AdsNaTa Android SDK Integration Guide 4.1.6

The following document provides detailed instructions on how to integrate the AdsNaTa Android SDK 4.1.6 into your Android Projects based on a simple Demo app. The AdsNaTa SDK is capable of displaying both traditional Banner Ads and (Rich Media) Interstitials.

Step 1: Set Up your Application in your AdsNaTa Control Panel

- Log into your AdsNaTa Administration Control Panel
- Click on Inventory>Add Publications if you'd like to create a new Publication
- If you want to integrate an existing Publication, navigate to Inventory>Integration
- Once done you will be provided with a unique Placement ID (Publisher ID in code) and a Request URL

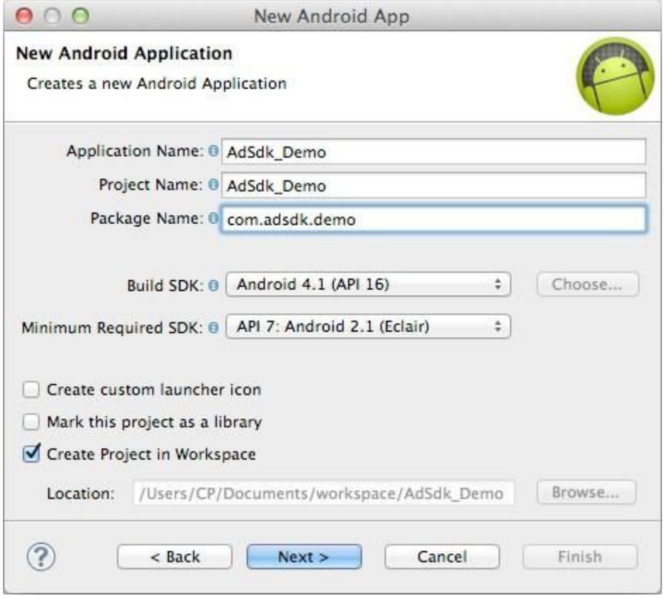
Step 2: Download the SDK

The downloaded ZIP contains the following files:

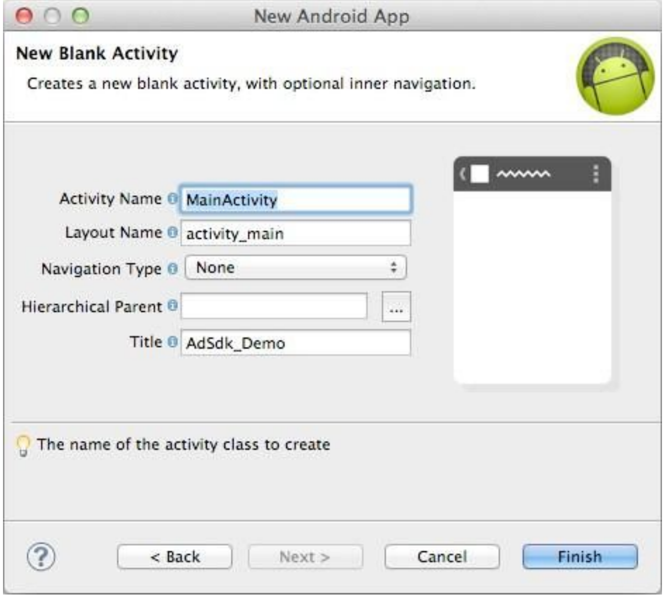
- PDF Documentation "AdsNaTa Android SDK Setup Guide 4.1.6.pdf" (this document)
- The SDK Framework itself: AdSdk_4.1.6.jar
- Demo Application "AdSdk_Demo" folder showcasing:
  - Request of Banner Ads and Interstitial Ads with Buttons or using code
  - Based on Eclipse 4.2.2 with Android SDK Rev. 21.1, SDK Platform Android 4.2.2 Rev. 2 (API 17)
  - Some Coding and Customization options

Step 3: Create and define a new Android Project in Eclipse

Download and open the latest version of Eclipse. Update the Android SDK Tools and Revisions listed at the Android SDK Manager. Now open the "Android Application Project" wizard and enter all needed fields as shown below. Set the minimum SDK to API 8 (Androids v2.2).



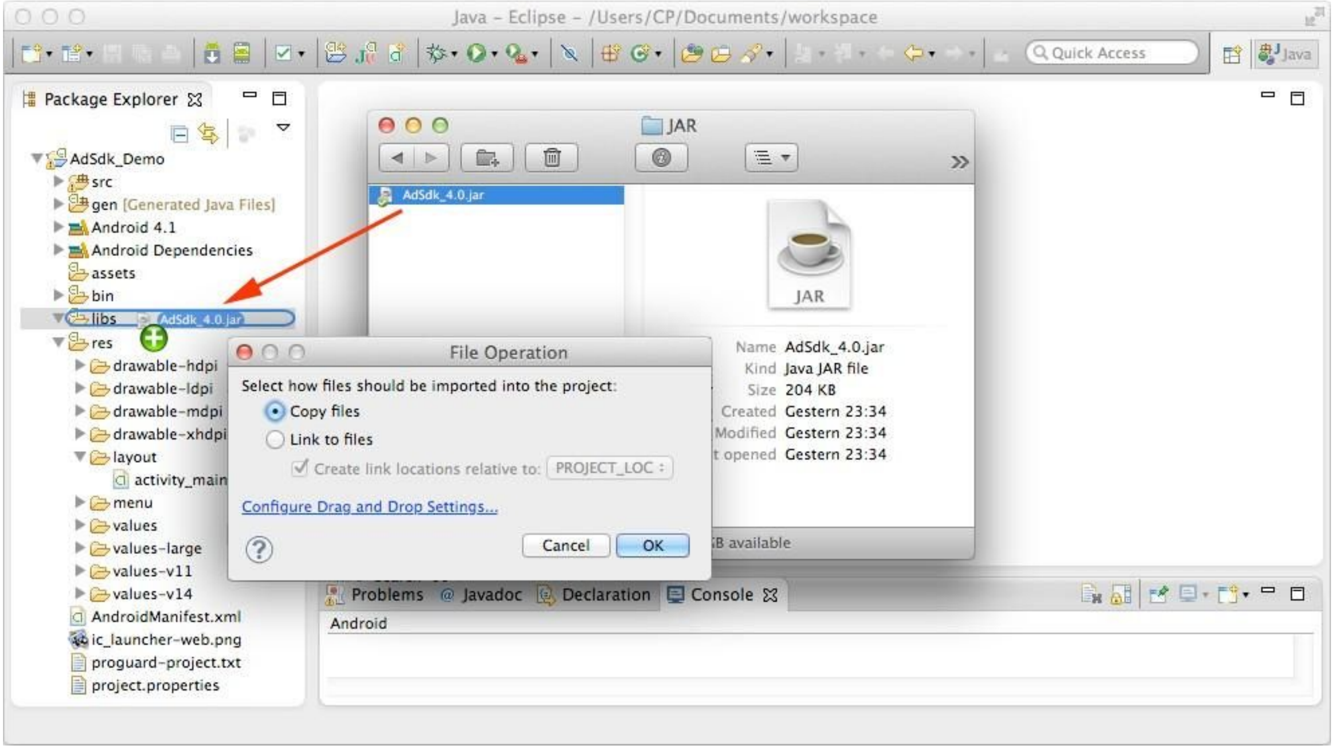
Application Name: AdSdk_Demo
Project Name: AdSdk_Demo
Package Name: com.adsdk.demo
Build SDK: Android 4.1 (API 16)
Minimum Required SDK: API 7: Android 2.1 (Eclair)
□ Create custom launcher icon
□ Mark this project as a library
☑ Create Project in Workspace
Location: /Users/CP/Documents/workspace/AdSdk_Demo


New Android App
New Blank Activity
Creates a new blank activity, with optional inner navigation.
< Back
Next >
Cancel
Finish

Activity Name: MainActivity
Layout Name: activity_main
Navigation Type: None
Hierarchical Parent:
Title: AdSdk_Demo
The name of the activity class to create

If you don't have a "libs" folder, create one by right clicking on the Project Root and select "New>Folder".

Copy the AdSdk_4.1.6.jar into your "libs" folder and select "Copy files" as shown below.



File Operation

Select how files should be imported into the project:

● Copy files

○ Link to files

☑ Create link locations relative to: PROJECT_LOC

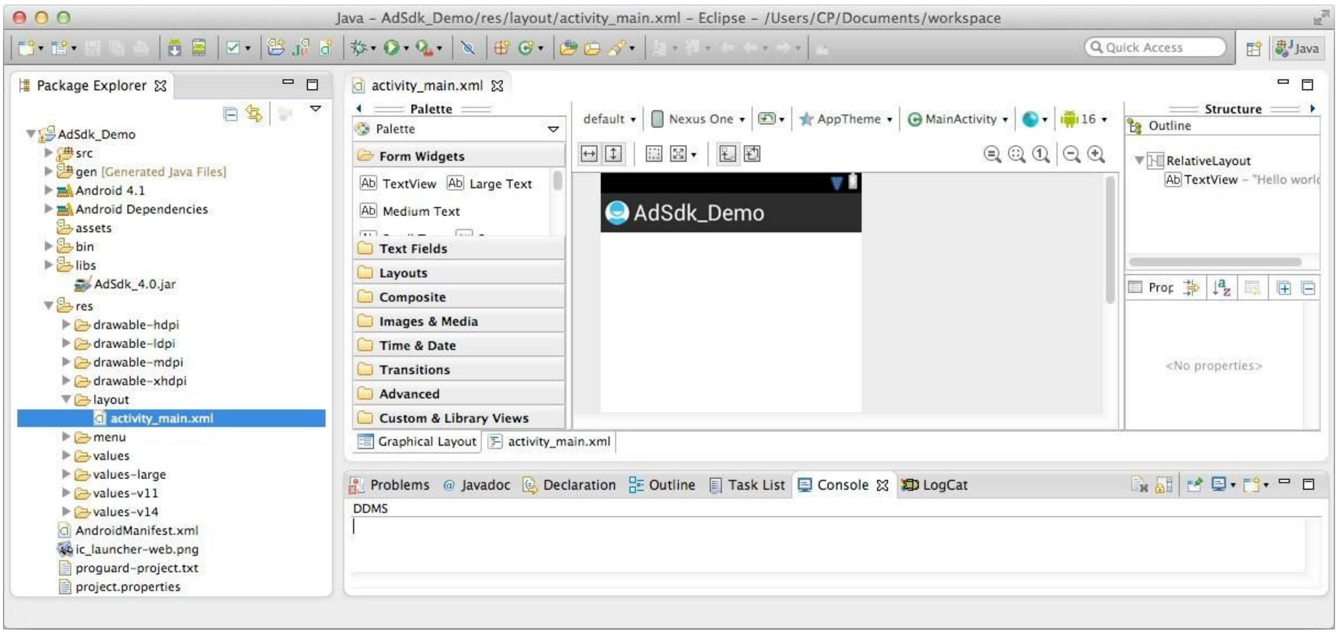
Configure Drag and Drop Settings...

Cancel OK

Name AdSdk_4.0.jar
Kind Java JAR file
Size 204 KB
Created Gestern 23:34
Modified Gestern 23:34
t opened Gestern 23:34

Note: If you are using SDK Rev 16 or earlier, you must additionally right click on Project, then Properties, go to "Java Build Path", select "Add JARs...", browse to your libs folder and select the AdSdk_4.1.6.jar. But we highly recommend to update your development tools to the latest version!

Please ensure the folder name is "libs". Your eclipse project should look as follows on SDK Rev 17 or later.



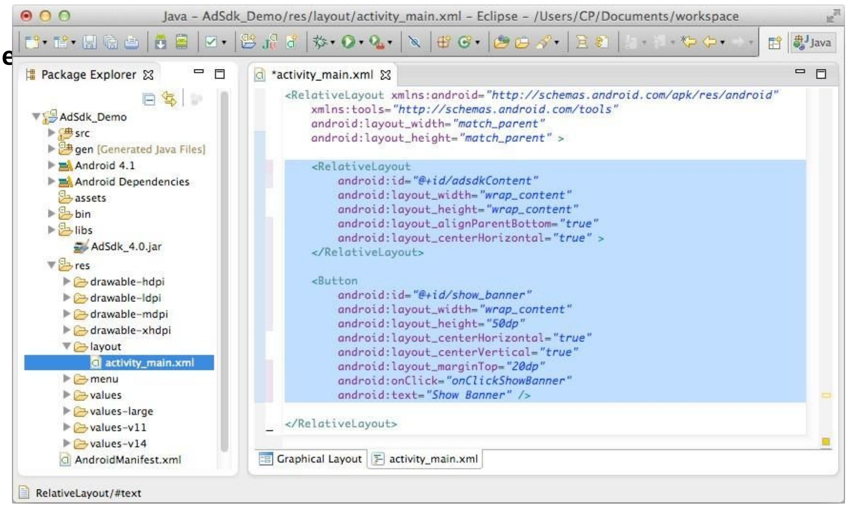
To display Banner Ads please continue with "Step 5".
To display Interstitial Ads please jump to "Step 6".

Open activity_main.xml and add the Banner without adding additional code as shown in AdSdk_XML_Demo.

Alternatively create a button to call "onClickShowBanner" (copy/paste activity_main.xml in AdSdk_Demo)

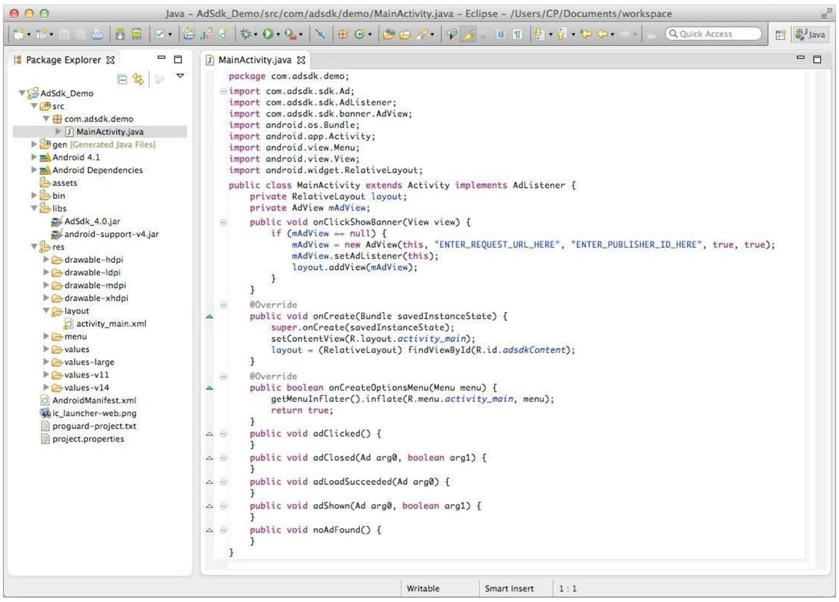
Instead of using a button display a Banner directly in MainActivity.java:

```
if (mAdView == null) {
   mAdView = new
   AdView(this,
      "ENTER_REQUEST_URL_HERE",
      "ENTER_PUBLISHER_ID_HERE"
      ,
      true, true);
   mAdView.setAdListener(this)
   ; layout.addView(mAdView);
   }
```



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <RelativeLayout
        android:id="@+id/adsdkContent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" >
    </RelativeLayout>

    <Button
        android:id="@+id/show_banner"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_marginTop="20dp"
        android:onClick="onClickShowBanner"
        android:text="Show Banner" />

</RelativeLayout>
```

Add some code now to your Activity (copy/paste from MainActivity.java provided in the AdSdk_Demo ZIP). This will initialize an instance of the Ad view with your Placement ID (called Publisher ID in code) and Request URL, init the button and set a listener on it to request an Ad. Listeners for the Ad view will handle events e.g. adLoadSucceeded,... Please note to edit both placeholders "ENTER_REQUEST_URL_HERE" and "ENTER_PUBLISHER_ID_HERE" with provided strings from your AdsNaTa control panel.



```
package com.adsdk.demo;
import com.adsdk.sdk.Ad;
import com.adsdk.sdk.AdListener;
import com.adsdk.sdk.banner.AdView;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.RelativeLayout;
public class MainActivity extends Activity implements AdListener {
    private RelativeLayout layout;
    private AdView mAdView;
    public void onClickShowBanner(View view) {
        if (mAdView == null) {
            mAdView = new AdView(this, "ENTER_REQUEST_URL_HERE", "ENTER_PUBLISHER_ID_HERE", true, true);
            mAdView.setAdListener(this);
            layout.addView(mAdView);
        }
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        layout = (RelativeLayout) findViewById(R.id.adsdkContent);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
    public void adClicked() {
    }
    public void adClosed(Ad arg0, boolean arg1) {
    }
    public void adLoadSucceeded(Ad arg0) {
    }
    public void adShown(Ad arg0, boolean arg1) {
    }
    public void noAdFound() {
    }
}
```

Add these Banner Ad specific lines to the Manifest (copy/paste from AndroidManifest.xml): (READ_PHONE_STATE and ACCESS_FINE_LOCATION are optional, but recommended)

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

<activity
       android:name="com.adsdk.sdk.banner.InAppWebView"
       android:configChanges="keyboard|keyboardHidden|orientation|screenLayou
       t|
                                uiMode|screenSize|smallestScreenSize" />
 <activity
       android:name="com.adsdk.sdk.mraid.MraidActivity"
       android:configChanges="keyboard|keyboardHidden|orientation|screenLayou
       t|
                                uiMode|screenSize|smallestScreenSize" />
```

Add the following to your activity with the banner (see .MainActivity at the AdSdk_Demo ZIP):

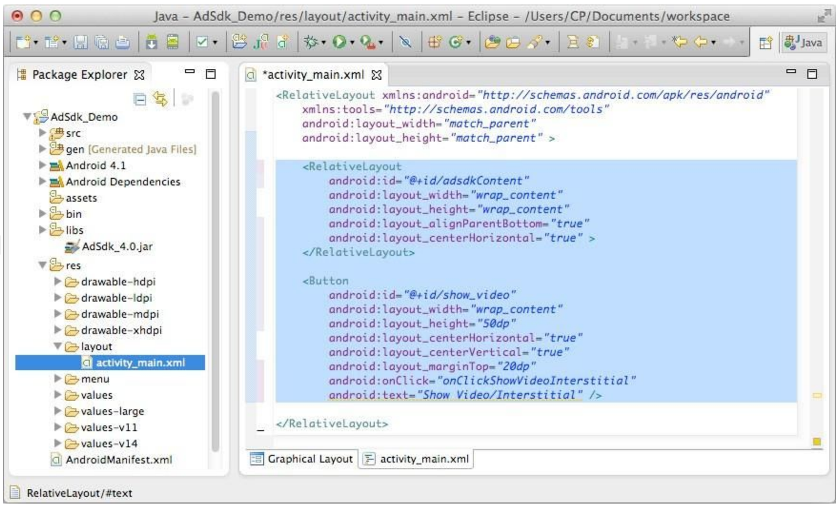
```
                android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|
                                 uiMode|screenSize|smallestScreenSize"
```

If you don't want to add Interstitial Ads please jump to "Step 7".

Open activity_main.xml and create a button which calls when pressed the method "onClickShowVideoInterstitial" (copy/paste from activity_main.xml)

Instead of using a button display an ad directly in MainActivity.java:

```
mManager = new AdManager(this,
      "ENTER_REQUEST_URL_HERE",
      "ENTER_PUBLISHER_ID_HERE"
      ,
       true);
mManager.setListener(this)
; mManager.requestAd();
```

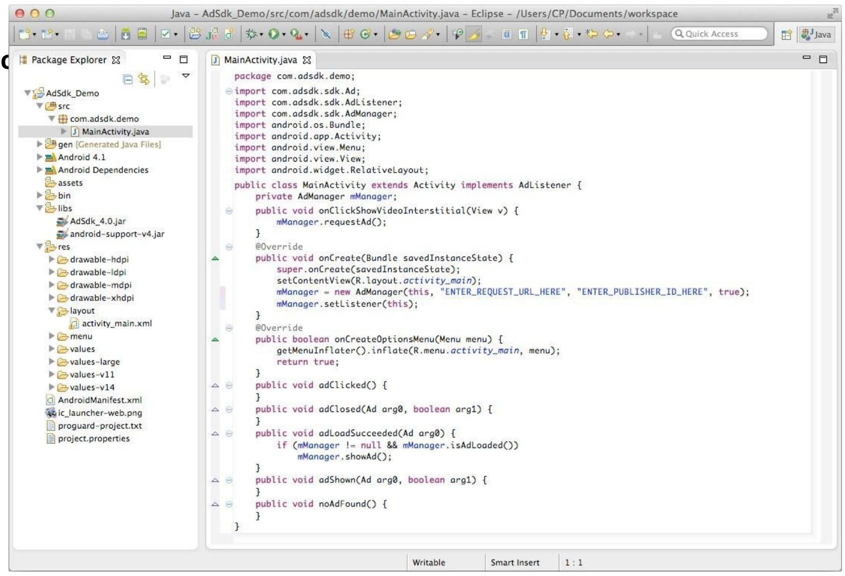


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <RelativeLayout
        android:id="@+id/adsdkContent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" >
    </RelativeLayout>

    <Button
        android:id="@+id/show_video"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_marginTop="20dp"
        android:onClick="onClickShowVideoInterstitial"
        android:text="Show Video/Interstitial" />

</RelativeLayout>
```

Add some code now to your Activity (copy/paste from MainActivity.java pro in the AdSdk_Demo ZIP).

The AdManager initialized with your Placement ID (Publisher ID in code) and Request URL and a listener will request an Ad. Listeners for the Ad view will handle events e.g. adLoadSucceeded,...

Make sure to edit both placeholders "ENTER_REQUEST_URL_HERE" and "ENTER_PUBLISHER_ID_HERE" with provided strings from your AdsNaTa control panel.



```
package com.adsdk.demo;
import com.adsdk.sdk.Ad;
import com.adsdk.sdk.AdListener;
import com.adsdk.sdk.AdManager;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.RelativeLayout;
public class MainActivity extends Activity implements AdListener {
    private AdManager mManager;
    public void onClickShowVideoInterstitial(View v) {
        mManager.requestAd();
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mManager = new AdManager(this, "ENTER_REQUEST_URL_HERE", "ENTER_PUBLISHER_ID_HERE", true);
        mManager.setListener(this);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
    public void adClicked() {
    }
    public void adClosed(Ad arg0, boolean arg1) {
    }
    public void adLoadSucceeded(Ad arg0) {
        if (mManager != null && mManager.isAdLoaded())
            mManager.showAd();
    }
    public void adShown(Ad arg0, boolean arg1) {
    }
    public void noAdFound() {
    }
}
```

Add these Interstitial Ad specific lines to the Manifest (copy/paste from AndroidManifest.xml): (READ_PHONE_STATE and ACCESS_FINE_LOCATION are optional, but recommended)

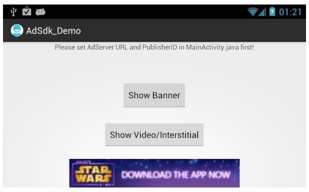
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<activity
        android:name="com.adsdk.sdk.video.RichMediaActivity"
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayou
        t|
                                uiMode|screenSize|smallestScreenSize
        " android:hardwareAccelerated="false" />
```

Optional and for Interstitial Ads only you can create an "anim" folder within your "res" folder. This will enable animations defined by the Ad Server. Uncompress animations.zip, drag and drop the

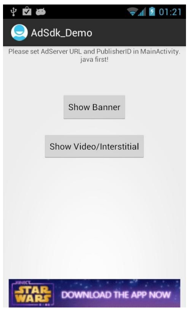
files to an "anim" folder and select "Copy files".

If you already created a campaign in your AdsNaTa Dashboard, you should now see the ad.

Note: Please always clear the heap before showing Ads and restore your apps orientation after an Ad is shown.



Please set AdServer URL and PublisherID in MainActivity.java first!



Please set AdServer URL and PublisherID in MainActivity.java first!

Explanation of Manifest

Internet Access Permission (mandatory and needed for the SDK to work)

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

Read Phone State Permission (optional but highly recommended)

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

Network State Permission (to identify the connection type and improve your app's user experience)

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

Location Permissions (to target ads for better customer experience)

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COURSE_LOCATION" />
```

Explanation of Methods

The Banner Ad constructor argument "Location" let's the SDK use the current location for better targeting Ads and "animation" enables transition animations between Banner Ads:

```
AdView(Context,"REQUEST_URL","PUBLISHER_ID", Location, animation);
```

The Interstitial Ad constructor argument "Location" let's the SDK use the current location for better targeting Ads:

```
AdManager(Context,"REQUEST_URL","PUBLISHER_ID", Location);
```

Once a new Ad has loaded, this AdListener method will be called on the main User Interface thread.

```
public void adLoadSucceeded(Ad advertisement) {}
```

If the Ad couldn't load, this AdListener method will be called and your can proceed to the next screen.

```
public void noAdFound() {}
```

For Interstitials you will be notified when the RichMediaActivity has finished to resume your activity.

```
public void adClosed(Ad advertisement, boolean completed) {}
```

After the Interstitial has shown and your activity has been paused, you will also be notified.

```
public void adShown(Ad advertisement, boolean succeeded) {}
```