PayUMoney SDK Integration Document

(For Android Studio)

4th Floor, Pearl Towers
Plot 51, Sector 32
Gurgaon, 122001
Phone: 0124-6624956
0124-6624970
Email: techsupport@payumoney.com

Overview

This note describes the how to do the technical integration between PayUMoney Payment Gateway and your website in respect of powering online transactions.

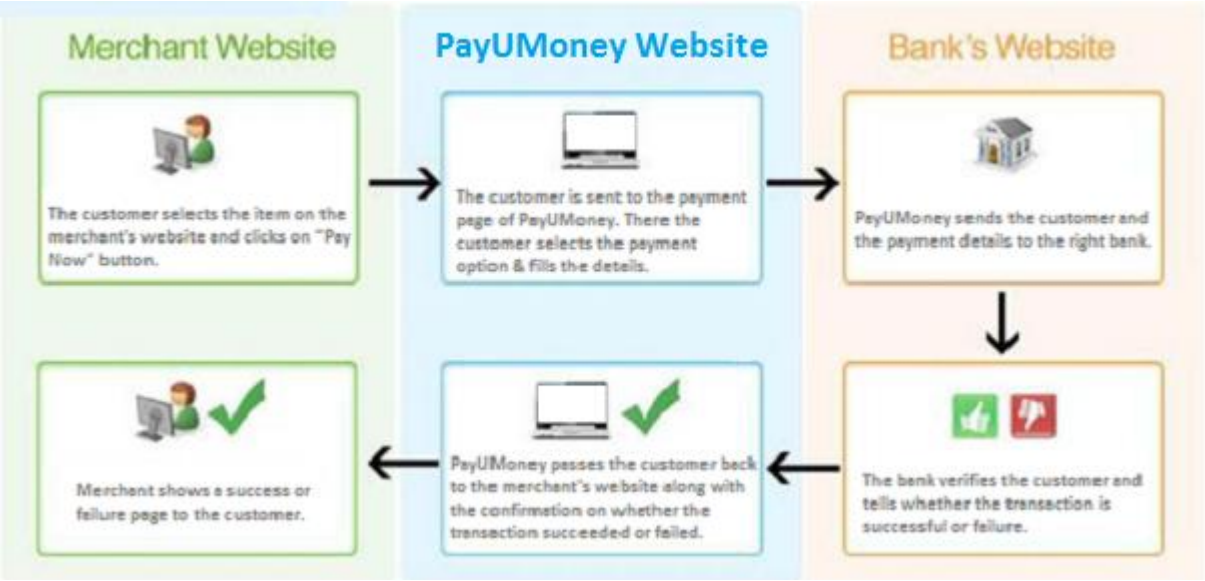
PayUMoney Payment Gateway

PayUMoney offers electronic payment service to your website through its various partnerships with banks and payment instrument companies. Through PayUMoney, your clients would be able to make electronic payments through credit card, debit card and online net banking account

PayUMoney also offers an online interface where the merchant can view transaction details, settlement reports, analytic reports etc. This online interface can be accessed through https://www.payumoney.com by using the username and password provided to you.

Payment Process Flow

The following diagram explains how the customer makes the payment and how the process flows:



Step 1: The consumer selects the product on your website and clicks on “Pay Now” button.

Step 2: The consumer is then taken from your website to the transaction page of www.payumoney.com where in all the payment related details are entered by the consumer.

Step 3: Payumoney.com.com redirects the consumer to Visa, MasterCard or the relevant bank for the next level of authorization.

Step 4: The Bank/Visa/MasterCard authorizes and confirms the transaction.

Step 5: The consumer is sent back to PayUMoney.

Step 6: PayUMoney sends the consumer back to your website along with the transaction status.

Status of a Transaction

A transaction can have several different statuses as explained below.

1. **Not Started** – The transaction has not been started yet.
2. **Initiated** – The transaction has been started but not completed.
3. **Money With PayUMoney**– The transaction was successful and the transaction amount is with PayUMoney.
4. **Under Dispute** – A dispute for the transaction has been raised.
5. **Refunded** – The entire amount of the transaction has been refunded.
6. **Partially Refunded** – A part of the amount of the transaction has been refunded.
7. **Bounced** – Incomplete or no details provided at PayUMoney payment page.
8. **Failed** – The transaction didn't complete due to a failure.
9. **Settlement in Process** – Settlement for the transaction is in process.
10. **Completed** – The transaction is settled and complete.

Settlement process

Settlement is the process by which the money gets transferred from the customer to the bank account of the merchant. PayUMoney follows a T+2 settlement scheme where T is the date on which the transaction is captured.

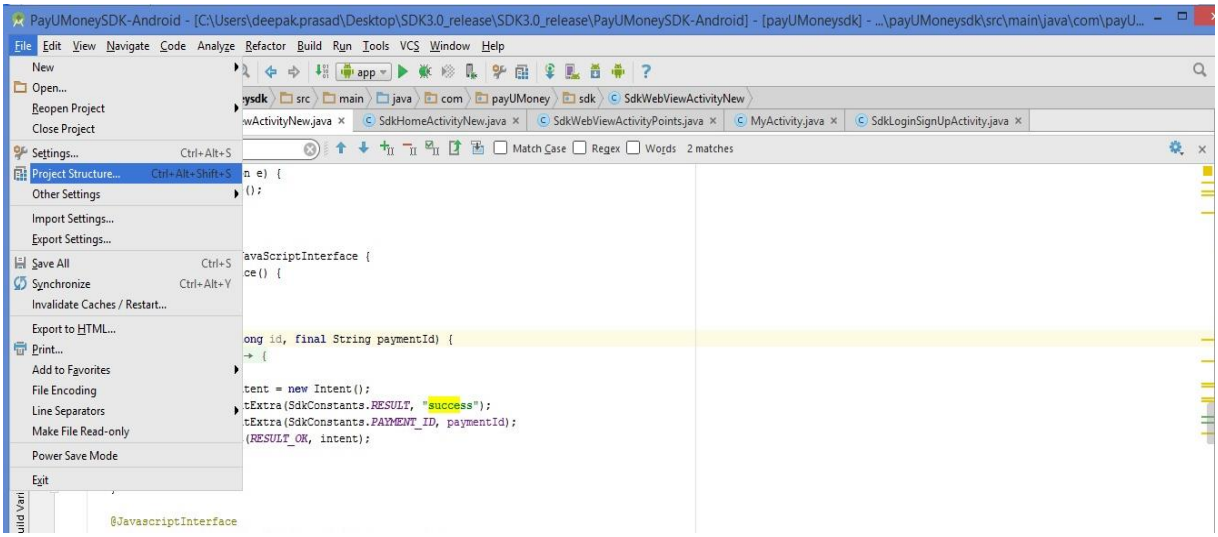
There is a reconciliation process at PayUMoney. On the next day, after you have captured the transactions, PayUMoney will reconcile the online transactions with the credits received based on batch files received from the banks. After reconciling, we will generate a report and payment will be made for all the transactions for which payment has been received from the bank. All the details will be visible to you in the online interface.

Technical Integration

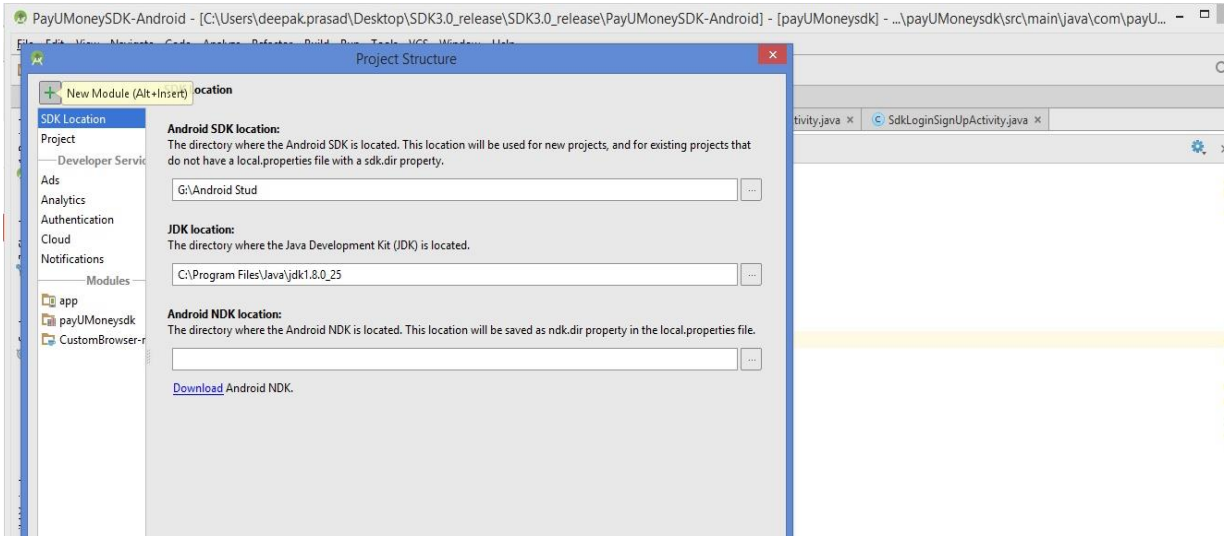
To integrate with the PayUMoney Android SDK –

1. Import PayUmoney SDK as a module or Gradle Project

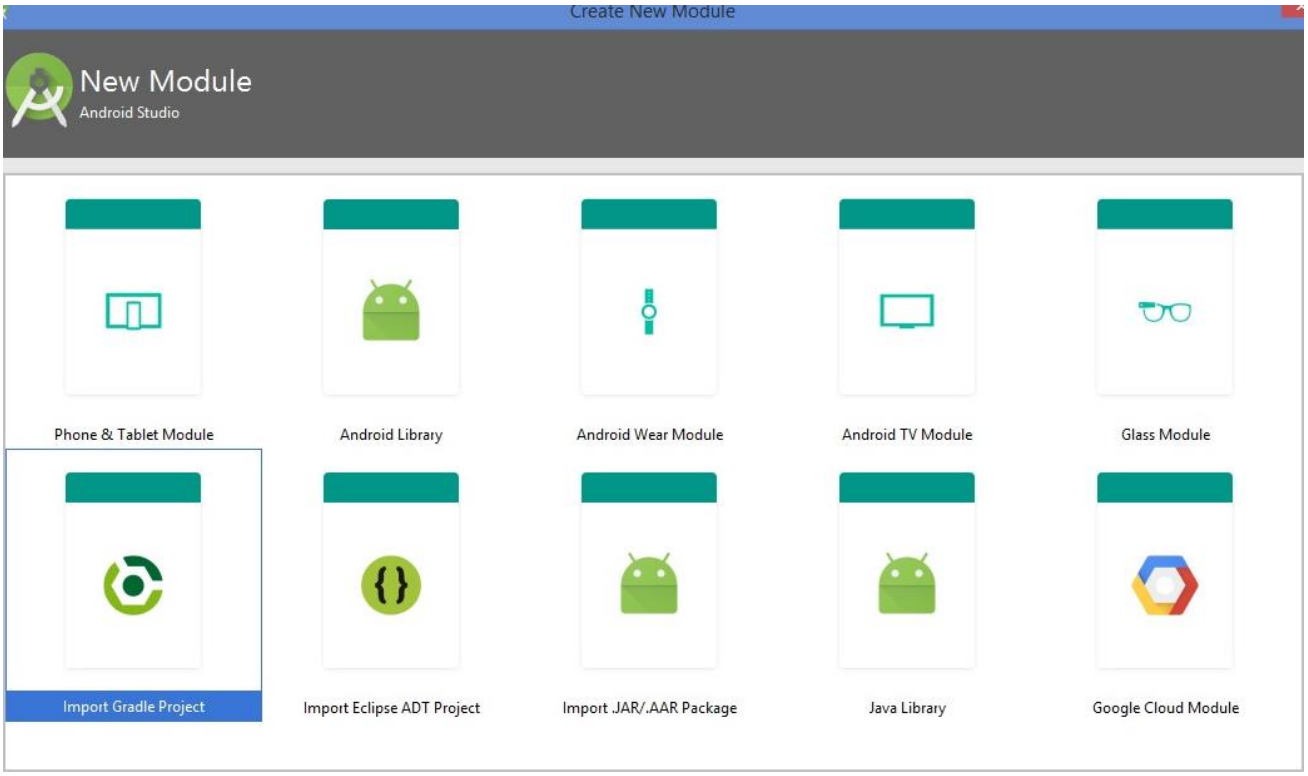
a. Go to Project Structure under file in the Android Studio.



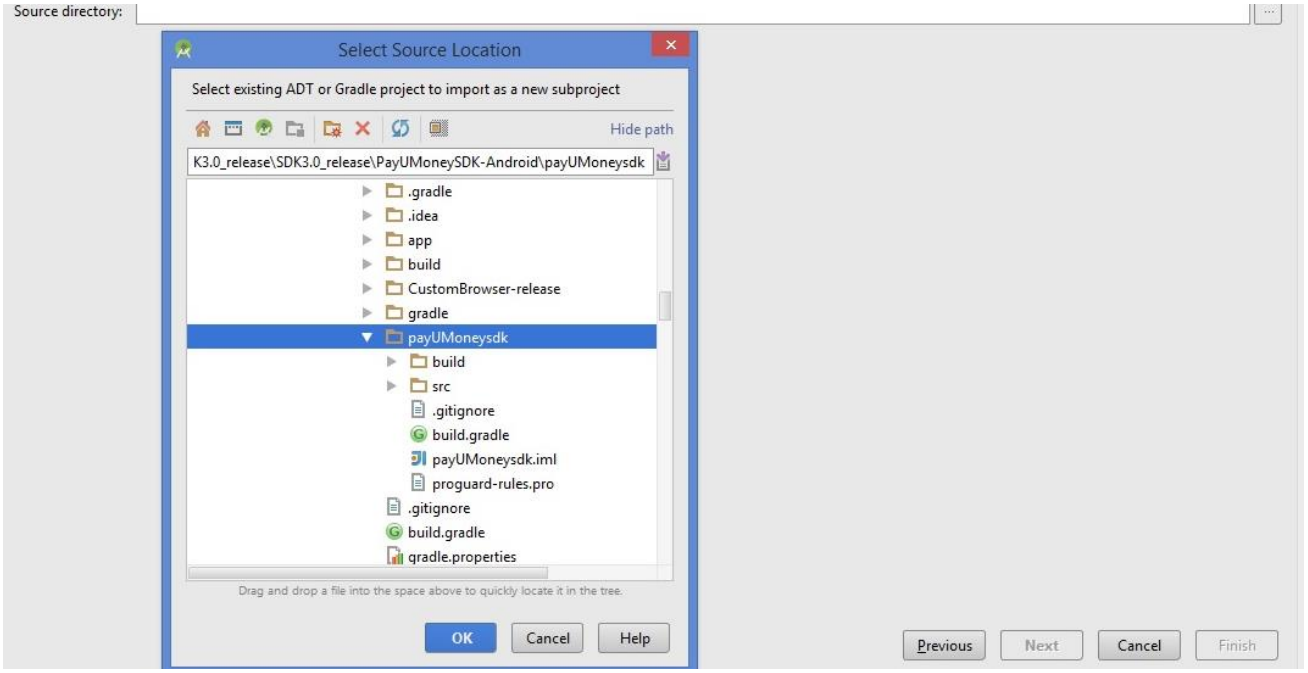
b. Click on New Module (+ sign).



c. Select ‘Import Gradle Project’.



d. Browse and select the PayUmoney SDK file. Now click on Next.

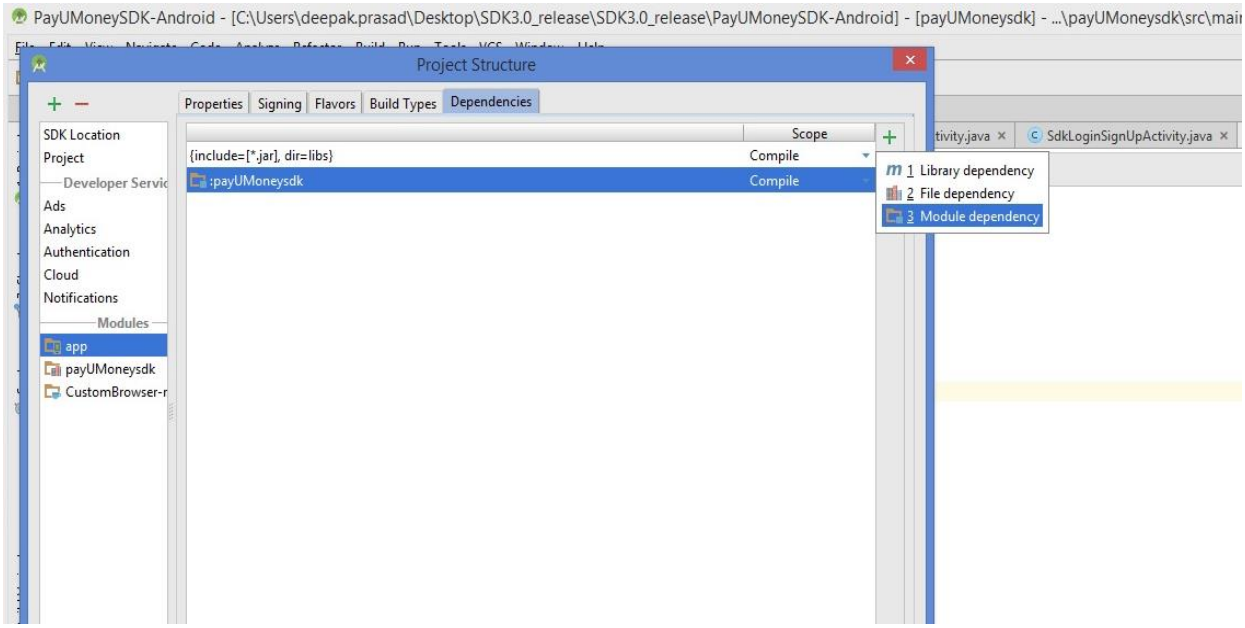


Please wait for some time while the synching is completed.

2. Add it as a dependency in your project.

a. Adding SDK as a dependency to your app.
Click on your app and then select the Dependencies tab on the top. Then click on Add (+ sign) on the top right corner and click on Module Dependency. After this, select PayuMoney SDK.

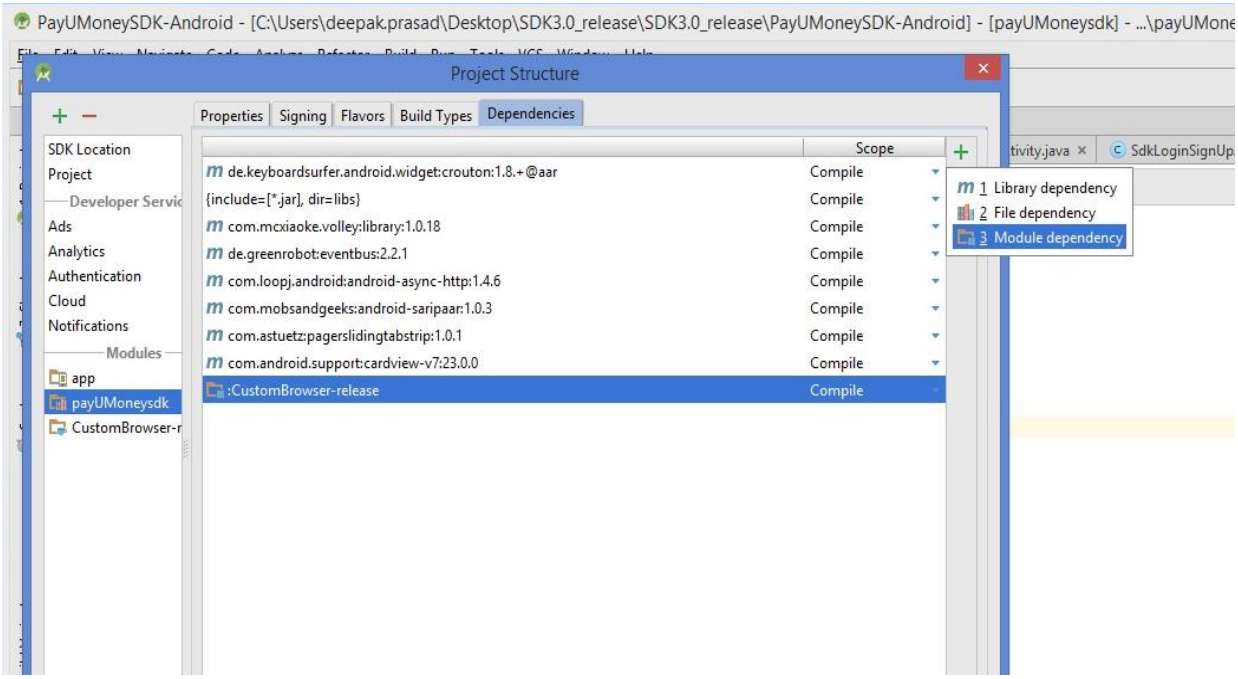
This will add PayUmoney SDK as a dependency to your app (refer the screenshot below).



b. Adding CustomBrowser as a dependency to PayUmoney SDK.

Click on your app and then select the Dependencies tab on the top. Then click on Add (+ sign) on the top right corner and click on Module Dependency. After this, select CustomBrowser-release.

This will add PayUmoney SDK as a dependency to your app (refer the screenshot below).



Once you add the sdk as module you will need to call the following function from app when the user clicks on **Pay Now** button on your app.

Sdk.Session.startPaymentProcess(context,params);

context is the activity context from where this function is called.

Params is the Hashmap that contains the required parameter for payment process.

Following is the list of required paramaters.
HashMap<String,String> params = new HashMap<String, String>();

S.No	Parameter Name	Required	Value
1	key	Compulsory	Merchant Key provided by PayUMoney
2	Txnid	Compulsory	
3	Amount	Compulsory	Payment amount (Type cast the amount to float)
4	ProductInfo	Compulsory	Product Description
5	firstName	Compulsory	(only alphabets a-z are allowed)
6	Email	Compulsory	Customer's email Id
7	phone	Compulsory	mobile number or landline number (numeric value only)
8	udf1		user defined field 1
9	udf2		user defined field 2
10	udf3		user defined field 3
11	udf4		user defined field 4

12	udf5		user defined field 5
13	SURL	Compulsory	The URL to be called when a payment is completed.
14	FURL	Compulsory	The URL to be called when a payment fails.
15	hash	Compulsory	Hash or Checksum =sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5|salt) (SALT will be provided by PayUMoney)

Please Note – udf1 to udf5 are user-defined fields. These are meant to send any additional values that you need to post. However, if you don't feel the need to post any additional params, even then you will need post these params with blank values.

Hash calculation

String hashSequence =
key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5|<salt>;
String hash = hashCal("SHA-512", hashSequence);

Hash Algorithm

```
public static String hashCal(String type, String str) {
byte[] hashseq = str.getBytes();
StringBuffer hexString = new StringBuffer();
try {
MessageDigest algorithm = MessageDigest.getInstance(type);
algorithm.reset();
algorithm.update(hashseq);
byte messageDigest[] = algorithm.digest();
for (int i = 0; i < messageDigest.length; i++) {
String hex = Integer.toHexString(0xFF & messageDigest[i]);
if (hex.length() == 1) {
     hexString.append("0");
       }
  hexString.append(hex);
    }
  } catch (NoSuchAlgorithmException nsae) {
  }
```

```
 return hexString. toString();

}
```

Payment Completion

To know when the payment has completed, override the onActivityResult in your activity.
Example :

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
if (requestCode == Session.PAYMENT_SUCCESS) {
if (resultCode == RESULT_OK) {
//success
Toast.makeText(this, "success",
Toast.LENGTH_LONG).show();
}
if (resultCode == RESULT_CANCELED) {
//failed
Toast.makeText(this, "failed",
Toast.LENGTH_LONG).show();
          }
      }
 }
```

Now, you will need to add these lines of JavaScript to your success and failure pages (SURL, FURL) for the Android SDK to be able to detect the result.
Success Page:
PayUMoney.success()
Failure Page:
PayUMoney.failure()

Using SDK in Test mode

For using sdk in test mode you need to follow the below mentioned steps.

1. Open Constants.java class under src folder in the sdk.
2. Search for Boolean DEBUG in the constant class.
3. Change the status to true for test mode and false for the live mode.
4. Send the SURL/FURL, MerchantId and key accordingly for test and live mode.

About Custom Browser

Custom Browser is a feature of reading the sms OTP automatically.

You can get in touch with us:

Phone: 0124-6624956 & 0124-6624970
Email: techsupport@payumoney.com