PayU Android SDK 4.0 Documentation

Getting Started

Welcome to Payu mobile sdk documentation. Payu supports multiple integration methods. Given below are tutorials and reference documents for easy integration.

Android

Our Android guide shows you how to include PayU's Native Android SDK into your Android application. This document will guide you through the Android Integration.

Prerequisites

You must already have installed and configured:

- **Java JDK version 1.6 or greater.**
- **Android SDK**
- **A Git client**
- **Android Studio / Eclipse**
- **All PayU PG Prerequisites. (Key, Salt)**

Installation

Get the latest payu sdk form github

https://github.com/payu-intrepos/Android-SDK

What are different Environments? When I can use those

1. SDK supports two different environments. Mobile Staging(Testing) and Production. Testing should be done on Mobile Staging. Once your app is working fine with Mobile Staging

environment, you can switch environment to Production. Make sure you have set environment to Production before releasing your app to Play-Store.

2. Setup your environment using

```
PayuConfig payuConfig = new PayuConfig();
payuConfig.setEnvironment(PayuConstants.PRODUCTION_ENV);
payuConfig.setEnvironment(PayuConstants.MOBILE_STAGING_ENV);
```

URLS For Production Server:

Web Service URL: https://info.payu.in/merchant/postservice

Payment URL : https://test.payu.in/_payment

For Mobile Staging Server:

Web Service URL: https://mobiletest.payu.in/merchant/postservice

Payment URL : https://secure.payu.in/_payment

User / Merchant Registered on Staging environment will not be available on production environment. You have to use your production credentials (key and salt) for Production environments.

Integration Steps

PayU sdk is the easiest way to integrate payu gateway on any merchant android application. It provides a PaymentDefaultParams object and a PaymentModeParam objects (CCDDCard, NBPostParams, CashCardPostParams, StoredCardPostParams, StoredCardPostParams, PayuWalletPostParams **) files containing functions which internally calls PayU gateway. A merchant just need to make use of these function calls to handle payment from customer. It provides functionality to handle payment using any credit card, debit card or net banking, Payumoeny options.**

1) Server side integration steps

a) Generate Hash

What is Hash?

Every transaction needs a hash by the merchant before sending the transaction details to PayU. This is required for PayU to validate the authenticity of the transaction. This should be done on your server.

Sample hash generation for payment

Payment Hash

sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf

4|udf5||||||SALT)

Webservice Hash

sha512(key|command|var1|salt)

please refer the web integration document for more details.

2) Client side (Front-end) Integration step.

Create an PaymentDefaultParams object and set the required params as follows.

```
PaymentDefaultParams paymentDefaultParams = new PaymentDefaultParams();
paymentDefaultParams.setKey(“gtKFFx”);
paymentDefaultParams.setAmount(“15.0”);
paymentDefaultParams.setProductInfo(“Tshirt”);
paymentDefaultParams.setFirstName(“Guru”);
paymentDefaultParams.setEmail(“guru@gmail.com”);
paymentDefaultParams.setTxnId(“0123479543689”);
paymentDefaultParams.setSurl(“https://payu.herokuapp.com/success”);
paymentDefaultParams.setFurl(“https://payu.herokuapp.com/failure");
paymentDefaultParams.setUdf1(“udf1l”);
```

```
            paymentDefaultParams.setUdf2(“udf2”);
            paymentDefaultParams.setUdf3(“udf3”);
            paymentDefaultParams.setUdf4(“udf4”);
            paymentDefaultParams.setUdf5(“udf5”);
            // calucalte hash from your server using the following formula
hash = sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5||||||SALT)
            PaymentDefaultParams.setHash(hash);
```

1. Integrating cc/dc card.

Create an activity for card payment
create PaymentDefaultParams object and set the mandatory params in onCreate function.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_card);

   findViewById(R.id.button_card_make_payment).setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {

 //create an CCDDCard object and set the mandatory params

CCDCCard ccdcCard = new CCDCCard();
ccdcCard.setCardName(“test card”);
ccdcCard.setNameOnCard(“pay test”);
ccdcCard.setCardNumber(“5123456789012346”);
ccdcCard.setCvv(“123”);
ccdcCard.setExpiryMonth(“01”); // MM
ccdcCard.setExpiryYear(“2017”); // YYYY

PostData postData = new CCDCPostParams(paymentDefaultParams, ccdcCard).getCCDCPostParams();

if (postData.getCode() == PayuErrors.NO_ERROR) {
   // okay good to go.. lets make a transaction
   // launch webview
   PayuConfig payuConfig = new PayuConfig();
   payuConfig.setEnvironment(PayuConstants.PRODUCTION_ENV);
   payuConfig.setData(postData.getResult());
```

```
  Intent intent = new Intent(this, PaymentsActivity.class);
  intent.putExtra(PayuConstants.PAYU_CONFIG, payuConfig);
  startActivityForResult(intent, PayuConstants.PAYU_REQUEST_CODE);
} else { // something went wrong
  Toast.makeText(this, postData.getResult(), Toast.LENGTH_LONG).show();
}
    }
  });

}
```

Create an activity for Webview

```
@Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_payments);

    // initialize
    Bundle bundle = getIntent().getExtras();
    PayuConfig payuConfig = bundle.getParcelable(PayuConstants.PAYU_CONFIG);

    WebView mWebView = (WebView) findViewById(R.id.webview);

    URL url = payuConfig.getEnvironment() == PayuConstants.PRODUCTION_ENV?
PayuConstants.PRODUCTION_PAYMENT_URL : PayuConstants.MOBILE_TEST_PAYMENT_URL ;

    byte[] encodedData = EncodingUtils.getBytes(payuConfig.getData(), "base64");
    mWebView.postUrl(url, encodedData);

    mWebView.getSettings().setSupportMultipleWindows(true);
    mWebView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
    mWebView.getSettings().setJavaScriptEnabled(true);
    mWebView.getSettings().setDomStorageEnabled(true);
    mWebView.setWebChromeClient(new WebChromeClient() {});
    mWebView.setWebViewClient(new WebViewClient() {});
  }
```

At this point you should be able to see Bank’s 3’d Secure page.

2. Integrating Netbanking

create PaymentDefalutParams object and assign all the mandatory params.
(same as credit card payment)

Get the PaymentDetails object of selected card from your spinner/list view adapter

```
PaymentDetails paymentDetails = mNetBankingList.get(position);
```

once this is done just like cc/dc card create postdata

```
PostData postData = new NBPostParams(mPaymentDefaultParams, mNetBank).getNBPostParams();
if (postData.getCode() == PayuErrors.NO_ERROR) {
   // launch webview
   payuConfig.setData(postData.getResult());
   Intent intent = new Intent(this, PaymentsActivity.class);
   intent.putExtra(PayuConstants.PAYU_CONFIG, payuConfig);
   startActivityForResult(intent, PayuConstants.PAYU_REQUEST_CODE);
} else {
   Toast.makeText(this, postData.getResult(), Toast.LENGTH_LONG).show();
}
```

At this point you should be able to see Bank’s 3’d Secure page.

Integrating CASH card

create payment default params just like netbank.

```
PostData postData = new CashCardPostParams(mPaymentDefaultParams,
mCashCardList.get(position)).getCashPostParams();

if(postData.getCode() == PayuErrors.NO_ERROR){
   // launch webview
   payuConfig.setData(postData.getResult());
   Intent intent = new Intent(this, PaymentsActivity.class);
   intent.putExtra(PayuConstants.PAYU_CONFIG, payuConfig);
   startActivityForResult(intent, PayuConstants.PAYU_REQUEST_CODE);
}else{
   Toast.makeText(this, postData.getResult(), Toast.LENGTH_LONG).show();
}
```

Integrating Stored Card

get all the payment default params just like credit card

```
// storedCardList.get(position) is the actual stored card
// we need cvv also . which we get from ui,

makePayment(storedCardList.get(position), cvvEditText.getText().toString());

// then

private void makePayment(StoredCard storedCard, String cvv) {
   PostData postData = new PostData();
   // lets try to get the post params
   postData = null;
   storedCard.setCvv(cvv); // make sure that you set the cvv also
   mPaymentDefaultParams.setHash(payuHashes.getPaymentHash()); // make sure that you set payment hash
   postData = new StoredCardPostParams(mPaymentDefaultParams, storedCard).getstoredCardPostParams();
   if (postData.getCode() == PayuErrors.NO_ERROR) {
       payuConfig.setData(postData.getResult());
       Intent intent = new Intent(this, PaymentsActivity.class);
       intent.putExtra(PayuConstants.PAYU_CONFIG, payuConfig);
       startActivityForResult(intent, PayuConstants.PAYU_REQUEST_CODE);
   } else {
       Toast.makeText(this, postData.getResult(), Toast.LENGTH_SHORT).show();
   }

}
```

At this point you should be able to see Bank’s 3’d Secure page.

Integrating PayuMoney

```
private void launchPayumoney() {
   PostData postData;

   // lets try to get the post params
   mPaymentDefaultParams.setHash(mPayUHashes.getPaymentHash());
```

```
    postData = new PayuWalletPostParams(mPaymentDefaultParams).getPayuWalletPostParams();
    if(postData.getCode() == PayuErrors.NO_ERROR){
        // launch webview
        payuConfig.setData(postData.getResult());
        Intent intent = new Intent(this, PaymentsActivity.class);
        intent.putExtra(PayuConstants.PAYU_CONFIG, payuConfig);
        startActivityForResult(intent, PayuConstants.PAYU_REQUEST_CODE);
    }else{
        Toast.makeText(this, postData.getResult(), Toast.LENGTH_LONG).show();
    }
}
```

3. Return URL - (SURL / FURL)

Return URL is where PayU redirects the user after the transaction is completed. PayU sends the data related to transactions while redirecting so that you can check the status of the transaction.

How to create surl/furl page?

The surl/furl page is hosted on your server to communicate back to client application when the transaction is completed. You may check the status of the transaction and take actions accordingly. Inside mobile applications, it is important that the user is redirected back to app whenever a transaction is completed. After the transaction is complete, Payu posts the response to the surl / furl.

sample code:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script>
var text = {key1: "data1", key2: "data2"};
var textString = JSON.stringify(text);
//PayU.onFailure(textString);
PayU.onSucces(textString);
</script>
 </head>
 <body> Failed </body>
</html>
```

Custom browser integration.

1. Download the CustomBrowser-release.aar file from github.
2. Import CustomBrowser as a module.
3. Make sure that app has a module dependency on CustomBrowser.
4. Make sure that you granted permission to read sms in your AndroidManifest.xml file.

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
```

5. Assign the layout as given below to your activity

```
<RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
       android:id="@+id/r_layout"
       android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal">
       <FrameLayout
           android:id="@+id/parent"
           android:visibility="gone"
           android:layout_alignParentBottom="true"
           android:layout_width="match_parent"
           android:layout_height="wrap_content"/>
       <WebView
           android:id="@+id/webview"
```

```
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:id="@+id/trans_overlay"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:background="@drawable/background_drawable">
        </LinearLayout>
      </RelativeLayout>
```

Note:

i. id of frame layout should be parent

ii. default visibility of frame layout should be “gone”. CustomBrowser will handle the same.

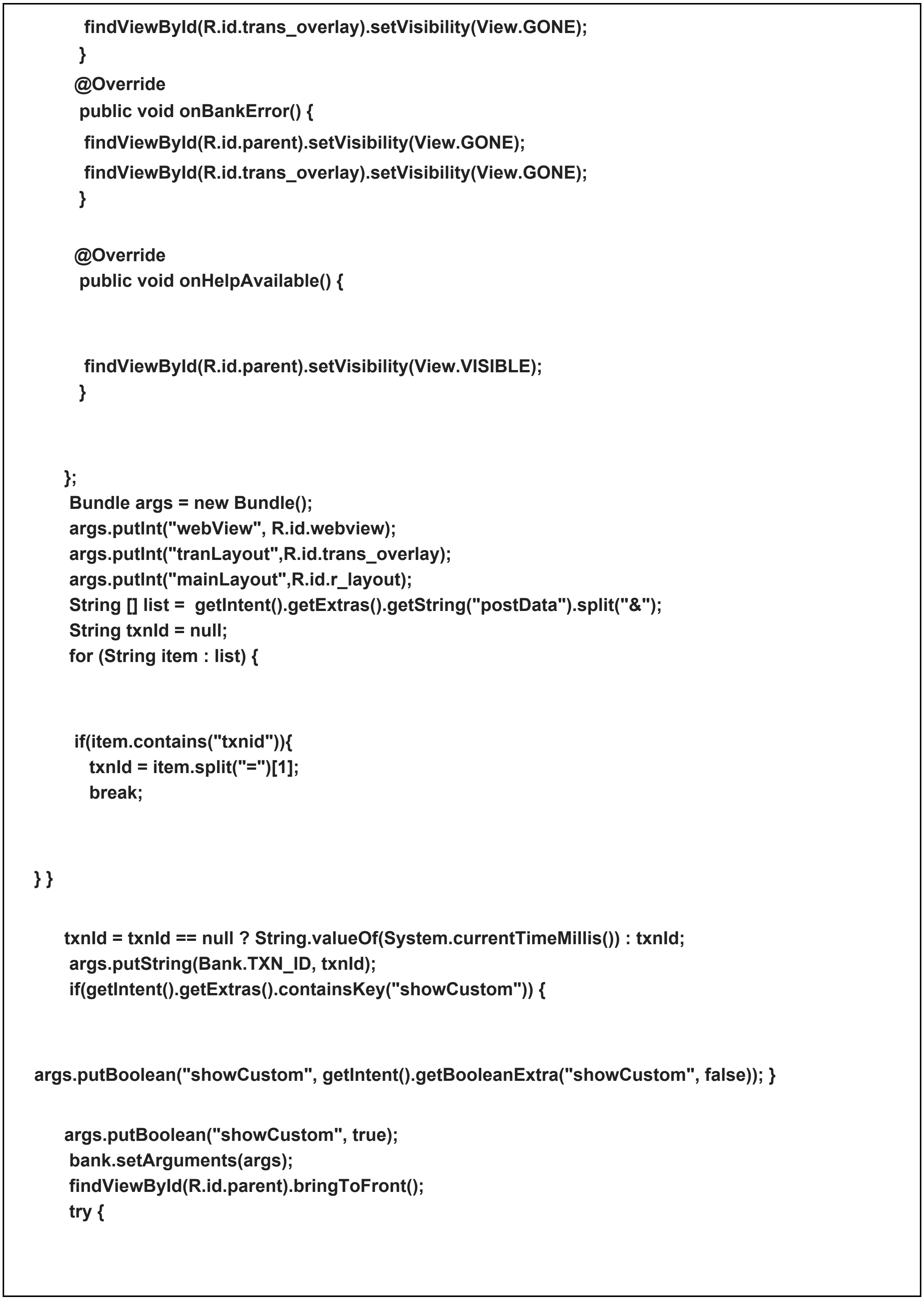
6. Make sure that your Activity is able to support Fragments

7. Inside your Activtiy onCreate() function add the following (Please make sure to change R.id.webiew to the id of your webview)

```
try {
      Class.forName("com.payu.custombrowser.Bank");
      final Bank bank = new Bank() {
@Override
public void registerBroadcast(BroadcastReceiver broadcastReceiver, IntentFilter filter) {
mReceiver = broadcastReceiver;
  registerReceiver(broadcastReceiver, filter);
        }

@Override
public void unregisterBroadcast(BroadcastReceiver broadcastReceiver) {
if(mReceiver != null){
            unregisterReceiver(mReceiver);
            mReceiver = null;
} }

        @Override
         public void onHelpUnavailable() {
          findViewById(R.id.parent).setVisibility(View.GONE);
```



```
          findViewById(R.id.trans_overlay).setVisibility(View.GONE);
         }
        @Override
         public void onBankError() {
          findViewById(R.id.parent).setVisibility(View.GONE);
          findViewById(R.id.trans_overlay).setVisibility(View.GONE);
         }

        @Override
         public void onHelpAvailable() {


          findViewById(R.id.parent).setVisibility(View.VISIBLE);
         }


      };
       Bundle args = new Bundle();
       args.putInt("webView", R.id.webview);
       args.putInt("tranLayout",R.id.trans_overlay);
       args.putInt("mainLayout",R.id.r_layout);
       String [] list =  getIntent().getExtras().getString("postData").split("&");
       String txnId = null;
       for (String item : list) {


        if(item.contains("txnid")){
           txnId = item.split("=")[1];
           break;


} }

      txnId = txnId == null ? String.valueOf(System.currentTimeMillis()) : txnId;
       args.putString(Bank.TXN_ID, txnId);
       if(getIntent().getExtras().containsKey("showCustom")) {


args.putBoolean("showCustom", getIntent().getBooleanExtra("showCustom", false)); }

      args.putBoolean("showCustom", true);
       bank.setArguments(args);
       findViewById(R.id.parent).bringToFront();
       try {
```

```
getSupportFragmentManager().beginTransaction().setCustomAnimations(R.anim.fade_in,
R.anim.cb_face_out).add(R.id.parent, bank).commit();


     }catch(Exception e)
      {



       e.printStackTrace();



finish(); }


     webView.setWebChromeClient(new PayUWebChromeClient(bank) {});










} catch (ClassNotFoundException e) {
      webView.getSettings().setSupportMultipleWindows(true);
      webView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
      webView.addJavascriptInterface(new Object() {



       @JavascriptInterface
        public void onSuccess() {



         onSuccess("");
        }



@JavascriptInterface
public void onSuccess(final String result) {


         runOnUiThread(new Runnable() {
            @Override
            public void run() {
```

```
       Intent intent = new Intent();
        intent.putExtra("result", result);
        setResult(RESULT_OK, intent);
        finish();



}
// }

}); }

    @JavascriptInterface
     public void onFailure() {


      onFailure("");
     }


@JavascriptInterface
public void onFailure(final String result) {


     runOnUiThread(new Runnable() {
       @Override
       public void run() {



        Intent intent = new Intent();
         intent.putExtra("result", result);
         setResult(RESULT_CANCELED, intent);
         finish();



} });


    }
   }, "PayU");
   webView.setWebChromeClient(new WebChromeClient() {
   });
   webView.setWebViewClient(new WebViewClient());



 }
  webView.getSettings().setJavaScriptEnabled(true);
```

```
    webView.getSettings().setDomStorageEnabled(true);
    webView.postUrl(Constants.PAYMENT_URL, EncodingUtils.getBytes(getIntent().getExtras().getString("postData"), "base64"));
```

8. Add following variable to class

private WebView webView;

private BroadcastReceiver mReceiver = null; boolean cancelTransaction = false;

For mobile related integration issues please send mail on tech@payu.in