

Admob integration lib and other options

Configuring your project

Step 1:

After the manifest tag (<manifest>) add this permissions:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Step 2:

In the application tag add this line :

```
<application
    android:name="com.google.admob.integration.Libs.AdMob"
    android:allowBackup="true"
```

Like this:

```
<application
    android:name="com.google.admob.integration.Libs.AdMob"
    android:allowBackup="true"
```

Step 3 :

Before </application> add this lines :

```
<!-- ads -->
    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
    <activity
        android:name="com.amazon.device.ads.AdActivity"

android:configChanges="keyboardHidden/orientation/screenSize" />
    <!-- Include the AdActivity configChanges and theme. -->
    <activity
        android:name="com.google.android.gms.ads.AdActivity"

android:configChanges="keyboard/keyboardHidden/orientation/screenLayout/uiMode/screenSize/smallestScreenSize"
        android:theme="@android:style/Theme.Translucent" />
```

Step 4 :

Configuring your unit add for admob banner, admob interstitial, analytics tacker_id, facebook and google play developer name

Open your strings.xml file, and add this line

```
<string name="admob_test">1</string>
<string name="admob_banner">admob_banner_code</string>
<string name="admob_interstitial">admob_interstitial_code</string>
<string name="google_analytics">google_tracker_id</string>
<string name="facebook_id">facebook_page_id</string>
<string
name="user_play_name">your_google_play_developer_name</string>
```

How you can you this line : `<string name="admob_test">1</string>` ?

When the value of admob_test is 1, you can use admob in test mode, so you do not have to wear about clicking ads

When you want to export your application for production don't forget to change the value to 0 or to delete the hall line

Now your project is ready to use admob and to use other option of this lib

Using interstitial:

Call interstitial just one time

When you want to show an interstitial ad, you have just to call one instruction

```
Admob.get().showInterstitial();
```

Using interstitial with timer:

You can use this when you have an app with article to read or with picture to view...

If you want to show the interstitial every 10 seconds for example, just put the time you want like this:

```
Admob.get().showInterstitial(10);
```

Every 30 seconds:

```
Admob.get().showInterstitial(30);
```

In this case, don't forget to call `Admob.get().closeInterstitial();` when you leaving your activity, like this :

```
@Override
protected void onStop() {
    Admob.get().closeInterstitial();
    super.onStop();
}
```

Using interstitial after N time :

If you want to show interstitial after a number of action executed by user, just call `showInterstitialNTime`

Example: showing the interstitial every 3 time

```
Admob.get().showInterstitialNTime(3);
```

Add Banner to your code

Now you do not have to use the xml admob ads banner declaration in your xml file, just in your code, you can add admob banner dynamically

First reach to area where you want to add you admob banner, and give him an id, the preference this area have to be a LinearLayout, so you can specify exactly where you want to add the banner, in the first place or in the last one

Then reach the id of your View and use it in this code instead of `LLMainApp`

For a Smart banner, use this :

```
LinearLayout ll = (LinearLayout) findViewById(R.id.LLMainApp);  
ll.addView(Admob.get().skSmartBanner(), 0);
```

For a normal banner use this:

```
LinearLayout ll = (LinearLayout) findViewById(R.id.LLMainApp);  
ll.addView(Admob.get().skBanner(), 0);
```

Adding exit option

In your main Activity (you will find it on the <application> in your manifest) add this code:

If you want all options, just add this one:

```
@Override
public void onBackPressed() {
    Admob.get().exitWithOption(this);
}
```

If you want to choose just some options or make options on an order that you prefer, use this code, you comment or delete option that you don't want:

```
@Override
public void onBackPressed() {

    List<String> choice = new ArrayList<String>();

    choice.add(Constants.RATE_APP);
    choice.add(Constants.LIST_APP);
    choice.add(Constants.FACEBOOK);
    choice.add(Constants.SHARE_APP);
    choice.add(Constants.RETURN_TO_APP);
    choice.add(Constants.EXIT);

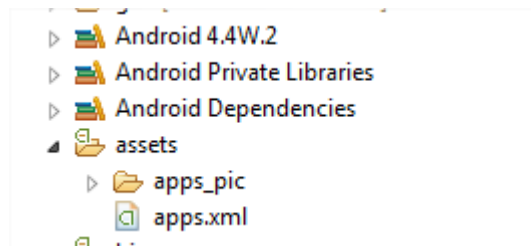
    Admob.get().exitWithOption(this, choice);
}
```

If you want just showing rate option, using this code:

```
@Override
public void onBackPressed() {
    Admob.get().rateApp(this);
}
```

Creating your app list

To create your app list to show them to the user, copy file and folder from the folder assets you will find in the config folder coming with this document to your asset folder like this:



Open the **apps.xml** file, and edit it by adding your applications

For each application you have to specify some informations :

name	The name of your application
description	A short description of your application
package	The package name of your application
icon	An icon of your application, you have to put an icon in the assets/apps_pic for each app, and put the nom of the picture in this tag