

HCIM Summer Workshop

Guião Hands-on

Exemplo de Aplicação Facebook

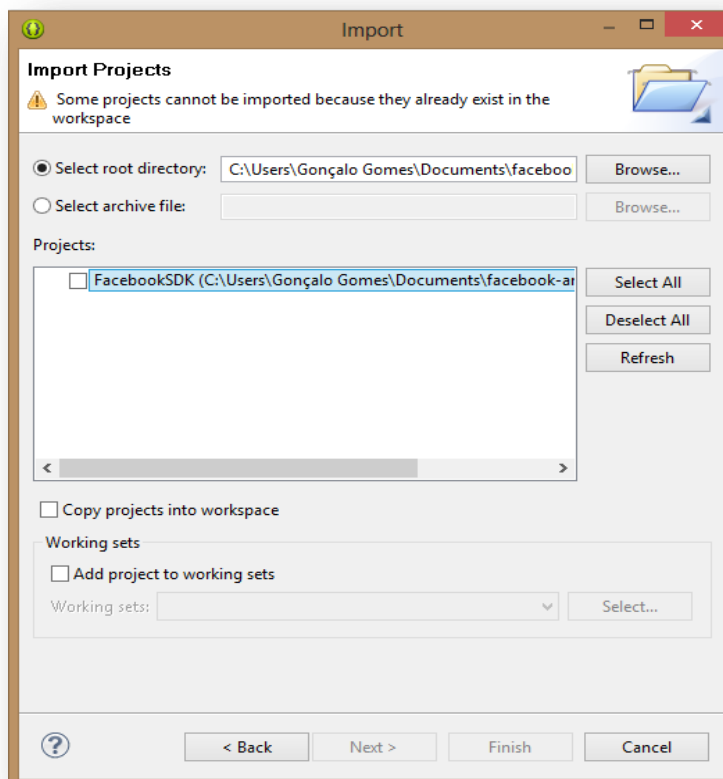
1. Download do Software Development Kit (SDK)

O download do SDK do Facebook encontra-se disponível [aqui](#).

Depois de fazer o download do ficheiro, extraia para uma pasta que seja facilmente reaccessível.

2. Importar para o IDE

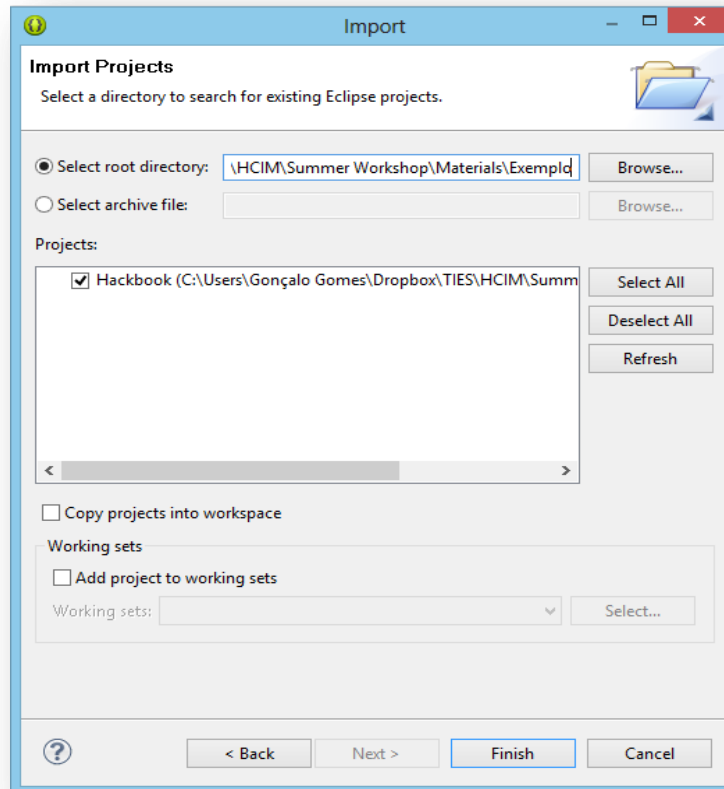
O ficheiro contém uma directoria `facebook`, a qual contém o SDK em si. Para importar o SDK para o seu IDE (Eclipse, no caso), vá a 'File' > 'Import' e no menu escolha 'General' > 'Existing Projects into Workspace' e importe essa pasta:



De seguida carregue em 'Finish' para dar o processo como concluído e verá como o SDK é adicionado como um novo "projecto" à sua biblioteca (lado direito do IDE).

3. Importar o exemplo

Para importar o exemplo de projecto que usa o SDK do Facebook, siga o exemplo análogo ao usado para importar o Facebook SDK no ponto anterior. Mais uma vez vá a 'File' > 'Import' e no menu escolha 'General' > 'Existing Projects into Workspace' e importe o projecto que se encontra neste [link](#) com o nome de `Exemplo`:



4. Pronto a correr!

Finalmente o exemplo está pronto a correr, com três funcionalidades básicas de experiência: Postar no Mural, Enviar Pedidos de Aplicação e Ver os Amigos.

A Sua Aplicação Facebook

1. Registrar Aplicação no Facebook Developers

[Clique](#) para aceder ao Facebook Devs

Carregue em **Criar nova aplicação**. De seguida dê-lhe um **nome**, **namespace** e **categoria** à sua escolha:

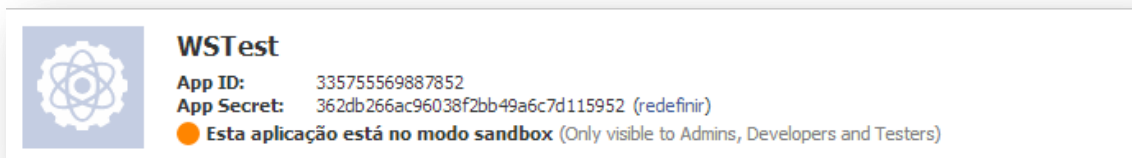


The screenshot shows the 'Criar aplicação nova' (Create new application) form in the Facebook Developers console. The form includes the following fields and options:

- Nome da aplicação:** Input field containing 'WSTest', with a 'Valid' status indicator.
- Namespace da aplicação:** Input field containing 'wstest_namespace', with a 'Disponível' (Available) status indicator.
- Categoria da aplicação:** A dropdown menu currently set to 'Comunicação' (Communication), with a subcategory selection dropdown next to it.
- Alojamento web:** A checkbox labeled 'Sim, gostaria de receber alojamento web grátis fornecido pela Heroku (Learn More)'. The checkbox is currently unchecked.

At the bottom of the form, there is a disclaimer: 'Ao prosseguires, estás a concordar com as Políticas da Plataforma do Facebook'. To the right of the disclaimer are two buttons: 'Continuar' (Continue) and 'Cancelar' (Cancel).

Uma vez criada, note que a aplicação possui um ID associado, visível no campo **App ID**:



The screenshot shows the application overview card for 'WSTest' in the Facebook Developers console. It includes the following information:

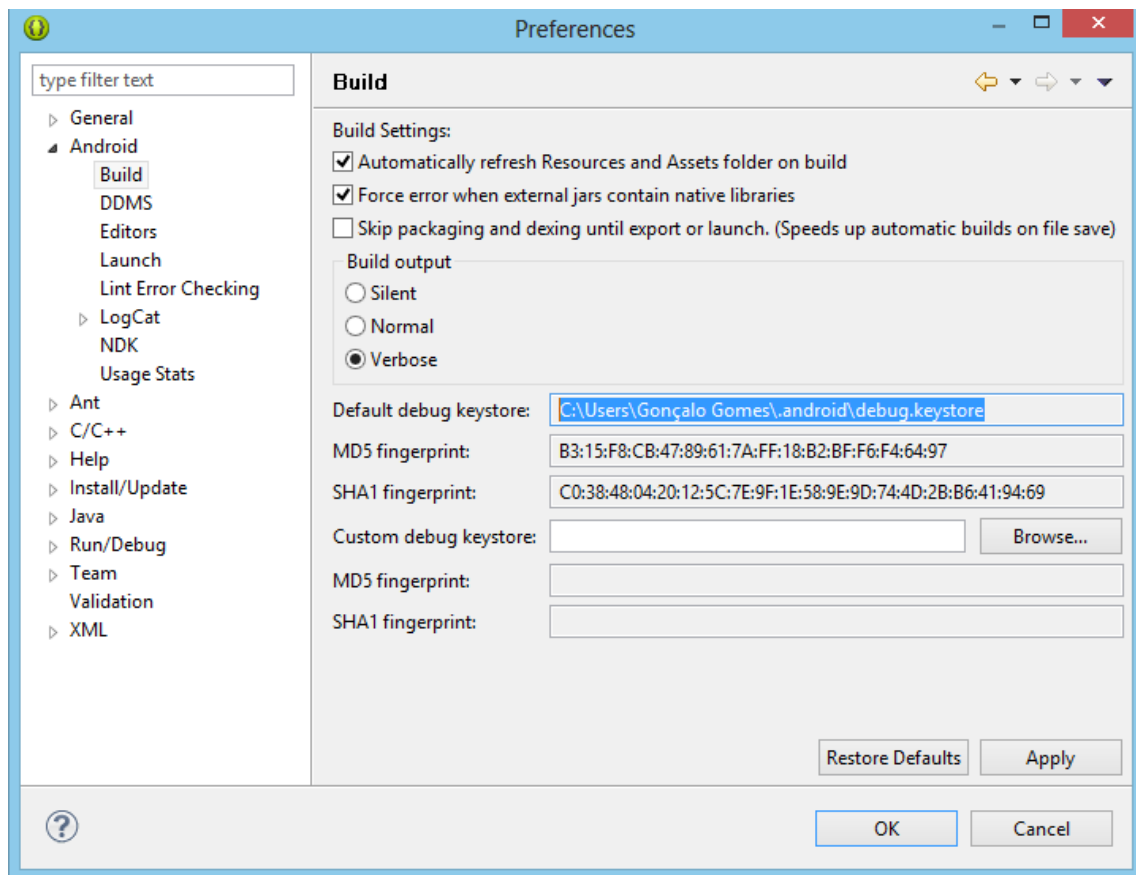
- App ID:** 335755569887852
- App Secret:** 362db266ac96038f2bb49a6c7d115952 (redefinir)
- Status:** Esta aplicação está no modo sandbox (Only visible to Admins, Developers and Testers)

Este ID é único e que faz a associação entre a aplicação e a chave única gerada para cada um delas. Para obter tal chave, é necessário que seja instalado o **OpenSSL**, o qual pode ser obtido [aqui](#). Após instalado o OpenSSL, deve-se abrir uma linha de comandos e correr o seguinte comando:

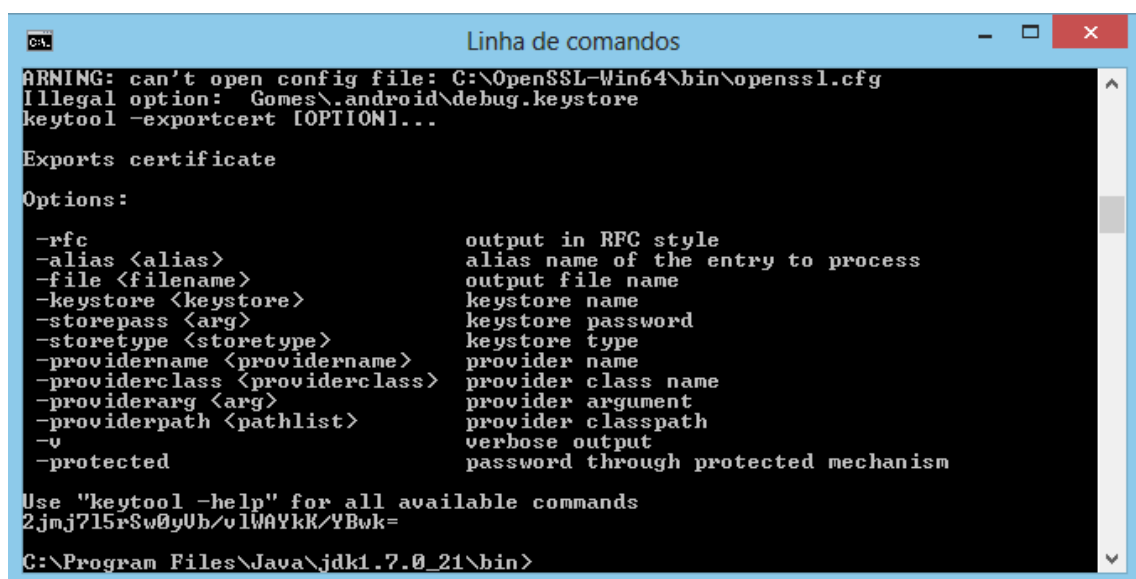
```
keytool -exportcert -alias <ALIAS_QUALQUER> -keystore <CHAVE_KEYSTORE> | openssl sha1 -binary | openssl base64
```

Nota importante: O comando keytool está tipicamente contido na pasta do **Java/jdk/bin/** ou **Java/jdk/bin/**, pelo que devem colocar o caminho antes do comando. O mesmo deve ser feito para o comando **OpenSSL**.

A chave da keystore é obtida em 'Window' > 'Preferences' > Tab 'Android' > Subtab 'Build' > Campo 'Default debug keystore':



Após isto, o resultado de correr o comando na linha de comandos deverá ser algo como:



Como se pode ver, é gerada então uma chave com MD5, para que possamos certificar a nossa aplicação e garantir a integridade e unicidade da mesma.

De seguida voltamos ao *lobby* da aplicação recém-criada do Facebook Developers e colocamos a chave gerada no campo “**Key Hashes**” e activamos o campo que diz “**Início de sessão no Facebook**”. Guardam-se a alterações efectuadas, posteriormente.

<input checked="" type="checkbox"/>	Site com início de sessão pelo Facebook	Log in to my website using Facebook.
<input checked="" type="checkbox"/>	Aplicação no Facebook	Use my app inside Facebook.com.
<input checked="" type="checkbox"/>	Internet no telemóvel	Bookmark my web app on Facebook mobile.
<input checked="" type="checkbox"/>	Aplicação nativa para iOS	Publish from my iOS app to Facebook.
<input checked="" type="checkbox"/>	Aplicação nativa para Android	<input type="text"/> <input type="text"/> Key Hashes: [?] <input type="text" value="2jnj7l5rSw0yVb/vlWAYkK/YBwk= x"/>
	Início de sessão no Facebook: [?]	<input checked="" type="radio"/> Ativar <input type="radio"/> Desativar
	Deep Linking: [?]	<input type="radio"/> Ativar <input checked="" type="radio"/> Desativar
<input checked="" type="checkbox"/>	Page Tab	Build a custom tab for Facebook Pages.

[Guardar alterações](#)

Nota: Para além disso, será necessário voltar a aceder a este ecrã, no qual precisaremos de registar o nome do *package* principal da aplicação registada.

2. Criar uma aplicação que usa o Facebook Login

Agora é tempo de criarmos a aplicação que usará o Login do Facebook. Para tal, no Eclipse clicamos em “File” > “New” > “Project” > “New Android Application”:

New Android Application

The application name for most apps begins with an uppercase letter

Application Name: facebook_test

Project Name: facebook_test

Package Name: com.example.facebook_test

Minimum Required SDK: API 10: Android 2.3.3 (Gingerbread)

Target SDK: API 10: Android 2.3.3 (Gingerbread)

Compile With: Google APIs (Google Inc.) (API 10)

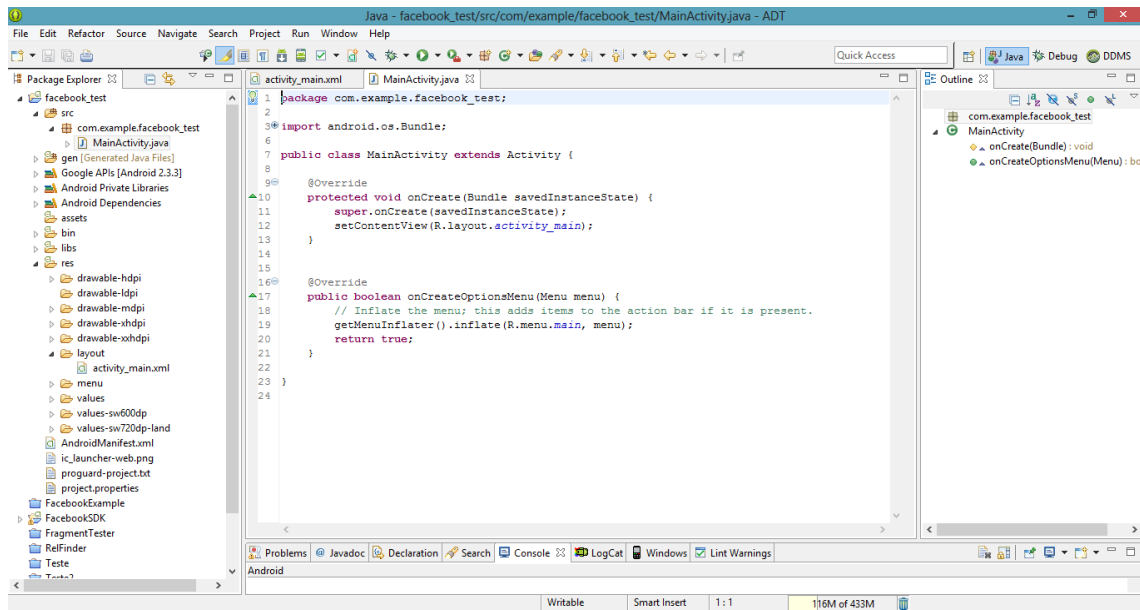
Theme: None

Choose the base theme to use for the application

< Back Next > Finish Cancel

Nota: Tendo em conta que grande parte dos dispositivos que possuímos correm a versão do 2.3.3 do Android, configuraremos o SDK para correr com tais definições.

Siga as instruções até criar uma actividade simples com o nome que desejar:



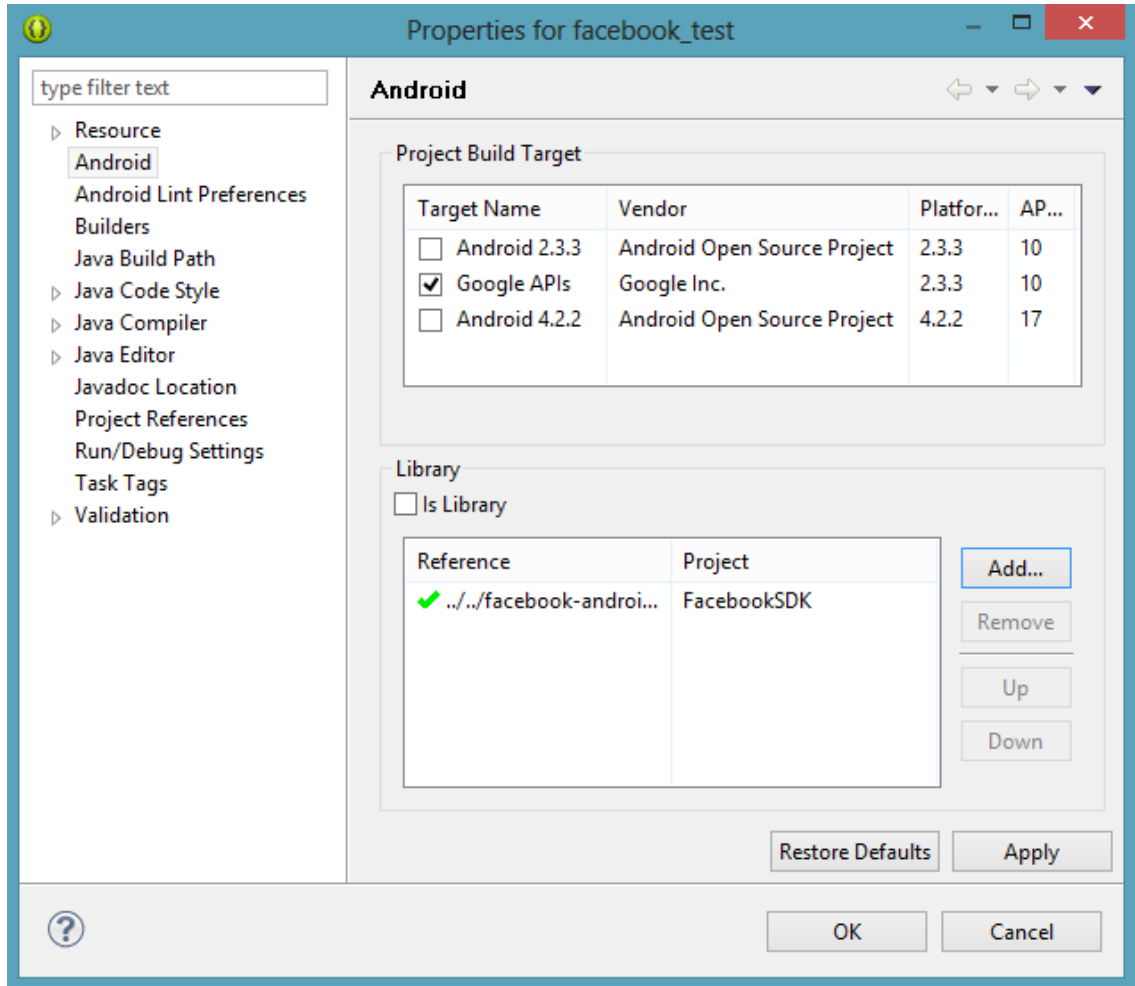
Agora sim, voltamos ao *lobby* da nossa actividade no Facebook Developers e **registamos a aplicação, nome da classe principal (que corre o Login) e o nome do package**:

✓ Aplicação nativa para Android	
Nome do pacote: [?]	com.example.facebook_test
Nome da cadeira: [?]	com.example.facebook_test.MainActivity
Key Hashes: [?]	2jnj715rSw0yVb/vlWAYkK/YBwk= x
Início de sessão no Facebook: [?]	<input checked="" type="radio"/> Ativar <input type="radio"/> Desativar
Deep Linking: [?]	<input type="radio"/> Ativar <input checked="" type="radio"/> Desativar
✓ Site com início de sessão pelo Facebook	Log in to my website using Facebook.
✓ Aplicação no Facebook	Use my app inside Facebook.com.
✓ Internet no telemóvel	Bookmark my web app on Facebook mobile.
✓ Aplicação nativa para iOS	Publish from my iOS app to Facebook.
✓ Page Tab	Build a custom tab for Facebook Pages.
Guardar alterações	

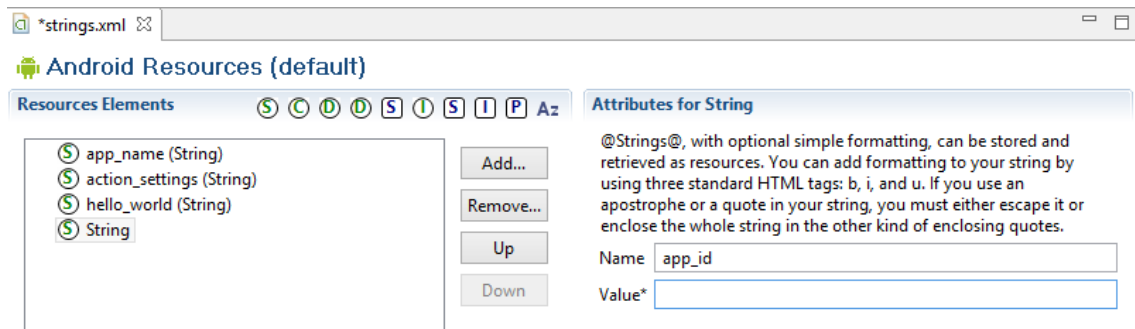
Caso se precise de verificar qual o nome do *package* com mais precisão, pode-se ainda consultar o **Manifest.xml** da aplicação, e consultar a *tag* “package”.

3. Usar o SDK como biblioteca da Aplicação

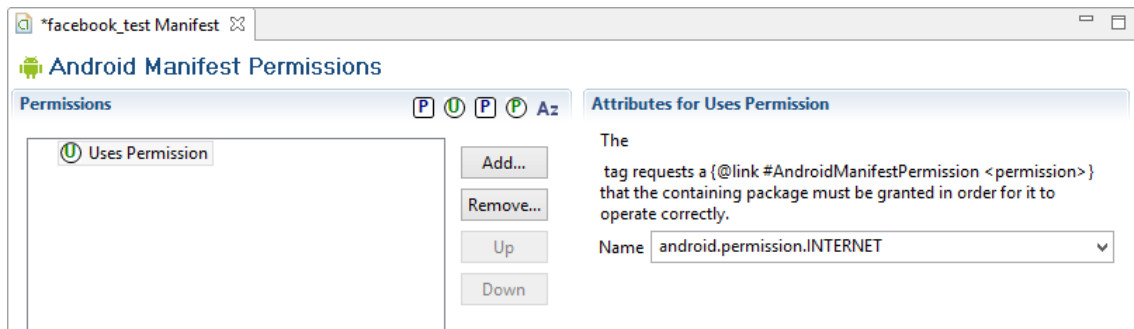
Temos então de ligar a aplicação ao SDK, fazendo uso deste último como biblioteca, podendo assim usar todas as chamadas de métodos do Facebook. Para tal, carregamos em “Project” > “Properties” > Tab “Android” > “Library” > “Add” e escolhemos a pasta do SDK usado no primeiro projecto:



Depois, vamos ao nosso ficheiro **res\strings** e adicionamos uma string que irá conter o ID da aplicação, para finalmente ligarmos com a chave gerada ainda há pouco no ponto anterior:



Depois disto, temos de dar acesso à Internet à nossa aplicação e para isso temos de definir no nosso **Manifest.xml** que vamos requisitar este serviço:



Finalmente temos de criar *metadata* para o ID da aplicação, para início de sessão. Adicionalmente temos de declarar a classe **com.facebook.LoginActivity** no nosso **Manifest.xml**, portanto neste adicionamos o seguinte:

```
<activity android:name="com.facebook.LoginActivity" ></activity>
```

e

```
<meta-data  
    android:name="com.facebook.sdk.ApplicationId"  
    android:value="@string/app_id" />
```

dentro da tag “**application**”

4. Implementar a aplicação

1. Começamos então a implementar a aplicação, colocando na mesma um objecto que sirva de *Label* (e.g. `TextView`). Esta *Label* servirá para “dar as boas vindas” ao utilizador logado.
2. Uma parte importante do SDK é a classe **Session**, a qual faz a gestão da autenticação e autorização dos utilizadores do Facebook. Tendo em conta que se quer fazer *login* e *logout* na aplicação, deve-se implementar um mecanismo que permita fazer isso e para tal, adiciona-se o seguinte código à classe:

```
@Override  
  
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
  
    super.onActivityResult(requestCode, resultCode, data);  
  
    Session.getActiveSession().onActivityResult(this, requestCode, resultCode,  
data);  
  
}
```


- De seguida, precisamos de fazer com que o utilizador seja logado quando a actividade se inicia. Para isso, precisamos de implementar o método **openActiveSession** quando a actividade começa (**onCreate()**):

```
// start Facebook Login

Session.openActiveSession(this, true, new Session.StatusCallback() {

    // callback when session changes state

    @Override

    public void call(Session session, SessionState state, Exception

exception) {

        }

});
```

- Como este método irá estar sempre a ser chamado (porque funciona com *callback*), nós só queremos que aconteça isso enquanto não houver uma sessão activa, então adicionamos a seguinte condição:

```
if (session.isOpened()) {

}
```

5. Finalmente, queremos fazer um pedido ao Facebook que nos retorne o nosso nome. Para que tal aconteça, fazemos um pedido assíncrono (a actividade continua a correr) ao Facebook, para que nos retorne informação sobre **nós** e filtramos o que queremos, no caso, o nosso nome:

```
// make request to the /me API

Request.executeMeRequestAsync(session, new Request.GraphUserCallback() {

    // callback after Graph API response with user object

    @Override

    public void onCompleted(GraphUser user, Response response) {

    }

});
```

Nota: O objecto GraphUser tem encapsulada informação relevante acerca do utilizador.

Desafio 1:

Apresentar a data de nascimento do utilizador, bem como o primeiro e último nomes (separados)

Desafio 2:

Apresentar a localização actual do utilizador e a sua terra natal

Desafio 3:

Obter o sitio onde o utilizador trabalha