**AIM: CREATE A NEW ANDROID APPLICATION USING ALERT BOX**

open an **activity\_main.xml** file from **\res\layout** path and write the code like as shown below

## activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent" android:layout\_height="match\_parent">  
      <Button  
        android:id="@+id/getBtn"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="150dp"  
        android:layout\_marginTop="200dp"  
        android:text="Show Alert" />  
</RelativeLayout>

If you observe above code we defined a one [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) control in [RelativeLayout](https://www.tutlane.com/tutorial/android/android-relativelayout-with-examples) to show the alert dialog on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click in XML layout file.

Once we are done with the creation of layout with required controls, we need to load the XML layout resource from our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) **onCreate()** callback method, for that open main [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **MainActivity.java** from **\java\com.tutlane.alertdialogexample** path and write the code like as shown below.

## MainActivity.java

package com.tutlane.alertdialogexample;  
import android.content.DialogInterface;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        Button btn = (Button)findViewById(R.id.getBtn);  
        btn.setOnClickListener(new View.OnClickListener() {  
  @Override  
  public void onClick(View v) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
   builder.setTitle("Login Alert").setMessage("Are you sure, you want to continue ?")  
                        .setCancelable(false)  
                        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
                            @Override  
                            public void onClick(DialogInterface dialog, int which) {  
                                Toast.makeText(MainActivity.this,"Selected Option: YES",Toast.LENGTH\_SHORT).show();  
                            }  
                        })  
                        .setNegativeButton("No", new DialogInterface.OnClickListener() {  
                            @Override  
                            public void onClick(DialogInterface dialog, int which) {  
                                Toast.makeText(MainActivity.this,"Selected Option: No",Toast.LENGTH\_SHORT).show();  
                            }  
                        });  
                //Creating dialog box  
                AlertDialog dialog  = builder.create();  
                dialog.show();  
            }  
        });  
    }  
}

If you observe above code we are calling our layout using **setContentView** method in the form of **R.layout.layout\_file\_name** in our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file. Here our xml file name is **activity\_main.xml** so we used file name **activity\_main** and we are trying to show the AlertDialog on [Button](https://www.tutlane.com/tutorial/android/android-button-with-examples) click.

Generally, during the launch of our [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle), **onCreate()** callback method will be called by android framework to get the required layout for an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle).

## Output

