Android Alert Dialog Example

Posted by: [Katerina Zamani](https://examples.javacodegeeks.com/author/katerina-zamani/" \o ") in [AlertDialog](https://examples.javacodegeeks.com/category/android/core/ui/alertdialog/) December 26th, 2013

Sometimes in our applications we want to alert the user for an event and/or ask him/her about taking a decision. For this purpose [AlertDialog](http://developer.android.com/reference/android/app/AlertDialog.html" \t "_blank)class can be used, where a message and one, two or three buttons are displayed in a popup window.

In this example, we are going to show an AlertDialog with three different buttons, where each one leads to a different action.

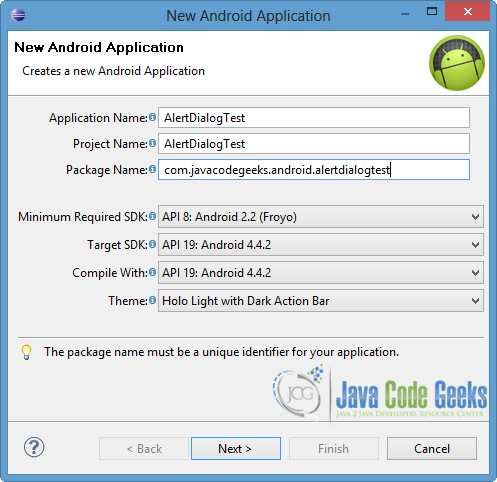
For this tutorial, we will use the following tools in a Windows 64-bit platform:

1. JDK 1.7
2. Eclipse 4.2 Juno
3. Android SDK 4.4

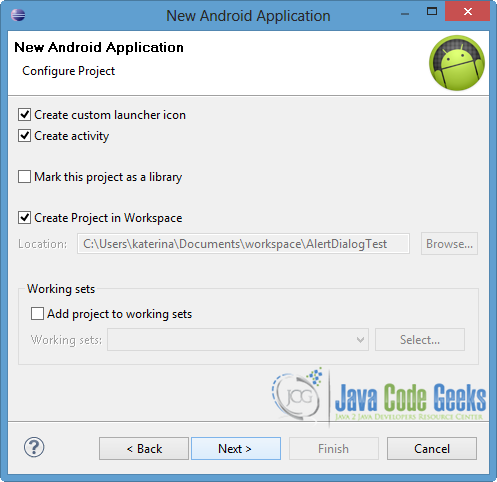
1. Create a New Android Application Project

Open Eclipse IDE and go to File → New → Project → Android Application Project.

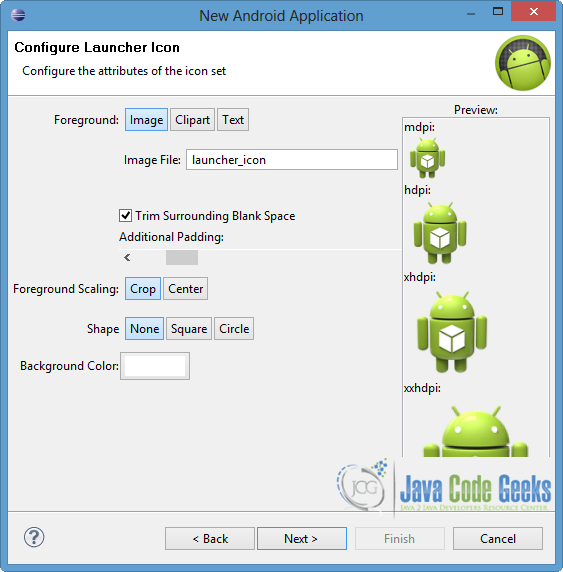
Fill in the name of the application, the project and the package in the appropriate fields and then click Next.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialProj1.png)

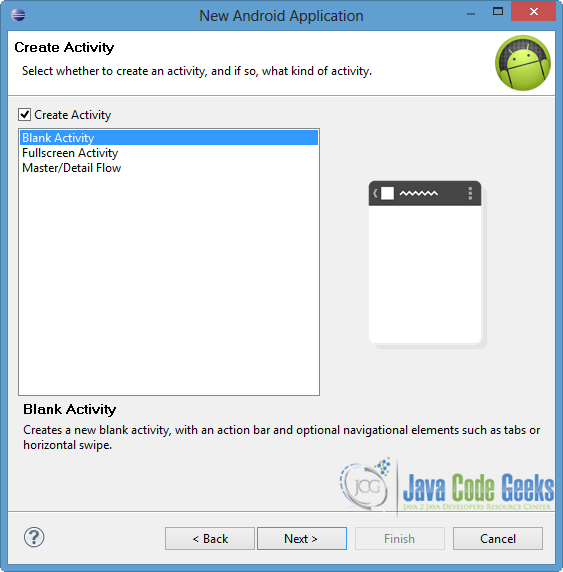
In the next window, check the “Create Activity” option. The new created activity will be the main activity of your project. Then press Next button.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialProj2.png)

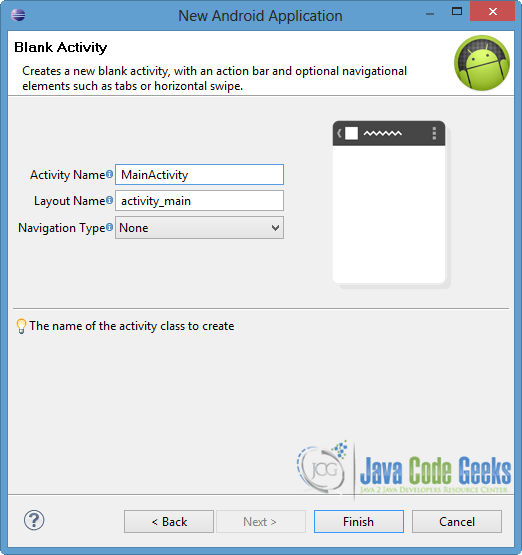
In “Configure Launcher Icon” window you should choose the icon you want to have in your app. We will use the default icon of android, so press Next button.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/11/createProject3.png)

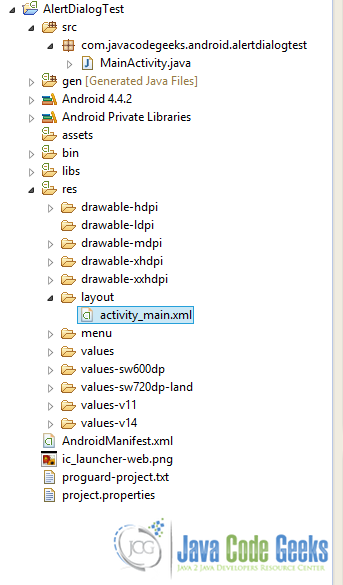
Select the “Blank Activity” option and click Next.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/11/createProject4.png)

Specify a name for the new Activity and a name for the layout description of your app. The .xml file for the layout will automatically be created in the res/layout folder. Finally, click Finish.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/11/createProject5.png)

You can see the structre of the AlertDialogTest project in the picture below.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialStructure.png)

2. Create the layout of the Main Activity

We will create a simple interface, so we will add a TextView and a Button, which enables the AlertDialog box.

Open res/layout/activity\_main.xml file and go to the xml tab. Then paste the following.

*activity\_main.xml:*

|  |  |
| --- | --- |
| 01 | <RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>" |
| 02 | xmlns:tools="<http://schemas.android.com/tools>" |

|  |  |
| --- | --- |
| 03 | android:layout\_width="match\_parent" |
| 04 | android:layout\_height="match\_parent" |

|  |  |
| --- | --- |
| 05 | tools:context=".MainActivity" > |
| 06 |  |

|  |  |
| --- | --- |
| 07 | <TextView |
| 08 | android:id="@+id/text" |

|  |  |
| --- | --- |
| 09 | android:layout\_width="wrap\_content" |
| 10 | android:layout\_height="wrap\_content" |

|  |  |
| --- | --- |
| 11 | android:text="Main activity of the app..." /> |
| 12 |  |

|  |  |
| --- | --- |
| 13 | <Button |
| 14 | android:id="@+id/button" |

|  |  |
| --- | --- |
| 15 | android:layout\_width="fill\_parent" |
| 16 | android:layout\_height="wrap\_content" |

|  |  |
| --- | --- |
| 17 | android:layout\_below="@+id/text" |
| 18 | android:layout\_marginTop="20dp" |

|  |  |
| --- | --- |
| 19 | android:text="Show Alert Box" /> |
| 20 |  |

|  |  |
| --- | --- |
| 21 | </RelativeLayout> |

3. Code the Main Activity

AlertDialog is a subclass of [Dialog](http://developer.android.com/reference/android/app/Dialog.html) that can display up to three Buttons. In order to create the interface of the alert box,[AlertDialog.Builder](http://developer.android.com/reference/android/app/AlertDialog.Builder.html" \t "_blank) should be called. AlertDialogBuilder class provides us positive, negative and neutral buttons by setting the setPositiveButton, setNegativeButton and setNeutralButton listeners respectively. At this point it is good to mention that the neutral button appears between the negative and the positive buttons.

In our example, we will use the three listeners. If the user chooses the positive button, a new Activity will launch by enabling an explicit Intent. In contrast, if the user presses the negative button, the alert box will close and we will go back to the main activity of our app. Finally, when the user clicks the neutral button, our application will close and it will lead the user to HOME.

Open src/com.javacodegeeks.android.alertdialogtest/MainActivity.java file and paste the following code.

*MainActivity.java:*

|  |  |
| --- | --- |
| 01 | package com.javacodegeeks.android.alertdialogtest; |
| 02 |  |

|  |  |
| --- | --- |
| 03 | import android.os.Bundle; |
| 04 | import android.app.Activity; |

|  |  |
| --- | --- |
| 05 | import android.app.AlertDialog; |
| 06 | import android.content.DialogInterface; |

|  |  |
| --- | --- |
| 07 | import android.content.Intent; |
| 08 | import android.view.View; |

|  |  |
| --- | --- |
| 09 | import android.view.View.OnClickListener; |
| 10 | import android.widget.Button; |

|  |  |
| --- | --- |
| 11 | import android.widget.Toast; |
| 12 |  |

|  |  |
| --- | --- |
| 13 | public class MainActivity extends Activity { |
| 14 |  |

|  |  |
| --- | --- |
| 15 | private Button mainBtn; |
| 16 |  |

|  |  |
| --- | --- |
| 17 | @Override |
| 18 | protected void onCreate(Bundle savedInstanceState) { |

|  |  |
| --- | --- |
| 19 | super.onCreate(savedInstanceState); |
| 20 | setContentView(R.layout.activity\_main); |

|  |  |
| --- | --- |
| 21 |  |
| 22 | mainBtn = (Button) findViewById(R.id.button); |

|  |  |
| --- | --- |
| 23 | mainBtn.setOnClickListener(new OnClickListener() { |
| 24 |  |

|  |  |
| --- | --- |
| 25 | @Override |
| 26 | public void onClick(View v) { |

|  |  |
| --- | --- |
| 27 | // TODO Auto-generated method stub |
| 28 | openAlert(v); |

|  |  |
| --- | --- |
| 29 | } |
| 30 | }); |

|  |  |
| --- | --- |
| 31 | } |
| 32 |  |

|  |  |
| --- | --- |
| 33 | private void openAlert(View view) { |
| 34 | AlertDialog.Builder alertDialogBuilder = newAlertDialog.Builder(MainActivity.this); |

|  |  |
| --- | --- |
| 35 |  |
| 36 | alertDialogBuilder.setTitle(this.getTitle()+ " decision"); |

|  |  |
| --- | --- |
| 37 | alertDialogBuilder.setMessage("Are you sure?"); |
| 38 | // set positive button: Yes message |

|  |  |  |
| --- | --- | --- |
| 39 | alertDialogBuilder.setPositiveButton("Yes",new DialogInterface.OnClickListener() { | |
| 40 | | public void onClick(DialogInterface dialog,int id) { |

|  |  |
| --- | --- |
| 41 | // go to a new activity of the app |
| 42 | Intent positveActivity = new Intent(getApplicationContext(), |

|  |  |
| --- | --- |
| 43 | PositiveActivity.class); |
| 44 | startActivity(positveActivity); |

|  |  |
| --- | --- |
| 45 | } |
| 46 | }); |

|  |  |  |
| --- | --- | --- |
| 47 | | // set negative button: No message |
| 48 | alertDialogBuilder.setNegativeButton("No",new DialogInterface.OnClickListener() { | |

|  |  |
| --- | --- |
| 49 | public void onClick(DialogInterface dialog,int id) { |
| 50 | // cancel the alert box and put a Toast to the user |

|  |  |
| --- | --- |
| 51 | dialog.cancel(); |
| 52 | Toast.makeText(getApplicationContext(), "You chose a negative answer", |

|  |  |
| --- | --- |
| 53 | Toast.LENGTH\_LONG).show(); |
| 54 | } |

|  |  |
| --- | --- |
| 55 | }); |
| 56 | // set neutral button: Exit the app message |

|  |  |
| --- | --- |
| 57 | alertDialogBuilder.setNeutralButton("Exit the app",newDialogInterface.OnClickListener() { |
| 58 | public void onClick(DialogInterface dialog,int id) { |

|  |  |
| --- | --- |
| 59 | // exit the app and go to the HOME |
| 60 | MainActivity.this.finish(); |

|  |  |
| --- | --- |
| 61 | } |
| 62 | }); |

|  |  |
| --- | --- |
| 63 |  |
| 64 | AlertDialog alertDialog = alertDialogBuilder.create(); |

|  |  |
| --- | --- |
| 65 | // show alert |
| 66 | alertDialog.show(); |

|  |  |
| --- | --- |
| 67 | } |
| 68 |  |

|  |  |
| --- | --- |
| 69 | } |

4. Crete a Custom Activity

We will create a new Activity for the explicit Intent of the positive answer.

To do so right click on com.javacodegeeks.android.alertdialogtest package → New → Class. Specify the name (PositiveActivity as we defined it in the MainActivity file) and the package that will belong to and finally click Finish.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialPositiveActivity.png)

Open src/com.javacodegeeks.android.alertdialogtest/PositiveActivity.java file and paste the code below.

*PositiveActivity.java:*

|  |  |
| --- | --- |
| 01 | package com.javacodegeeks.android.alertdialogtest; |
| 02 |  |

|  |  |
| --- | --- |
| 03 | import android.app.Activity; |
| 04 | import android.os.Bundle; |

|  |  |
| --- | --- |
| 05 |  |
| 06 | public class PositiveActivity extends Activity { |

|  |  |
| --- | --- |
| 07 | @Override |
| 08 | protected void onCreate(Bundle savedInstanceState) { |

|  |  |
| --- | --- |
| 09 | super.onCreate(savedInstanceState); |
| 10 | setContentView(R.layout.positive\_view); |

|  |  |
| --- | --- |
| 11 | } |
| 12 |  |

|  |  |
| --- | --- |
| 13 | } |

5. Create the layout of the Custom Activity

We are going to create a simple UI for the layout of the PositiveActivity class.

Right click on res/layout → New → Android XML File. Then, specify the name of the file and choose LinearLayout as root element, like in the image below. Then click Finish.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialpositive_view.png)

Open res/layout/positive\_view.xml and paste the following.

*positive\_view.xml:*

|  |  |
| --- | --- |
| 01 | <?xml version="1.0" encoding="utf-8"?> |
| 02 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>" |

|  |  |
| --- | --- |
| 03 | android:layout\_width="match\_parent" |
| 04 | android:layout\_height="match\_parent" |

|  |  |
| --- | --- |
| 05 | android:orientation="vertical" > |
| 06 |  |

|  |  |
| --- | --- |
| 07 | <TextView |
| 08 | android:id="@+id/textPositive" |

|  |  |
| --- | --- |
| 09 | android:layout\_width="fill\_parent" |
| 10 | android:layout\_height="wrap\_content" |

|  |  |
| --- | --- |
| 11 | android:text="You chose a positive answer..." /> |
| 12 |  |

|  |  |
| --- | --- |
| 13 | </LinearLayout> |

6. Define the rules for the Custom Activity

We have to define the PositiveActivity activity in our AndroidManifest.xml file. In our occasion, we want PositiveActivityto leads us back to the MainActivity, so we will define the MainActivity as it’s parent Activity. For this reasonparentActivityName android element is used.

Open AndroidManifest.xml file and choose the xml tab. Then paste the following code.

*AndroidManifest.xml:*

|  |  |
| --- | --- |
| 01 | <?xml version="1.0" encoding="utf-8"?> |
| 02 | <manifest xmlns:android="<http://schemas.android.com/apk/res/android>" |

|  |  |
| --- | --- |
| 03 | package="com.javacodegeeks.android.alertdialogtest" |
| 04 | android:versionCode="1" |

|  |  |
| --- | --- |
| 05 | android:versionName="1.0" > |
| 06 |  |

|  |  |
| --- | --- |
| 07 | <uses-sdk |
| 08 | android:minSdkVersion="8" |

|  |  |
| --- | --- |
| 09 | android:targetSdkVersion="19" /> |
| 10 |  |

|  |  |
| --- | --- |
| 11 | <application |
| 12 | android:allowBackup="true" |

|  |  |
| --- | --- |
| 13 | android:icon="@drawable/ic\_launcher" |
| 14 | android:label="@string/app\_name" |

|  |  |
| --- | --- |
| 15 | android:theme="@style/AppTheme" > |
| 16 | <activity |

|  |  |  |
| --- | --- | --- |
| 17 | android:name="com.javacodegeeks.android.alertdialogtest.MainActivity" | |
| 18 | | android:label="@string/app\_name" > |

|  |  |
| --- | --- |
| 19 | <intent-filter> |
| 20 | <action android:name="android.intent.action.MAIN" /> |

|  |  |
| --- | --- |
| 21 |  |
| 22 | <category android:name="android.intent.category.LAUNCHER" /> |

|  |  |
| --- | --- |
| 23 | </intent-filter> |
| 24 | </activity> |

|  |  |
| --- | --- |
| 25 |  |
| 26 | <activity |

|  |  |  |
| --- | --- | --- |
| 27 | android:name="com.javacodegeeks.android.alertdialogtest.PositiveActivity" | |
| 28 | | android:label="Positive Answer" |

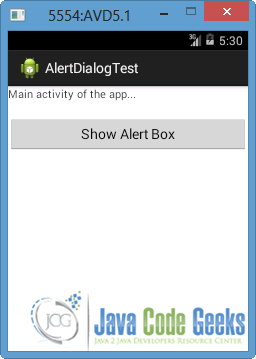
|  |  |  |
| --- | --- | --- |
| 29 | android:parentActivityName="com.javacodegeeks.android.alertdialogtest.MainActivity"> | |
| 30 | | </activity> |

|  |  |
| --- | --- |
| 31 | </application> |
| 32 |  |

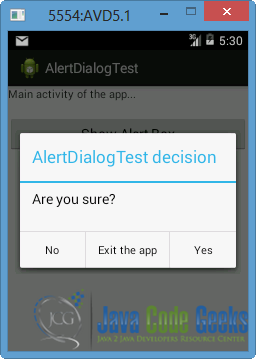
|  |  |
| --- | --- |
| 33 | </manifest> |

7. Run the application

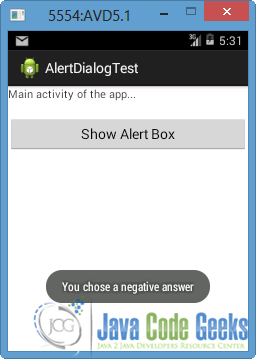
To run our application, right click on our project → Run as → Android Application. The AVD will appear with the app loaded, as you can see below.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AVDAlertDialog1.png)

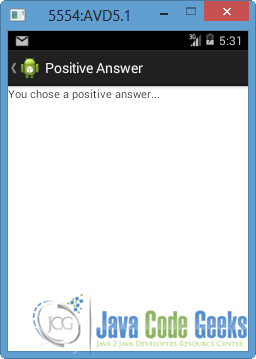
Now press the “Show Alert Box” button. An alert box will popup and the user should take a decision, in order to proceed. Notice that the buttons don’t appear in the order we set them at the MainActivity.java, because Android system defines the button order as NEGATIVE - NEUTRAL - POSITIVE.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AVDAlertDialog2.png)

If we click the “No” button, the alert box will be canceled and the user will go back to the MainActivity with the appearance of a Toast, as you can see in the next picture.

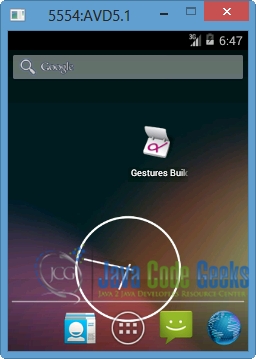
[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AVDAlertDialog4.png)

When the user presses the “Yes” button, the setPositiveButton listener will be enabled, so the PositiveActivity will launch.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AVDAlertDialog3.png)

Now we can go back at the parent Activity, by pressing the button at the top left of the screen.

Finally, if we press “Exit the app” button, our app will close down and we will get lead to the main screen of the emulator.

[](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AVDAlertDialog5.png)

Of course, the user can choose and launch this application again.

Download Eclipse Project

This was an example of Alert Dialog in Android. Download the Eclipse Project of this example: [AlertDialogTest.zip](http://examples.javacodegeeks.com/wp-content/uploads/2013/12/AlertDialogTest.zip)