

Tecnológico de Costa Rica

Laboratorio - Integración con PayPal

Desarrollo de aplicaciones móviles



Rodolfo Artavia López, Daniel Montoya Aguilar
2-11-2016

Integración de PayPal a una aplicación Android.

Parte I – PayPal.

1. Antes de empezar con Android, si no tiene cuenta en PayPal debe crear una. Para eso siga el siguiente link.

<https://www.paypal.com/signup/account?intent=true&intentId=intent1478137709237>

Regístrese gratis.

Cuenta Personal

Envíe o reciba pagos en línea a nivel internacional. Todo ello, sin compartir sus datos personales y financieros.

Costa Rica

▼

Correo electrónico

Cree su contraseña

Confirme su contraseña

Continuar

2. Luego de crear la cuenta en PayPal, necesitamos una cuenta de desarrollador, la creamos en el siguiente link. <https://developer.paypal.com/developer/accounts/create>

Create Test Account

A personal or business test account can also be created on sandbox.paypal.com.

Account Details

Country

United States

▼

Account Type

- ☒ Personal (Buyer Account)
- ☐ Business (Merchant Account)

Email Address

Password (8-20 characters)

- Ahora se debe crear una cuenta para la aplicación, esta cuenta es la que se asociará la aplicación a desarrollar. Para crear la aplicación y su ID debe seguir el siguiente link
<https://developer.paypal.com/developer/applications/create>

Create an app to receive REST API credentials for testing and live transactions.

Note Features available for live transactions are listed in your [account eligibility](#).

Application Details

App Name

PayPalIntegration


Sandbox developer account

As a reminder, all apps created under your account should be related to your business and the type of business it conducts.

By clicking the button below, you agree to [PayPal Developer Agreement](#).

Create App

- Quando se crear la aplicación se generan las credenciales que se usaran dentro del app en Android, en la siguiente imagen el Client ID (marcado en rojo) es lo que se usará en Android Studio.

App display name: TestApp 

SANDBOX API CREDENTIALS

Sandbox account

@gmail.com

Client ID

AXywwifZCytX-h2aF14UTj26B5jReCeweJchQPCfXbhr7VRcERE2aVuBZyoGuEPV1y

Secret

Show

- Ahora ya se tiene todo preparado para empezar con aplicación en Android Studio.

Parte II – Android Studio.

1. Creamos un proyecto con un empty activity en Android Studio, y lo llamaremos PayPalIntegration.
2. Creamos una clase que se llamará PayPalConfig.java. Esta clase tendrá el Client-ID que generamos cuando creamos la aplicación en PayPal Developers. El valor de la variable `PAYPAL_CLIENT_ID` la cambiamos por nuestro Client ID.

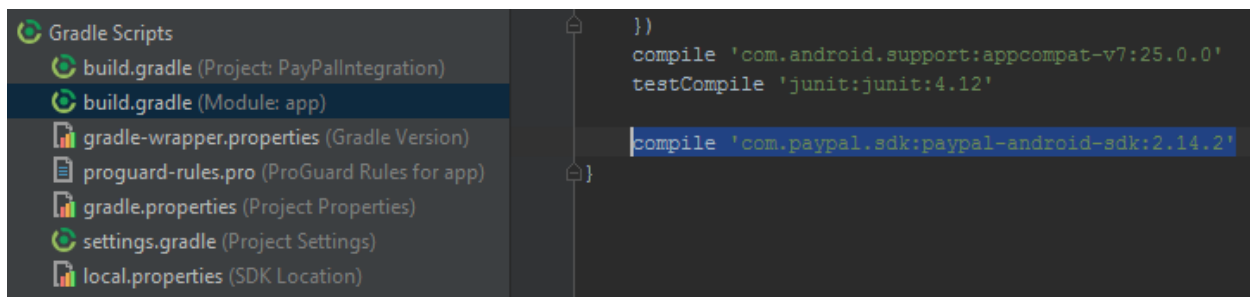
```
package com.example.lily.paypalintegration;

/**
 * Created by Rodolfo on 02/11/2016.
 */

public class PayPalConfig {
    public static final String PAYPAL_CLIENT_ID =
"AU4wxwyuyLYHwmbrytQcfcq53MGQZ14tQFImJNxrTpaAy7KioymmlSnfZJNVO-
CBkwLpEKSUJdKrqn1";
}
```

3. Ahora en el build.gradle del app, agregamos el SDK de PayPal.

```
compile 'com.paypal.sdk:paypal-android-sdk:2.14.2'
```



4. Sincronizamos el build.gradle.

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

[Sync Now](#)

5. Una vez finalizada la sincronización, vamos al **activity_main.xml** y agregamos un **TextView** para el título, un **EditText** para capturar cuánto dinero vamos a pagar y un **Button** que iniciará el proceso de pago.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.lily.paypalintegration.MainActivity">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Enter Amount"
            android:textAlignment="center" />

        <EditText
            android:id="@+id/editTextAmount"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textAlignment="center" />

        <Button
            android:id="@+id/buttonPay"
            android:text="Pay Now"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </LinearLayout>
</RelativeLayout>
```

6. Creamos una empty activity que llamaremos ConfirmationActivity. Esta actividad desplegará si el proceso de pago fue exitoso o no.
7. En el activity_confirmation.xml generado, agregamos las siguientes estructuras.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_confirmation"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.lily.paypalintegration.ConfirmationActivity">
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">
        <TableRow>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Payment Amount : " />
            <TextView
                android:id="@+id/paymentAmount"
                android:layout_width="wrap_content"
                android:textStyle="bold"
                android:layout_height="wrap_content" />
        </TableRow>
        <TableRow>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Status : " />
            <TextView
                android:id="@+id/paymentStatus"
                android:layout_width="wrap_content"
                android:textStyle="bold"
                android:layout_height="wrap_content" />
        </TableRow>
        <TableRow>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Payment Id : " />
            <TextView
                android:id="@+id/paymentId"
                android:textStyle="bold"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </TableRow>
    </TableLayout>
```

```
</TableLayout>
</RelativeLayout>
```

8. Volviendo al MainActivity, agregamos las siguientes variables además de agregar que implemente el View.OnClickListener.

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private Button buttonPay;
    private EditText editTextAmount;
    private String paymentAmount;

    public static final int PAYPAL_REQUEST_CODE = 123;
    //Aquí se configura el ambiente de paypal, se le pasa el ID del proyecto
    // se especifica que es un ambiente de prueba (ENVIRONMENT_SANDBOX)
    private static PayPalConfiguration config = new PayPalConfiguration()
        .environment(PayPalConfiguration.ENVIRONMENT_SANDBOX)
        .clientId(PayPalConfig.PAYPAL_CLIENT_ID);
}
```

9. El onCreate, debe contener lo siguiente. Aquí referenciamos las variables del botón y el campo de texto a su correspondiente en la interfaz. Luego con el intent iniciamos el Servicio de PayPal.

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

    buttonPay = (Button) findViewById(R.id.buttonPay);
    editTextAmount = (EditText) findViewById(R.id.editTextAmount);

    buttonPay.setOnClickListener(this);

    Intent intent = new Intent(this, PayPalService.class);
    intent.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION, config);

    startService(intent);
}
```

10. Agregamos en onClick.

```
@Override
public void onClick(View v) {
    getPayment();
}
```

11. Y luego el onDestroy para que cuando se cierre el app cierre el servicio de PayPal también.

```
@Override
public void onDestroy() {
    stopService(new Intent(this, PayPalService.class));
    super.onDestroy();
}
```

12. Agregamos el metodo `getPayment` que es llamado en el `onClick`. Este método es el que se encarga de inicializar las actividades propias de PayPal para el pago.

```
private void getPayment() {
    //Tomamos la cantidad que se digito en el campo de texto editText
    paymentAmount = editTextAmount.getText().toString();
    //Creamos el paypalpayment
    PayPalPayment payment = new PayPalPayment(new
BigDecimal(String.valueOf(paymentAmount)), "USD", "Simplified Coding Fee",
PayPalPayment.PAYMENT_INTENT_SALE);
    //Creamos el intent activity para el Paypal Payment
    Intent intent = new Intent(this, PaymentActivity.class);
    //Colocamos la configuracion en el intent
    intent.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION, config);
    //Agregamos el payment al intent
    intent.putExtra(PaymentActivity.EXTRA_PAYMENT, payment);
    //Iniciamos el intent activity
    startActivityForResult(intent, PAYPAL_REQUEST_CODE);
}
```

13. Por ultimo para el `MainActivity` agregamos un `onActivityResult`.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    //Si el resultado es de PayPal
    if (requestCode == PAYPAL_REQUEST_CODE) {
        //si el resultado es OK
        if (resultCode == Activity.RESULT_OK) {
            PaymentConfirmation confirm =
data.getParcelableExtra(PaymentActivity.EXTRA_RESULT_CONFIRMATION);
            if (confirm != null) {
                try {
                    //Tomamos los detalles del pago realizado
                    String paymentDetails =
confirm.toJSONString(4);
                    Log.i("paymentExample", paymentDetails);
                    //Iniciamos la actividad para mostrar los detalles del
pago realizado
                    startActivity(new Intent(this,
ConfirmationActivity.class)
                        .putExtra("PaymentDetails", paymentDetails)
                        .putExtra("PaymentAmount", paymentAmount));
                } catch (JSONException e) {
                    Log.e("paymentExample", "an extremely unlikely failure
occurred: ", e);
                }
            }
        } else if (resultCode == Activity.RESULT_CANCELED) {
            Log.i("paymentExample", "The user canceled.");
        } else if (resultCode == PaymentActivity.RESULT_EXTRAS_INVALID) {
            Log.i("paymentExample", "An invalid Payment or
PayPalConfiguration was submitted. Please see the docs.");
        }
    }
}
```


14. Ahora en el ConfirmationActivity, el onCreate lo modificamos para que tome el Json que se le pasó en el putExtra y lo despliegue.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_confirmation);
    //Getting Intent
    Intent intent = getIntent();
    try {
        JSONObject jsonDetails = new
JSONObject(intent.getStringExtra("PaymentDetails"));

        //desplegamos los detalles del pago realizado
        showDetails(jsonDetails.getJSONObject("response"),
intent.getStringExtra("PaymentAmount"));
    } catch (JSONException e) {
        Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
    }
}
```

15. Para desplegarlo creamos el metodo showDetails.

```
private void showDetails(JSONObject jsonDetails, String paymentAmount) throws
JSONException {
    //Views
    TextView textViewId = (TextView) findViewById(R.id.paymentId);
    TextView textViewStatus= (TextView) findViewById(R.id.paymentStatus);
    TextView textViewAmount = (TextView) findViewById(R.id.paymentAmount);

    //Showing the details from json object
    textViewId.setText(jsonDetails.getString("id"));
    textViewStatus.setText(jsonDetails.getString("state"));
    textViewAmount.setText(paymentAmount+" USD");
}
```

Resultado.

Cuando corremos el App, digitamos la cantidad a pagar y presionamos el botón de pagar. Si todo salió bien debería de desplegarse una pantalla como la siguiente, la cual nos permite hacer el pago por medio de PayPal o por medio de una tarjeta.

