**Payment Gateway Integration for Android App**

This section will guide you to integrate our Payment Gateway SDK seamlessly inside your android app.

**Prerequisites**

Below are the prerequisites to integrate with Payment Gateway:

* Be an Approved Merchant: To use our payment gateway, you should be a registered and approved merchant by us. If not registered, please register to get started!
* Obtain API Key: You should have received an API key from us on Approval.
* API to receive Payment Response: You should have a self hosted web server to receive the response and verify the payment details post payment.
* Min SDK Requirements: You should ensure that your Android App’s SDK version should be greater than or equal to 21.

**Downloads**

* payment gateway android integrations Link:

<https://github.com/Baabujiventuress/BasisPay-Android-KIT.git>

* Our payment gateway android SDK can be downloaded here:

Download AAR for SDK 19 to 27

Download AAR for AndroidX

* A sample Android App that illustrates the integration of any app with our SDK can be downloaded here:

Download SampleAPP with AAR for SDK 19 to 27

Download SampleAPP with AAR for AndroidX

**Integration Steps**

1. Download the BasisPay-Android-KIT zip folder and unzip it.

Copy the PGSDKVR4 AAR file and save this in your Desktop.

**Prerequisites**

Your Android App’s SDK Version must be greater than or equal to 19.

**Add the PGSDKVR4 AAR library to your app:**

* Click File > New > New Module.
* Click Import .JAR/.AAR Package then click Next.
* Enter the location of the PGSDKVR4 AAR file then click Finish.

Make sure PGSDKVR4 library is listed at the top of your settings.gradle file, as shown below:

**For SDK 19-27**

include ':app', ':PGSDKVR4'

**For AndroidX**

include ':app', ':pgsdkv5ax'

Open your app module’s build.gradle file and add the below line to the dependencies block as shown in the following snippet:

**For SDK 19-27**

dependencies

{

implementation project(":PGSDKVR4")

}

**For AndroidX**

dependencies

{

implementation project(":pgsdkv5ax")

}

**Click Sync Project with Gradle Files**

Make sure you have the below permissions in your manifest file:

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

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

**Code Explanation**

**Note**

Your App must use the latest security standards to prevent your code being compromised.



* User starts the payment in the client's app.
* Client’s code then sets the payment parameters and initiates the payment process via the SDK.
* The SDK in-turn interacts with the Payment Gateway server during the payment process.
* After the payment, the Payment Gateway sends the payment response to the Client’s web server(Via the Return URL)and to the SDK.At this point, the client’s code in the Client’s web server code should re verify the hash in the payment response and store the response in the Database.
* The SDK parses the payment response and converts into json and provides it to the Client’s code. At this point, the client should compare the amount and order id with their corresponding value DB in the web server.
* If the values match,Client’s Code then displays the payment response to the User.

**Initiate the “com.test.pg.secure.pgsdkv4.PaymentParams” CLASS to**

**set the payment parameters:**

**Set the payment parameters**

**PaymentParams pgPaymentParams = new PaymentParams();**

**pgPaymentParams.setAPiKey(SampleAppConstants.PG\_API\_KEY);**

**pgPaymentParams.setAmount(SampleAppConstants.PG\_AMOUNT);**

**pgPaymentParams.setEmail(SampleAppConstants.PG\_EMAIL);**

**pgPaymentParams.setName(SampleAppConstants.PG\_NAME);**

**pgPaymentParams.setPhone(SampleAppConstants.PG\_PHONE);**

**pgPaymentParams.setOrderId(SampleAppConstants.PG\_ORDER\_ID);**

**pgPaymentParams.setCurrency(SampleAppConstants.PG\_CURRENCY);**

**pgPaymentParams.setDescription(SampleAppConstants.PG\_DESCRIPTION);**

**pgPaymentParams.setCity(SampleAppConstants.PG\_CITY);**

**pgPaymentParams.setState(SampleAppConstants.PG\_STATE);**

**pgPaymentParams.setAddressLine1(SampleAppConstants.PG\_ADD\_1);**

**pgPaymentParams.setAddressLine2(SampleAppConstants.PG\_ADD\_2);**

**pgPaymentParams.setZipCode(SampleAppConstants.PG\_ZIPCODE);**

**pgPaymentParams.setCountry(SampleAppConstants.PG\_COUNTRY);**

**pgPaymentParams.setReturnUrl(SampleAppConstants.PG\_RETURN\_URL);**

**pgPaymentParams.setMode(SampleAppConstants.PG\_MODE);**

**pgPaymentParams.setUdf1(SampleAppConstants.PG\_UDF1);**

**pgPaymentParams.setUdf2(SampleAppConstants.PG\_UDF2);**

**pgPaymentParams.setUdf3(SampleAppConstants.PG\_UDF3);**

**pgPaymentParams.setUdf4(SampleAppConstants.PG\_UDF4);**

**pgPaymentParams.setUdf5(SampleAppConstants.PG\_UDF5);**

**Initialize the com.test.pg.secure.pgsdkv4.PaymentGatewayPaymentInitializer class with payment parameters and initiate the payment:**

**Initiate the payment**

**PaymentGatewayPaymentInitializer pgPaymentInitialzer = new PaymentGatewayPaymentInitializer(pgPaymentParams,MainActivity.this);**

**pgPaymentInitialzer.initiatePaymentProcess();**

To receive the json response, override the onActivityResult() using the REQUEST\_CODE and PAYMENT\_RESPONSE variables from com.test.pg.secure.pgsdkv4.PaymentParams class

**Payment Response Code**



**Note**

Request parameters are the parameters that will be sent to our server API for payment initiation. Client should store the order id and the amount before payment initiation and compare it with the order id and amount in the response Json from our server post payment process to ensure no end user tampering on the requested parameters.

**List of Request Parameters**







**List of Response Codes**

**Note**

Below are the response codes that come in the payment response post payment from our server, that must be handled by the client.









**THANK YOU**