AdditionalInfo": {
                  "Split1": "99991",
                  "Split2": "99992",
                  "Split3": "99993"
          }
  }
}
```

8.1.1.2 Void

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1001",
        "VersionNo": "1.0"
    },
    "Detail": {
        "TransactionType": "4006",
        "BillingRefNo": "TXN12345678",
        "PaymentAmount": "9999000",
        "BankCode": "01",
        "InvoiceNo": "000012",
    }
}
```

8.1.2 Response Parameter

Tag Name	Description	Data Type
Payments	**Array of payments object**	**Object Array**
BillingRefNo	Transaction reference number from external application. Plutus will only use this value for printing on charge slip.	String (10)
ApprovalCode	Credit card authorization code if transaction was approved. Otherwise empty string. **Presence of non-zero length approval code string indicates successful authorization of transaction. This logic holds true for Pine 360 transactions as well**	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	Card number will be present if card was swiped. Otherwise, empty string.	String (19)
ExpiryDate	Card expiration date, expressed in format YYMM, if valid card was swiped. Otherwise, empty string. Some acquirers mandate Expiry date to be masked, in that case a value of "XXXX" will be returned.	String (4)
CardholderName	Cardholder's name from card track 1, if valid card was swiped and card holder name present on Track 1. Otherwise, empty string	String (25)
CardType	Card association name if valid card was swiped. Otherwise, empty string. E.g. "VISA"	String (12)
InvoiceNumber	EFT transaction invoice number if transaction authorized. Otherwise, 0/EDC ROC (the same is printed on charge slip)	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	Enumeration of Card Entry modes: 0 – Manual entry 1 – Swipe entry 2 – Chip card entry Any other value – card not validated	Integer
PrintCardholderName	This is used if external application is to print Plutus chargeslip. 0 – Do not print cardholder's name 1 – Print cardholder's name Any other value – card not validated.	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	Plutus Version	String (40)
AcquiringBankCode	Code for bank used for processing transaction. Enumeration of possible values: 01 – HDFC BANK 02 – ICICI BANK 03 – AMERICAN EXPRESS 04 – CITIBANK 05 – AXIS BANK 06 – SBI 07 – HSBC 09 – CORP BANK 10 – CUB (City Union Bank) 14 – IDBI Bank 17 – LVB (Lakshmi Vilas Bank) 51- PINE 360 81 – Loyalty Reward 82 – Aimia	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	Card Processing Fee in Rs. (decimal) Or Coupon Code. Coupon code is the value which will be coming as a response to voucher	String (23)

	redemption. This field will be present in case of voucher redemption. OR Loyalty number fetched if the transaction type is 4301.	
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 charge slip. In MMDDYYYY Format.	String (8)
TransactionTime	Time of the Transaction as per acquiring host. Time to be printed on charge slip. 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 Response


```
{
    "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": "VISA",
          "InvoiceNumber": 11,
          "BatchNumber": 2,
          "TerminalId": "30000001",
          "LoyaltyPointsAwarded": 1,
          "Remark": "PROCESSED",
          "AcquirerName": "Acquiring Bank 1",
          "MerchantId": "000100090015607",
          "RetrievalReferenceNumber": "624615343002",
```

```
            "CardEntryMode": 1,
            "PrintCardholderName": 1,
            "MerchantName": "HPCL Area 18",
            "MerchantAddress": "Kamala Mills",
            "MerchantCity": "Noida",
            "PlutusVersion": "1.51 ICICI BANK",
            "AquiringBankCode": 2,
            "TransactionDate": "02012011",
            "TransactionTime": "210403",
            "PineLabsClientId": 12345,
            "PineLabsBatchId": 9002,
            "PineLabsRoc": 105
          }]
    }
}
```


8.2 Print Data

This API will be called when Billing App wants to print paper-receipt on Plutus Smart Device.

8.2.1 Request Parameter

Tag Name	Description	Type	Is Mandatory
PrintRefNo	Unique reference number from Billing App	String (10)	Yes
SavePrintData	Set this parameter to save the Print Data at Plutus Smart Device. Default value is TRUE	Boolean	Yes
Data	Array of print lines	Array []	Yes
PrintDataType	Data Type will be as following PrintText =0 PrintImageByPath =1 PrintImageDump =2 PrintBarcode=3 PrintQRCode=4	Integer	Yes
PrinterWidth	Line Width of Printer, Possible values: 24,32,48	Integer	Yes
IsCenterAligned	It will contain true or false for data to be printed in center-aligned or not	Boolean	Yes
DataToPrint	It contains data to print in form of String.	String	No
ImagePath	It contains image path from Device **external** storage	String	No
ImageData	It contains image data in form of encoded string	String	No
PrintDataInfo	Reserved for Future use	Array []	No
Key	Key Name	String (10)	
Value	Value Text	String (100)	

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

Sample JSON Request

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1002",
    "VersionNo": "1.0"
  },
  "Detail": {
    "PrintRefNo": "123456789",
    "SavePrintData": true,
    "Data": [
      {
        "PrintDataType": "0",
        "PrinterWidth": 24,
        "IsCenterAligned": true,
        "DataToPrint": "String Data",
        "ImagePath": "0",
        "ImageData": "0"
      },
      {
        "PrintDataType": "1",
        "PrinterWidth": 24,
        "IsCenterAligned": true,
        "DataToPrint": "",
        "ImagePath": "Image Path",
        "ImageData": "0"
      },
      {
        "PrintDataType": "2",
        "PrinterWidth": 24,
        "IsCenterAligned": true,
        "DataToPrint": "",
        "ImagePath": "",
        "ImageData": "Image Data String"
      },
      {
        "PrintDataType": "3",
        "PrinterWidth": 24,
        "IsCenterAligned": true,
        "DataToPrint": "Bar Code Data in String",
        "ImagePath": "",
        "ImageData": ""
      },
      {
        "PrintDataType": "4",
        "PrinterWidth": 24,
        "IsCenterAligned": true,
```

```
        "DataToPrint": "QR Code Data in String",
        "ImagePath": "",
        "ImageData": ""
      }
    ]
  }
}
```

8.2.2 Response Parameter

Tag Name	Description	Type
ResponseCode	Response codes for printer response: PRINTER_SUCCESS = 0; PRINTER_FAILED = 1; PRINTER_BUSY = 1001; PRINTER_OUT_OF_PAPER = 1002; PRINTER_LOW_PAPER = 1003; PRINTER_LOW_BATTERY = 1004; PRINTER_HARDWARE_ERROR = 1005; PRINTER_OVERHEAT = 1006; PRINTER_BUFFER_OVERFLOW = 1007; PRINTER_PAPER_ALIGN_POSITION = 1008; PRINTER_PAPER_JAM = 1009; PRINTER_CUT_POSITION_ERROR = 1010;	Integer
ResponseMessage	Response message for Printer response: SUCCESS FAILED PRINTER IS BUSY PRINTERIS OUT OF PAPER PRINTER HAS LOW PAPER PRINTER_LOW_BATTERY PRINTER HARDWARE ISSUE PRINTER IS OVERHEAT PRINTER BUFFER OVERFLOW PAPER IS NOT ALIGNED PROPERLY PAPER STUCKED PAPER CUT KNIFE IS NOT IN ORIGINAL PLACE	String
AppVersion	Peripheral App Version	String
ParameterJson	Additional parameters to be sent	String

Sample JSON Response

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
```

```
        "UserId": "user1234",
        "MethodId": "1002",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "00",
         "ResponseMsg": "Success"
    },
    "Detail": {
        "AppVersion": "Plutus v1.5"
    }
}
```


8.3 Settlement

There are two ways to settle the current batch of payment transactions:

- Settlement API can be used to settle current batch in Plutus Smart App. On calling this API and successful response, charge slip will be printed on the terminal.
- User can go to Plutus Smart App menu to manually settle the batch.

8.3.1 Request Parameter

Sample JSON Request


```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1003",
        "VersionNo": "1.0"
    }
}
```


8.3.2 Response Parameter

Tag Name	Description	Data Type
SettlementSummary	Settlement Summary Data in List Format	Array []
BatchName	Batch name	String
AcquirerCode	Acquirer Code	String
TID	Terminal Identifier	String
MID	Merchant Identifier	String
CreditCount	Count of Credit transactions in batch	Long
CreditAmount	Total Credit Amount in smallest unit	Long
DebitCount	Count of Debit transactions in batch	Long
DebitAmount	Total Debit Amount in smallest unit	Long
SettlementInfo	**Reserved for future use**	**Array []**
Key	Key Name	String (10)

Value	Value Text	String (100)
AdditionalInfo	Reserved for Future use	Array []
Key	Key Name	String (10)
Value	Value Text	String (100)

Sample JSON Response


```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1003",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "00",
         "ResponseMsg": "Success"
    },
    "Detail": {
        "SettlementSummary": [
         {
                "BatchName":  "HDFC",
                "AcquirerCode": "01",
                "TID": "01000234",
                "MID": "123411234",
                "CreditCount": 10,
                "CreditAmount": 502100,
                "DebitCount": 5,
                "DebitAmount": 324000
         },
         {
                "BatchName":  "ICICI",
                "AcquirerCode": "02",
                "TID": "013000123",
                "MID": "123411224",
                "CreditCount": 1,
                "CreditAmount": 2100,
                "DebitCount": 0,
                "DebitAmount": 324000
         }
       ]
    }
}
```


8.4 Get Terminal Info

This API will be called when the Billing App wants to get terminal details configured on Plutus Smart Device. It is an optional API, can be used to fetch and display store information on Billing App.

8.4.1 Request Parameter

Sample JSON Request

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1004",
        "VersionNo": "1.0"
    }
}
```

8.4.2 Response Parameter

Tag Name	Description	Type
PlutusStoreId	Plutus Store Identifier	String (50)
PlutusTerminalId	Plutus Terminal ID / Client ID / POS ID	String (50)
MerchantName	Merchant Name	String (100)
StoreName	Store Name	String (100)
AdditionalInfo	This Array will hold additional information	Object []
Key	Tag name	String (10)
Value	Tag Value	String (100)

Sample JSON Response

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1004",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "0",
         "ResponseMsg": "Success"
    },
    "Detail": {
        "PlutusStoreId": "19345",
        "PlutusTerminalId": "4523900",
        "MerchantName": "Payment India",
        "StoreName": "Delhi Store"
    }
}
```

8.5 Connect Bluetooth

This API will be called when the Billing App wants to Connect Bluetooth on Plutus Smart Device.

8.5.1 Request Parameter

Tag Name	Description	Type	Is Mandatory

BaseSerialNumber	Base Serial Number is defined at the back side of the base	String	Yes

Sample JSON Request

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1005",
    "VersionNo": "1.0"
  },
  "Detail": {
    "BaseSerialNumber": "121234"
  }
}
```

8.5.2 Response Parameter

Tag Name	Description	Type
ResponseCode	BLUETOOTH_CONNECTION_SUCCESS = 0; BLUETOOTH_CONNECTION _FAILED = 1; DEVICE_ALREADY_CONNECTED = 2; BLUETOOTH_IS_OFF = 3; INVALID_BASE_SERIAL_NUMBER = 4;	Integer
ResponseMessage	Bluetooth Connection Success Bluetooth Connection Failed Device Already connected with base over Bluetooth Bluetooth is disable Please enable Bluetooth Base serial number is invalid	String
AppVersion	Peripheral App Version	String
ParameterJson	Additional parameters to be sent	String

Sample JSON Response

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1005",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "0",
         "ResponseMsg": "Success"
    }
}
```

8.6 Disconnect Bluetooth

This API will be called when the Billing App wants to Disconnect Bluetooth on Plutus Smart Device.

8.6.1 Request Parameter

Sample JSON Request

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1006",
    "VersionNo": "1.0"
  }
}
```

8.6.2 Response Parameter

Tag Name	Description	Type
ResponseCode	BLUETOOTH_CONNECTION_SUCCESS = 0;	Integer
ResponseMessage	Bluetooth Disconnection Success	String
AppVersion	Peripheral App Version	String
ParameterJson	Additional parameters to be sent	String

Sample JSON Response

```
{
    "Header": {
        "ApplicationId": "abcdefgh",
        "UserId": "user1234",
        "MethodId": "1006",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "0",
         "ResponseMsg": "Bluetooth Disconnection Success"
    }
}
```

8.7 Scan QR Code/Barcode

This API will be called when the Billing App wants to scan any QR/Barcode on Plutus Smart Device.

8.7.1 Request Parameter

Sample JSON Request

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
```

```
    "MethodId": "1007",
    "VersionNo": "1.0"
  }
}
```

8.7.2 Response Parameter

Tag Name	Description	Type
ResponseCode	SUCCESS = 0; DEVICE_UNCONNECTED = 5; SCANNER_NOT_FOUND = 6; SCANNER_DATA_RECEIVE_SUCCESS = 10; SCANNER_DATA_RECEIVE_FAILED = 11; APPLICATION_BUSY = 2001; UNKNOWN_ERROR = 2099;	Integer
ResponseMessage	Success Device is not connected with base or USB device not found Scanner not found Data Scanned Success Data Scanned Failed Application is busy Unknown Error	String
AppVersion	Peripheral App Version	String
ParameterJson	Additional parameters to be sent	String

Sample JSON Response

```
{
  "Header": {
    "ApplicationId": "abcdefgh",
    "UserId": "user1234",
    "MethodId": "1007",
    "VersionNo": "1.0"
  },
  "Response": {
    "ResponseCode": "0",
    "ResponseMsg": "Success"
  },
  "Detail": {
    "ScannedData": "2345"
  }
}
```

8.8 Upload Invoice

This API uploads an invoice data coming from the Billing App to the server.

8.8.1 Request Parameter

Tag Name	Description	Type	Is Mandatory
BillingUserName	Username	String (10)	No
Customer	Customer Data	Object	No
AddressLine1	Address Line 1	String (50)	No
AddressLine2	Address Line 2	String (50)	No
City	City Name	String (50)	No
Country	Country Name	String (50)	No
DOB	Date of Birth in YYYY-MM-DD Format	String (10)	No
Email	Email Address	String (256)	No
FirstName	First Name	String (50)	No
Gender	Gender (MALE/FEMALE)	String (10)	No
LastName	Last Name	String (50)	No
Phone	Phone Number	String (20)	No
PinCode	Pin Code	Integer	No
State	State	String (50)	No
DiscountTotalValue	Total discount value	Long	No
GrossBill	Gross Bill Amount	Long	No
InvoiceNumber	Invoice Number	Long	No
NetBill	Net Bill Amount	Long	No
OrderCreationTimeLocal	Order creation time in YYYY-MM-DD format	String (10)	No
OrderId	Order Id	String (20)	No
PaymentStatus	Payment Status (PAID/PENDING)	String (10)	No
Payments	**List of Payments**	**Array []**	**No**
Amount	Amount	Long	No
CardType	Card Type (Credit/Debit)	String (10)	No
PaymentId	Payment Id	String (20)	No
PaymentType	Payment Type (Cash/Card)	String (10)	No
Products	**List of Products**	**Array []**	**No**
AdditionalChargeValue	Additional Charge Value	Long	No
BarCode	Bar Code	Long	No

DiscountTotalValue	Discount Total Value	Long	No
ProductBasePrice	Product Base Price	Long	No
ProductId	Product Id	String (20)	No
ProductName	Product Name	String (50)	No
ProductValue	Product Value	Long	No
Quantity	Quantity	Long	No
SkuId	SKU ID	Long	No
TaxTotalValue	Tax Total Value	Long	No
VoidAmount	Void Amount	Long	No
VoidQuantity	Void Quantity	Long	No
Status	Status (Delivered/Pending)	String (10)	No
Taxes	Taxes	Long	No

Sample JSON Request


```
{
  "Detail": {
    "BillingUserName": "nikhil",
    "Customer": {
      "AddressLine1": "E-block, Sector-62",
      "AddressLine2": "",
      "City": "Noida",
      "Country": "India",
      "DOB": "1992-06-13",
      "Email": "himanshu@gmail.com",
      "FirstName": "Himanshu",
      "Gender": "MALE",
      "LastName": "Jain",
      "Phone": 8506062503,
      "PinCode": 201309,
      "State": "Uttar Pradesh"
    },
    "DiscountTotalValue": 50,
    "GrossBill": 5500,
    "InvoiceNumber": 1,
    "NetBill": 5000,
    "OrderCreationTimeLocal": "2019-04-01",
    "OrderId": "OR-123-1",
    "PaymentStatus": "PAID",
    "Payments": [
      {
        "Amount": 2000,
        "CardType": "",
        "PaymentId": "123",
        "PaymentType": "PAYMENT_CASH"
      },
```

```
      {
        "Amount": 3500,
        "CardType": "CARD_VISA",
        "PaymentId": "456",
        "PaymentType": "PAYMENT_CARD"
      }
    ],
    "Products": [
      {
        "AdditionalChargeValue": 0,
        "BarCode": 0,
        "DiscountTotalValue": 50,
        "ProductBasePrice": 1000,
        "ProductId": "PROD2001",
        "ProductName": "Food Packet",
        "ProductValue": 5500,
        "Quantity": 5,
        "SkuId": 0,
        "TaxTotalValue": 100,
        "VoidAmount": 0,
        "VoidQuantity": 0
      }
    ],
    "Status": "DELIVERED",
    "Taxes": 100
  },
  "Header": {
    "ApplicationId": "1001",
    "MethodId": "1008",
    "UserId": "1234",
    "VersionNo": "1.0"
  }
}
```


8.8.2 Response Parameter

TagName	Description	Type
TerminalLogId	Identifier that identifies the log in that terminal	Integer
TerminalId	Identifier to identify a terminal	String (8)
ResponseCode	SUCCESS = 0; UPLOAD_INVOICE_NETWORK_ERROR = 23 UPLOAD_INVOICE_ERROR = 24	Integer
Response Msg	Success Saved locally, Upload to server failed Unable to save invoice data. Please Retry	String

Sample JSON Response


```
{
    "Header": {
        "ApplicationId": "1001",
        "UserId": "userId",
```

```
        "MethodId": "1008",
        "VersionNo": "1.0"
    },
    "Response": {
         "ResponseCode": "0",
         "ResponseMsg": "Success"
    },
    "Detail": {
        "TerminalLogId": 2,
         "TerminalId": "30000001"
    }
}
```

8.9 Scan QR/Barcode Code from Camera

This API will be called when the Billing App wants to scan any QR/Barcode through Camera on Plutus Smart Device.

8.9.1 Request Parameter

Sample JSON Request

```
{
  "Header": {
          "ApplicationId": "abcdefgh",
          "UserId": "user1234",
          "MethodId": "1009",
          "VersionNo": "1.0"
  }
}
```

8.9.2 Response Parameter

Tag Name	Description	Data Type
ResponseCode	SUCCESS = 0; APPLICATION_BUSY = 2001; UNKNOWN_ERROR = 2099;	Integer
ResponseMessage	Success Application is busy Unknown Error	String
ScannedData	Barcode code value	String

Sample JSON Response

```
{
  "Header": {
          "ApplicationId": "abcdefgh",
          "UserId": "user1234",
          "MethodId": "1009",
          "VersionNo": "1.0"
  },
```

```
  "Response": {
          "ResponseCode": "0",
          "ResponseMsg": "Success"
  },
  "Detail": {
          "ScannedData": "2345"
  }
}
```

9. Security Requirement

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

1. **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.**
2. **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.
3. **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.
4. **Business should review change management process in place with Third party service provider**
5. **Business should review processes in place by Third party service provider to periodically share patches for applications to minimize risks to Pine Labs**
6. **Following technical controls should be evaluated by Business such as:**
   a) Application interface authentication
   a) Data masking techniques
   b) Rest full API's
7. **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:

Sr. No.	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	

8. **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.

9. **Internal Approval**

Pine Labs Security team also requires approval from **Integration Team Head** before initiating any signing or security Assessment of application.

10. Glossary

10.1 Response Code

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

10.2 Method ID's

These Method ID's need to use in request parameter to perform different actions.

Code	Message
1001	Do Transaction
1002	Print Data
1003	Settlement
1004	Get Terminal Info
1005	Connect Bluetooth
1006	Disconnect Bluetooth
1007	Scan Code
1008	Upload Invoice
1009	Scan QR/Barcode from Camera

10.3 Transaction Types

These transaction type values can be use in DoTransaction Method ID to perform different transactions.

Sr. No.	Transaction Description	Transaction Type Value
1.	Sale Transaction	4001
2.	Refund Transaction	4002
3.	Tip Adjust Transaction	4015
4.	Adjust Transaction	4005
5.	Void Transaction	4006
6.	Pre Auth Transaction	4007
7.	Sale Complete Transaction	4008
8.	Loyalty Mine redemption	4201

9.	mWallet redemption	4214
10.	Pine 360 Loyalty Award	4208
11.	Pine 360 Loyalty Redeem	4209
12.	Pine 360 Loyalty Bal. Enquiry	4210
13.	Pine 360 PPC/GV Load	4202
14.	Pine 360 PPC/GV Redeem	4203
15.	Pine 360 PPC/GV Bal. Enquiry	4204
16.	Pine 360 Voucher Redeem	4215
17.	Pine 360 GC Load	4211
18.	Pine 360 GC Redeem	4212
19.	Pine 360 GC Bal Enquiry	4213
20.	Fetch Loyalty Number	4301
21.	Reward Redemption	4101
22.	Reward Void	4102
23.	Payback Earn	4401
24.	Payback Redemption	4402
25.	Payback Void	4403
26.	Sale with rebate	4501
27.	Sale with cash	4502
28.	Cash Only	4503
29.	Reprint	4504
30.	COD Sale / Cash	4507
31.	COD Void / Cash Void	4508
32.	Wallet Pay	5102
33.	Wallet Load	5103
34.	Wallet Void	5104
35.	Sodexo Sale	5106
36.	Sodexo Void	5107
37.	UPI Sale	5120
38.	Void	5121
39.	Get Status	5122
40.	Bharat QR Sale	5123

11. References

- Github link to download simulator app: Click here
- Github link to download sample billing app: Click here