Simple Button for Android – XML onClick + public method

Probably the simplest way to create Android Button. Create a new public method (e.g. goButtonClicked) and add it to your layout XML.

Prequisites: Android Hello World Explained

In Java, define a new public method that takes view as parameter.


```
public void goButtonClicked(View v) {
        // do stuff
}
```

In your XML file, create a new button. Name your method as onClick property for the button.

```
<Button android:text="Go!" android:onClick="goButtonClicked"
        android:id="@+id/goButton"></Button>
```

So, the first step is to add a resource for this button:

```
<LinearLayout 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" android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

  <!--<TextView android:text="@string/hello_world" android:layout_width="wrap_content"
   android:layout_height="wrap_content" />-->

  <Button android:layout_height="wrap_content"
    android:layout_width="wrap_content" android:text="Go!"
    android:onClick="goButtonClicked" android:id="@+id/goButton"></Button>
</LinearLayout>
```

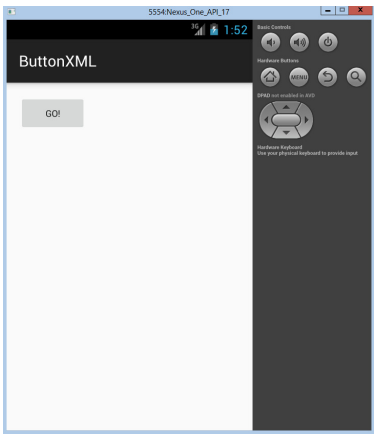
Secondly at Java side is to add the method for the button:

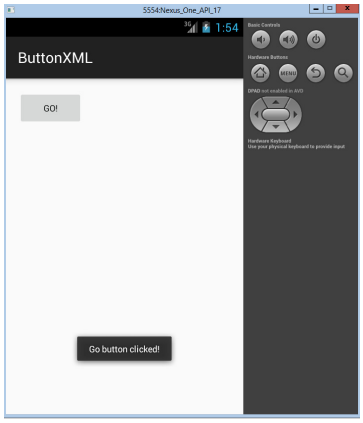
```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

  public void goButtonClicked(View v) {
    tToast("Go button clicked!");
  }
```

And finally add the code for the toast:

```
private void tToast(String s) {
    Context context = getApplicationContext();
    int duration = Toast.LENGTH_LONG;
    Toast toast = Toast.makeText(context, s, duration);
    toast.show();
  }
```



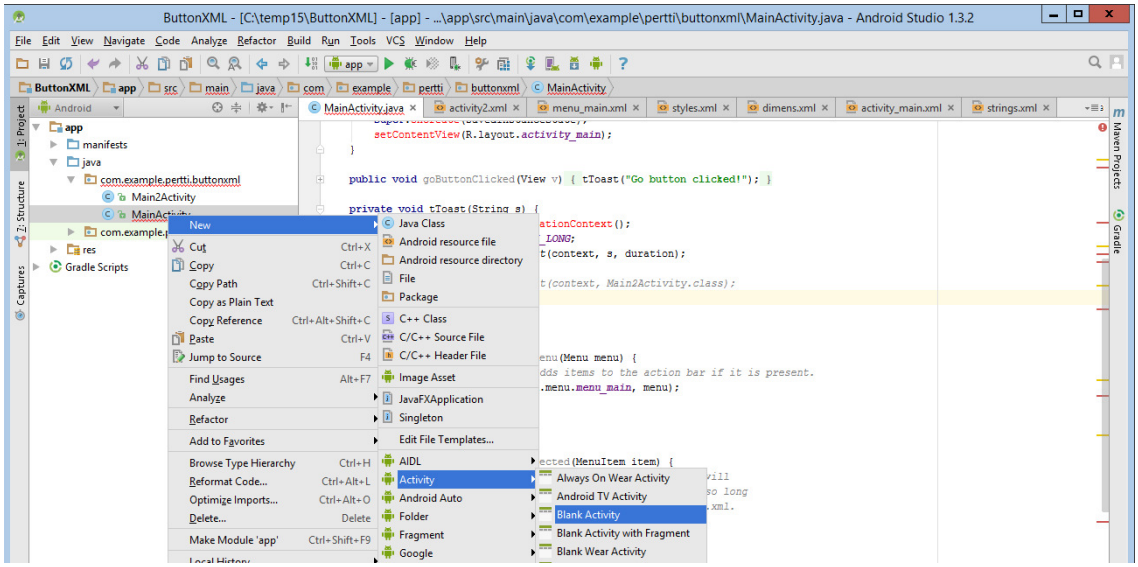


Android Activity – From One Screen To Another Screen

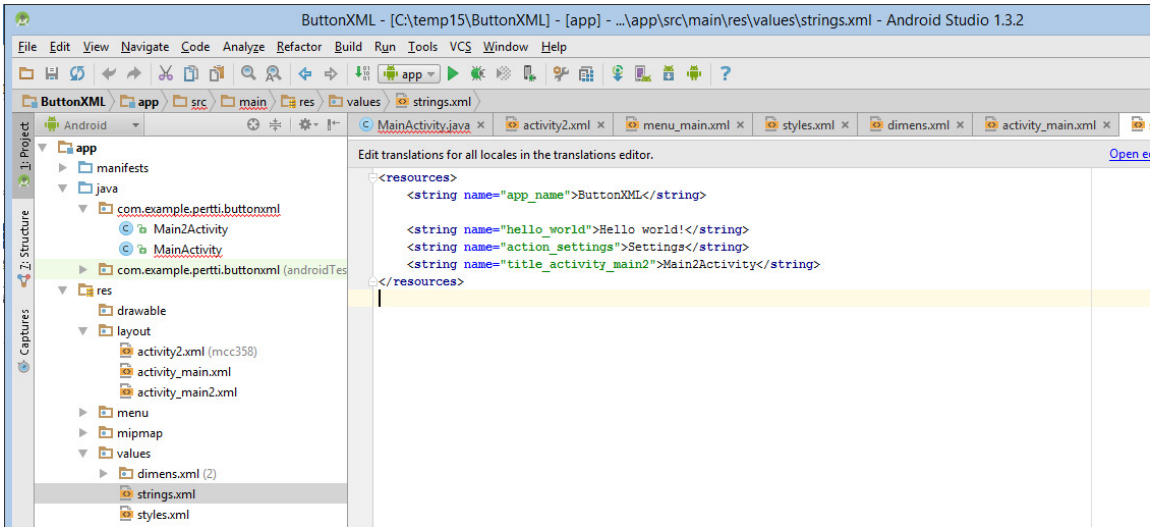
In Android, an activity is represent a single screen. Most applications have multiple activities to represent different screens, for example, one activity to display a list of the application settings, another activity to display the application status.

Now we show you how to interact with activity, when a button is clicked, navigate from current screen (current activity) to another screen (another activity).

In current activity (ButtonXML/MainActivity) add new activity (Main2Activity) by mouse right:



See how this builds new resources, Java code, and strings needed for the new activity:



As well also Manifest includes the new activity:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.pertti.buttonxml" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Main2Activity"
            android:label="@string/title_activity_main2" >
        </activity>
    </application>

</manifest>
```

Test the application after inserting the new lines (in previous application ButtonXML) for the 2nd activity:

```
    private void tToast(String s) {
            Context context = getApplicationContext();
            int duration = Toast.LENGTH_LONG;
            Toast toast = Toast.makeText(context, s, duration);
            toast.show();
            Intent intent = new Intent(context, Main2Activity.class);
            startActivity(intent);
      }
```