TrustPay Library API

Version 1.0

2014-04-16

Terminology

“Client Application” - The application installed on the handset that wishes to make use of the TrustPay billing services.

“TrustPay API” - A library that is embedded in the Client Application and which provided facilities to interact with the TrustPay Backend

“TrustPay Backend” - The server infrastructure taking care of operations, billing and administration around the TrustPay service.

Theory of Operation

The TrustPay library acts as an intermediary between the client application and the consumer who wants to pay for something.

When the client app wishes to bill the consumer, the request is passed on to the TrustPay library. This library then presents a screen with all available billing methods for the amount in the requested currency based on the phone’s locality. The consumer selects a method. A method-specific screen gets displayed to the consumer to supply information and/or to validate the request. If the consumer accepts, the billing takes place and the result is returned to the client application.

Android API

There are two separate mechanisms for communicating with the Android TrustPay library and backend, depending on the purpose of the communication.

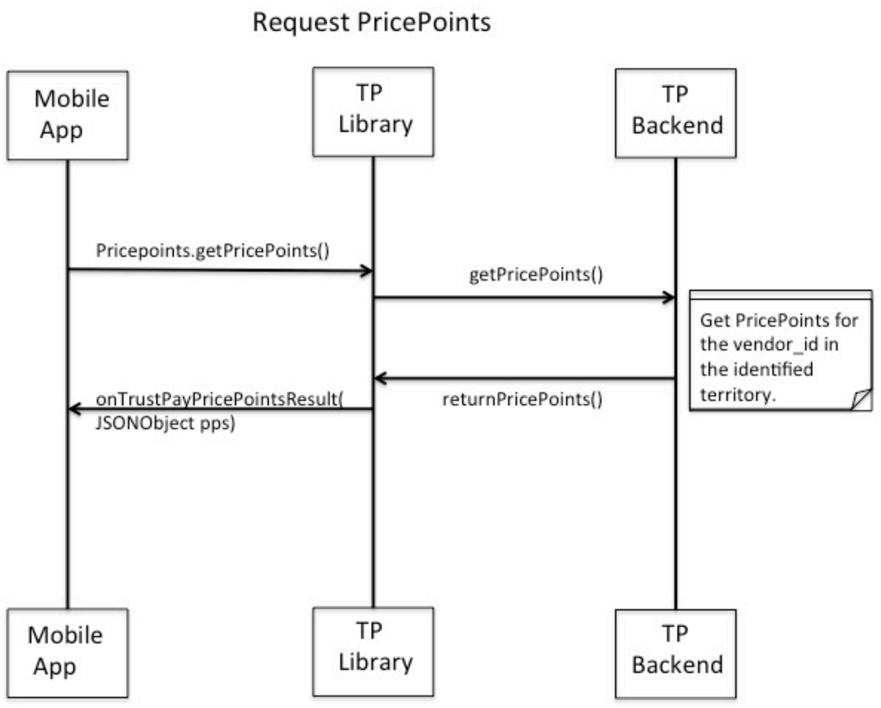
Firstly, a bound service is used to interrogate it to find out what options are available, without requiring consumer interaction.

Secondly, an Intent is launched for effect, to allow the consumer to interact with the TrustPay app to actually bill.

The overall process in the client app is managed in an Activity, which provides an entry point to launch the payment Activity from the TrustPay library. The typical scenarios are presented below.

Requesting Price Points

This function is used for the Client App to get the price points supported by TrustPay for the current handset territory and network.


Request PricePoints
Mobile App
TP Library
TP Backend
Pricepoints.getPricePoints()
getPricePoints()
Get PricePoints for the vendor_id in the identified territory.
returnPricePoints()
onTrustPayPricePointsResult(JSONObject pps)
Mobile App
TP Library
TP Backend

```
public class ApiDemo extends Activity implements PricePointListener{
/**The class must extend a Object that extends Context and implement PricePointListener*/

  private void getPricePoints(){
      Pricepoints pps = new Pricepoints("ap.d0a19fa2-f324-4df0-a4f4-
      7f3938b0e2c4",context,this);
      pps.getPricePoints();
    }

@Override
      public void onTrustPayPricePointsResult(final JSONObject pps) {
      //Do something with the returned PricePoints

      }
```

"ap.d0a19fa2-f324-4df0-a4f4-7f3938b0e2c4" is the vendor_id received on https://my.trustpay.biz when you registered your application.
context is the current application context
this is the activity that implements PricePointListener

The returned PricePoints JSON has the following format

```
 {
      "AppName": "My Demo Application",
      "Currencies": [
        {
          "upper": "22000",
          "code": "ZAR",
          "lower": "0",
          "symbol": "R",
          "isprefix": false,
          "step": "0.01",
          "name": "South African Rand"
        }
      ],
      "Country": "ZA",
      "Provider": "VISA\/MASTER",
      "Type": "CREDIT CARD"
    }
```

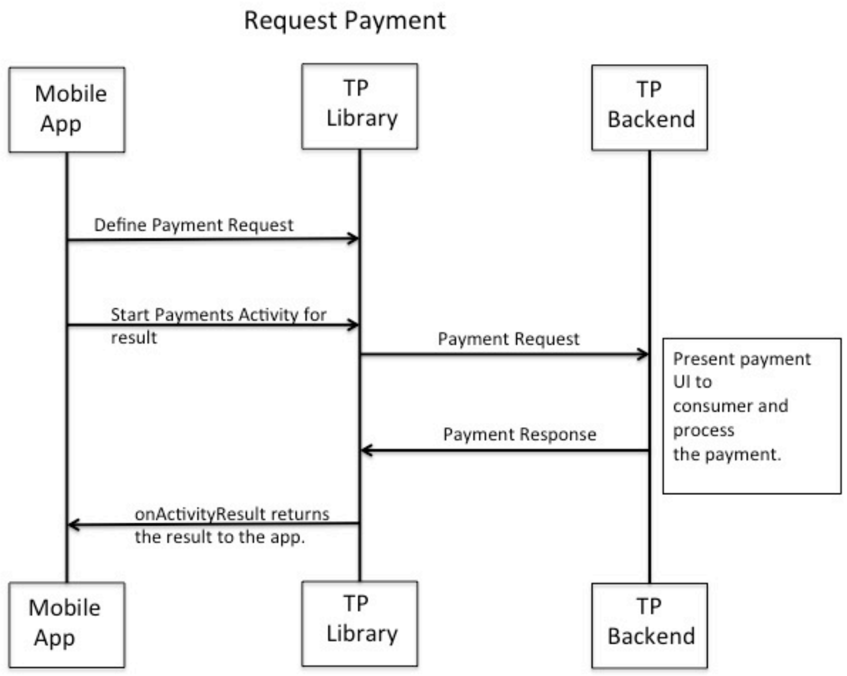
Per payment method,

where: **code** is always populated with the currency code.
lower is the lower value of the pricepoint range
upper is the upper value of the pricepoint range
step is the increment values between lower and upper
e.g.: for all full rand values between 20 and 99, the values will be:
lower = “20.00” upper = “99.00” step = “1.00”
for all cent values between 1 and 100 rand, the values will be:
lower = “1.00” upper =”100.00” step = ”0.01”
for a single pricepoint of 5 rand, the values will be:
lower = “5.00” upper = ”5.00” step = “0.00”
symbol is the symbol used to display the currency.
isprefix “true” means the symbol goes before the abount and “false” after.
name is the fullname of the currency.
country is the iso3166 code in which the phone identified itself as.
provider is the payment type provider.
type is the transaction type, values currently supported are:

CARRIER BILLING
VOUCHER
CREDIT CARD
FNB
CASHU
PAGA
MPESA

Request Payment

This function is used to actually bill the consumer. It does this by presenting a screen explaining the amount to be billed and the name of the application requesting it. The consumer can either agree to bill, or cancel the transaction. The status is returned to the calling app.



Request Payment
Mobile App
TP Library
TP Backend
Define Payment Request
Start Payments Activity for result
Payment Request
Present payment UI to consumer and process the payment.
Payment Response
onActivityResult returns the result to the app.
Mobile App
TP Library
TP Backend

```
public void pay() {
            Request request = new Request();
            request.setAmount("100.00");
            request.setApp_id("ap.d0a19fa2-f324-4df0-a4f4-7f3938b0e2c4");
            request.setApp_ref(myTransactionId);
            request.setApp_user("Demo User");
            request.setCurrency("ZAR");
            request.setTx_description("Buying credits for the demo");
            request.setIstest(false);
            Intent intent = new Intent(this, Payments.class);
            Bundle bundle = new Bundle();
            bundle.putSerializable("request", request);
            intent.putExtras(bundle);
            startActivityForResult(intent, 1);
            }
```

Where : "ap.d0a19fa2-f324-4df0-a4f4-7f3938b0e2c4" is the vendor_id received on https://my.trustpay.biz when you registered your application. myTransactionId is a String that will identify the transaction from a reporting perspective.

```
      // This method is called when the billing is done
  @Override
      protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
            if (requestCode == 1) {
                  if (resultCode == RESULT_OK) {
                        //Successful Transaction

                  } else if (resultCode == Activity.RESULT_CANCELED) {
                        //Cancelled Transaction

                    } else {
                        //Strange result … should never happen

                    }
                } else {
                    //Unknown Result Code
                }
            }
```

You will need the following lines added to the AndroidManifest.xml file:

```
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<activity android:name="biz.trustpay.ui.Payments"
        android:screenOrientation="portrait" >
</activity>
```

Changes

Date	Version	Changes
2014-04-16	1.0	Document created