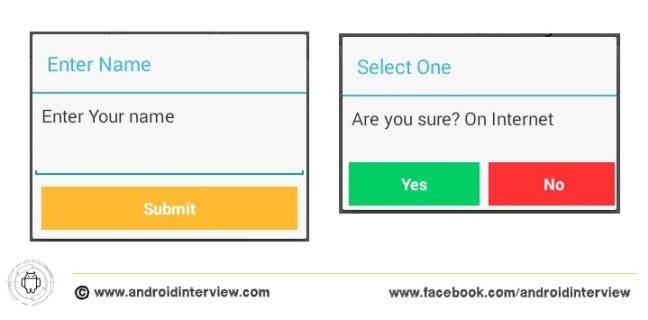
Android custom Dialog box Tutorial

Dialog box  in Android :

In Android, a dialog is usually a small floating window that appears in front of the present activity. once a dialog appears, the present activity loses focus and user interaction becomes focused on the dialog.

A dialog is usually wont to get a response from the user. In Android, a dialog is implemented as a dialog fragment, that floats on top of an activity and is displayed Model. Dialog fragments are useful once you need to acquire the user’s response before continuing with the execution or any type of permission etc .

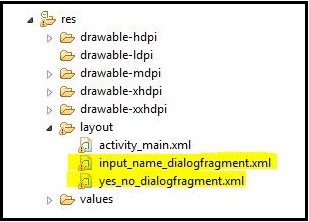
[](http://i1.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-custom-dialog-5.jpg)

To make a dialog fragment, you will need to extend the [DialogFragment](http://developer.android.com/guide/topics/ui/dialogs.html" \l "DialogFragment) base class. This example shows you ways to make 2 styles of dialogs: AN input dialog and a Status dialog.

Note : This example (Project) is developed in Eclipse Version Kepler,tested on Android 4.4.2 (KitKat Version) and SDK build-tools 22.6.3

Solution Stuff :

For Display dialog box in this example you need  add two Android XML layout files to the **res/layou**t folder and name them as shown in Figure :

[](http://i1.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-custom-dialog-6.jpg)

1. Open “**res/layout/input\_name\_dialogfragment.xml**” file and put this xml code it into file:

*File : res/layout/input\_name\_dialogfragment.xml*

This Userinterface basically contains a page with a label, a text box, and a button which display in dialog.

XHTML

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical" >        <TextView  android:text="Enter Your name"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:padding="10dp"  android:textSize="20dp"/>  <EditText  android:id="@+id/txtName"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:inputType="text"  android:imeOptions="actionDone"  android:layout\_marginTop="5dp"/>  <Button  android:id="@+id/btnDone"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Submit"  android:layout\_margin="10dp"  android:background="#FFFFBB33"  android:textColor="#fff"  android:textStyle="bold"/>    </LinearLayout> |

2. Open “**res/layout/yes\_no\_dialogfragment.xml**” file and put this xml code it into file:

*File : res/layout/yes\_no\_dialogfragment.xml*

This Userinterface contains a label and two buttons of android dialog box.

XHTML

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical" >        <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:padding="10dp"  android:text="Are you sure? On Internet"  android:textSize="20dp"/>  <LinearLayout  android:layout\_width="300dp"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:orientation="horizontal"  android:layout\_marginTop="20dp">    <Button      android:id="@+id/btnYes"      android:layout\_width="145dp"      android:layout\_height="wrap\_content"      android:text="Yes"      android:background="#00CC66"      android:textColor="#fff"      android:textStyle="bold"      android:layout\_margin="5dp"/>    <Button      android:id="@+id/btnNo"      android:layout\_width="145dp"      android:layout\_height="wrap\_content"      android:text="No"      android:background="#FF3333"      android:textColor="#fff"      android:textStyle="bold"      android:layout\_margin="5dp" />  </LinearLayout>  </LinearLayout> |

Add two new Java classes to the project for Custom Dialog as shown in Figure :

[](http://i1.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-custom-dialog-7.jpg)

3. Open “**InputNameDialogFragment.java**” file and add following JAVA code.

*File : src/package-name/InputNameDialogFragment.java*

Java

|  |
| --- |
| package androidinterview.com.customdialog;    import android.os.Bundle;  import android.support.v4.app.DialogFragment;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.view.WindowManager.LayoutParams;  import android.widget.Button;  import android.widget.EditText;    public class InputNameDialogFragment extends DialogFragment {  EditText txtname;  Button btnDone;  static String DialogboxTitle;    public interface InputNameDialogListener {  void onFinishInputDialog(String inputText);  }    //---empty constructor required  public InputNameDialogFragment() {    }  //---set the title of the dialog window  public void setDialogTitle(String title) {  DialogboxTitle = title;  }    public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle saveInstanceState){    View view = inflater.inflate(  R.layout.input\_name\_dialogfragment, container);    //---get the EditText and Button views  txtname = (EditText) view.findViewById(R.id.txtName);  btnDone = (Button) view.findViewById(R.id.btnDone);    //---event handler for the button  btnDone.setOnClickListener(new View.OnClickListener()  {  public void onClick(View view) {    //---gets the calling activity  InputNameDialogListener activity = (InputNameDialogListener) getActivity();  activity.onFinishInputDialog(txtname.getText().toString());    //---dismiss the alert  dismiss();  }  });    //---show the keyboard automatically  txtname.requestFocus();  getDialog().getWindow().setSoftInputMode(  LayoutParams.SOFT\_INPUT\_STATE\_VISIBLE);    //---set the title for the dialog  getDialog().setTitle(DialogboxTitle);    return view;  }  } |

This class extends the DialogFragment base class and loads the UI defined within the *input\_name\_dialogfragment.xml*file. in this class, you created an interface called InputNameDialogListener with an abstract method named onFinishInputDialog(). This method are implemented by the activity that calls this class. In essence, it’s dismissed once the user clicks the button to shut the dialog.The text entered into the text box is then sent back to the calling activity through this method.

4. Open “**YesNoDialogFragment.java**” file and add following JAVA code.

*File : src/package-name/YesNoDialogFragment.java*

Java

|  |
| --- |
| package androidinterview.com.customdialog;    import android.os.Bundle;  import android.support.v4.app.DialogFragment;  import android.view.LayoutInflater;  import android.view.View;  import android.view.View.OnClickListener;  import android.view.ViewGroup;  import android.widget.Button;    public class YesNoDialogFragment extends DialogFragment {  Button btnYes,btnNo;  static String DialogBoxTitle;    public interface YesNoDialogListener {  void onFinishYesNoDialog(boolean state);  }    //---empty constructor required  public YesNoDialogFragment(){    }  //---set the title of the dialog window---  public void setDialogTitle(String title) {  DialogBoxTitle= title;  }    public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState ) {    View view= inflater.inflate(R.layout.yes\_no\_dialogfragment, container);  //---get the Button views---  btnYes = (Button) view.findViewById(R.id.btnYes);  btnNo = (Button) view.findViewById(R.id.btnNo);    // Button listener  btnYes.setOnClickListener(btnListener);  btnNo.setOnClickListener(btnListener);    //---set the title for the dialog  getDialog().setTitle(DialogBoxTitle);    return view;  }    //---create an anonymous class to act as a button click listener  private OnClickListener btnListener = new OnClickListener()  {  public void onClick(View v)  {  //---gets the calling activity---  YesNoDialogListener activity = (YesNoDialogListener) getActivity();  boolean state =  ((Button) v).getText().toString().equals("Yes") ? true : false;  activity.onFinishYesNoDialog(state);  //---dismiss the alert---  dismiss();  }  };  } |

same as InputNameDialogFragment class, this class extends the DialogFragment class and implements an interface.

5. Open “**res/layout/activity\_main.xml**” file and put this xml code it into file:

*File : res/layout/activity\_main.xml*

XHTML

|  |
| --- |
| <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:orientation="vertical" >    <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Show Input Dialog"  android:onClick="btnShowDialog" />    <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Show Yes/No Dialog"  android:onClick="btnShowYesNoDialog" />  </LinearLayout> |

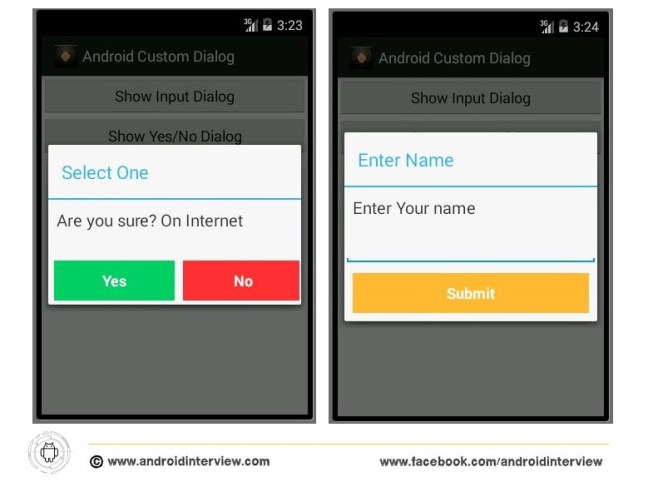
6. Open “**MainAcitvity.class**” file and add following JAVA code.

*File : src/package-name/MainActivity.java*

Java

|  |
| --- |
| package androidinterview.com.customdialog;    import android.os.Bundle;  import android.support.v4.app.FragmentActivity;  import android.support.v4.app.FragmentManager;  import android.view.View;  import android.widget.Toast;  import androidinterview.com.customdialog.InputNameDialogFragment.InputNameDialogListener;  import androidinterview.com.customdialog.YesNoDialogFragment.YesNoDialogListener;    public class MainActivity extends FragmentActivity implements YesNoDialogListener, InputNameDialogListener {    *@Override*  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  }    //-- InputName Dialog box  public void btnShowDialog(View view) {  showInputNameDialog();  }    private void showInputNameDialog() {  FragmentManager fragmentManager = getSupportFragmentManager();  InputNameDialogFragment inputNameDialog = new InputNameDialogFragment();  inputNameDialog.setCancelable(false);  inputNameDialog.setDialogTitle("Enter Name");  inputNameDialog.show(fragmentManager, "Input Dialog");  }    *@Override*  public void onFinishInputDialog(String inputText) {  // -- Finish dialog box show msg  Toast.makeText(this, "Input Name to dialog: " + inputText,  Toast.LENGTH\_SHORT).show();  }    //-- YES-No Dialog box  public void btnShowYesNoDialog(View view) {  showYesNoDialog();  }    private void showYesNoDialog() {  FragmentManager fragmentManager = getSupportFragmentManager();  YesNoDialogFragment yesnoDialog = new YesNoDialogFragment();  yesnoDialog.setCancelable(false);  yesnoDialog.setDialogTitle("Select One");  yesnoDialog.show(fragmentManager, "Yes/No Dialog");  }  *@Override*  public void onFinishYesNoDialog(boolean state) {  // -- Finish dialog box show msg  Toast.makeText(this, "Which Option Selected: " + state,  Toast.LENGTH\_SHORT).show();  }    } |

**Output of  Android Custom Dialog box in emulator :**

[](http://i0.wp.com/www.androidinterview.com/wp-content/uploads/2014/08/android-custom-dialog.jpg)