Pine Labs Payment Gateway Android SDK Integration V1.0

CONTENTS

[1 About This Guide 3](#_Toc9261702)

[1.1 Objectives and Target Audience 3](#_Toc9261703)

[1.2 Connecting to the Payment Gateway through Android SDK 3](#_Toc9261704)

[1.3 Related Documentation 3](#_Toc9261705)

[2 Android SDK Payment Flow 3](#_Toc9261706)

[2.1 Flow Diagram 4](#_Toc9261707)

[3 SDK Integration 5](#_Toc9261708)

[3.1 Step 1 SDK Installation and Setup 5](#_Toc9261709)

[3.2 Step 2 Initialization 5](#_Toc9261710)

[3.3 Step 3 Initiate Payment 6](#_Toc9261711)

[3.4 Step 4 Handling call back from Pine Labs SDK. 7](#_Toc9261712)

[4 Transaction Verification 7](#_Toc9261713)

# About This Guide

## Objectives and Target Audience

This guide provides the details on how to connect merchant‘s android application with Pine Labs payment gateway and use its services. It is intended for users who want to carry out integration with payment gateway.

## Connecting to the Payment Gateway through Android SDK

When customer selects make payment or pay now in android app, merchant app needs to call the Pine labs android SDK interface which contains information about the payment such as MID, Order Id, Amount to be paid and other several fields.

## Related Documentation

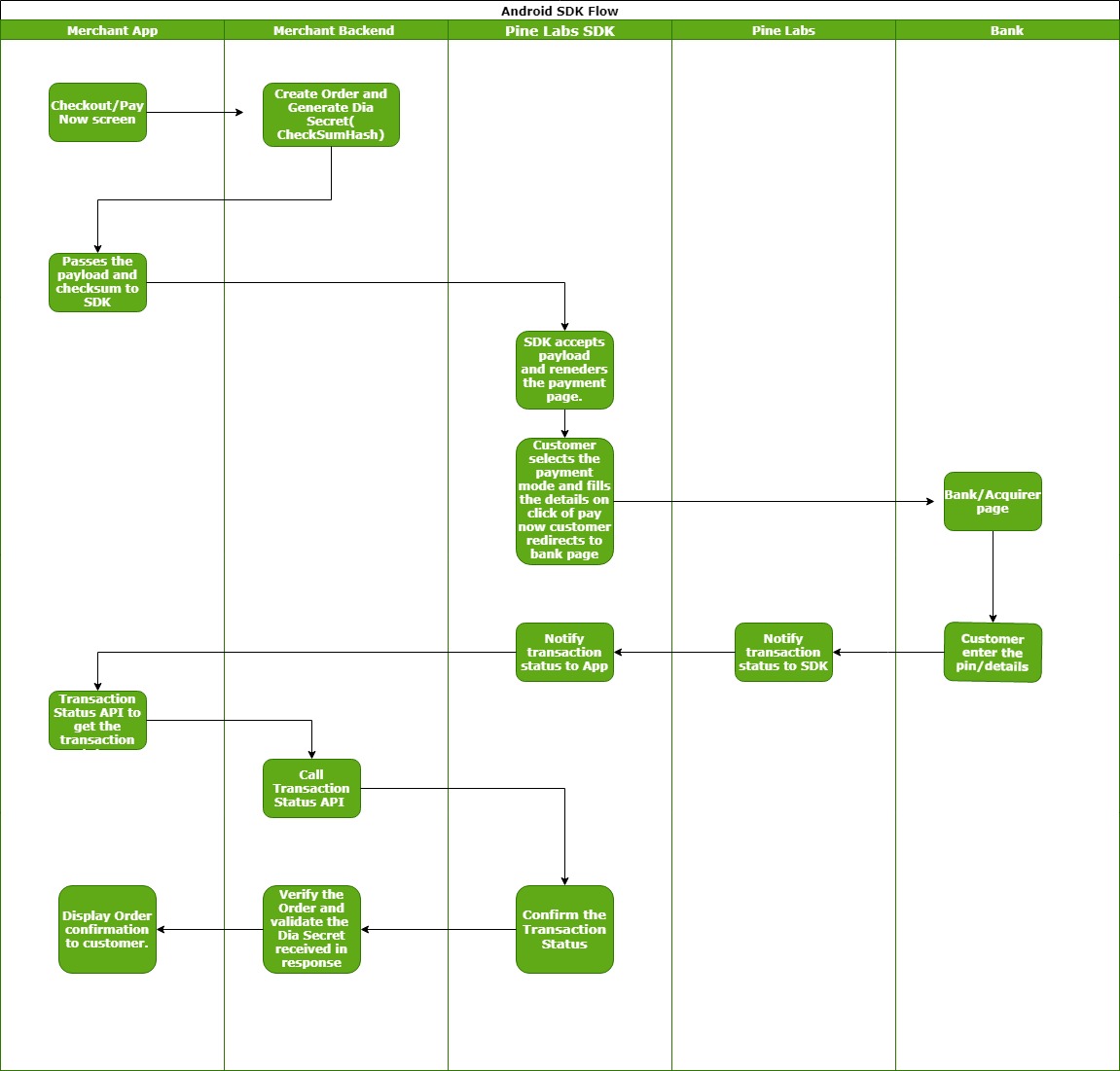
This guide should be used together with the additional documents as described below.

|  |  |
| --- | --- |
| **Document** | **Description** |
| Pine Labs payment gateway integration guide.pdf | Payment gateway Integration guide |

# Android SDK Payment Flow

* **Step 1** : On click of make payment/pay now button,order payload is passed to merchant server by the app.
* **Step 2**: Order payload is used to generate checksum by Pine Labs server side utility and secret key(merchant key)on your server. Secret key is shared by Pine labs with merchant. Checksum is a signature used by Pine Labs to ensure the integrity of request.
* **Step 3** : Merchant server pass the payload , Dia secret and Dia secret type back to app .Merchant apps pass all the details to Pine Labs android sdk.
* **Step 4** : Pine Labs sdk accepts and forward the payload to Pine Labs payment gateway.
* **Step 5** : Pine labs payment gateway verifies the payload and accept/denied the request. If payload is valid payment options will be displayed.
* **Step 6** : Once the customer fill the payment details and complete the payment, then merchant app is notified via call back.
* **Step 7** : Merchant verifies the transaction status with transaction status API via server to server call.

## Flow Diagram



# SDK Integration

## Step 1 SDK Installation and Setup

Install Pine Labs android SDK using android studio and IntelliJ. To add the SDK to your app, add the following dependency in your build.gradle :

dependencies {

implementation project(‘:PineLabsSDK’)

}

**Add the following code to your AndroidManifest.xml to get static permission**

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

## Step 2 Initialization

Object Service : To initialize the Pine Labs SDK ,create the service object

**PinePGPaymentManager service=new PinePGPaymentManager();**

Object order : Stores all order related information which are required to be passed by you to Pine Labs.  Order object is created by following code snippet

Map<String,String> orderParams=new HashMap<>();

orderParams.put("ppc\_UniqueMerchantTxnID" ,"Order ID 123");

orderParams.put("ppc\_MerchantID" ,"34133");

orderParams.put("ppc\_Amount" ,"12345");

orderParams.put("ppc\_MerchantAccessCode" ,"58ad283b-7c93-4f19-b072-b17e8ecfb20e");

orderParams.put("ppc\_NavigationMode" ,"2");

orderParams.put("ppc\_TransactionType" ,"1");

orderParams.put("ppc\_LPC\_SEQ" ,"1");

orderParams.put("ppc\_Product\_Code" ,"40");

orderParams.put("ppc\_PayModeOnLandingPage" ,"1,3,4,7"");

orderParams.put("ppc\_CustomerEmail" ,"test@pinelabs.com");

orderParams.put("ppc\_CustomerMobile" ,"9876543210");

orderParams.put("ppc\_CustomerId" ,"123");

orderParams.put("ppc\_CustomerAddress1" ,"Noida B block");

orderParams.put("ppc\_CustomerAddressPIN" ,"201301");

orderParams.put("ppc\_DIA\_SECRET" ,"w2QDRMgp1234567JEAPCIOmNgQvsi+BhpqijfM9KvFfRiPmGSt3Ddzw+oTaGCLneJwxFFq5mqTMwJXdQE2EzK4px2xruDqKZjHupz9yXev4=");

orderParams.put("ppc\_DIA\_SECRET\_TYPE" ,"SHA256");

Parameter details can be found in Purchase request section ***6.3.1.3***  *Pine Labs payment gateway integration guide.pdf*

## Step 3 Initiate Payment

service.startPayment(orderParams,context,iThemeIdVal,isHeaderTobeShow,isProductionRequest, new IPinePGResponseCallback(){

@Override

public void internetNotAvailable(int code, String message) {}

@Override

public void onErrorOccured(int code, String message) { }

@Override

public void onTransactionResponse() {}

@Override

public void onCancelTxn(int code, String message) {}

@Override

public void onPressedBackButton(int code, String message) {}

});

**Parameters used in start Payment in order are –**

1. **context** : context of your Activity is where this method is called.
2. **iThemeIdVal:** A Integer variable(possible value 0,1,2) to apply theme on Pine labs SDK.
3. **isHeaderTobeShow** :A Boolean variable (true/false) to hide or show header bar.
4. **isProductionRequest** : A boolean variable (true/false) to determine whether request is for production environment .
5. **IPinePGResponseCallback** : IPinePGResponseCallback instance to send call back messages back to merchant android app.

## Step 4 Handling call back from Pine Labs SDK.

You need to implement call back methods to handle payment response. This will provide the payment status and reason for transaction failures. Based on the reasons for failures, handling can be built at your end. Transaction call backs can be listened via overriding methods of IPinePGResponseCallback.

**onTransactionResponse** method is called when transaction is complete. Transaction can be fail or success.

**internetNotAvailable** method is called when internet is not available.

**onErrorOccured** method is called when SDK is unable to load the payment a page.

**onPressedBackButton** method is called when user press the back button

**onCancelTxn** method is called when user cancels the transaction.

# Transaction Verification

It is the Rest API used to get the transaction status of the transaction. Merchant needs to call this api after receiving callback of transation.

Parameter details can be found in dependent transaction request section ***6.4***  *Pine Labs payment gateway integration guide.pdf*