Mobile Application Development

Topic: ***Android ListView Tutorial***

Prepared by Mr. Muhammad Nabeel

Android ListView

Android ListView is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an Adapter that pulls content from a source such as an array or database.

An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter holds the data and send the data to adapter view, the view can take the data from adapter view and shows the data on different views like as spinner, list view, grid view etc.

The **ListView** and **GridView** are subclasses of **AdapterView** and they can be populated by binding them to an Adapter, which retrieves data from an external source and creates a View that represents each data entry.

Android provides several subclasses of Adapter that are useful for retrieving different kinds of data and building views for an AdapterView (i.e. ListView or GridView). The common adapters are ArrayAdapter,Base Adapter, CursorAdapter, SimpleCursorAdapter,SpinnerAdapter and WrapperListAdapter. We will see separate examples for both the adapters.

ListView Attributes

Following are the important attributes specific to GridView –

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:divider	This is drawable or color to draw between list items. .
android:dividerHeight	This specifies height of the divider. This could be in px, dp, sp, in, or mm.
android:entries	Specifies the reference to an array resource that will populate the ListView.
android:footerDividersEnabled	When set to false, the ListView will not draw the divider before each footer view. The default value is true.
android:headerDividersEnabled	When set to false, the ListView will not draw the divider after each header view. The default value is true.

Code to Implement for switching Activity (Android alert dialog with One button)

1) Step 1: Create a new Empty Project named it "SimpleListView "

2) Create a main file named it as Activity_Main.java (If not created already)

3) Create XML file and named it as activity_main.xml (if not created already)

4) Create list_data.xml in values folder and write following code in it

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="adobe_products">
        <item>Adobe After Effects</item>
        <item>Adobe Bridge</item>
        <item>Adobe Dreamweaver</item>
        <item>Adobe Edge</item>
        <item>Adobe Fireworks</item>
        <item>Adobe Flash</item>
        <item>Adobe Photoshop</item>
        <item>Adobe Premiere</item>
        <item>Adobe Reader</item>
        <item>Adobe Illustrator</item>
        <item>PHP</item>
        <item>.Net</item>
        <item>C++</item>
    </string-array>
</resources>
```

5) In layout folder add following line in acivity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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"
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">

    <ListView
        android:id="@+id/mobile_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>

</RelativeLayout>
```

6) In layout folder create activity_listview.xml and add following code into it.

```
<?xml version="1.0" encoding="utf-8"?>
<!--  Single List Item Design -->

<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dip"
    android:textSize="20dip"
    android:textStyle="bold" >
</TextView>
```

7) In layout folder create single_list_item_view.xml and add following code into it

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView android:id="@+id/product_label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="25dip"
        android:textStyle="bold"
        android:padding="10dip"
        android:textColor="#ffffff"/>
</LinearLayout>
```

8) Open MainActivity.java and add following code into it.

Important Note: Do not remove your package name in a file that is written on top of the files.

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);

        // storing string resources into Array
        String[] adobe_products =
getResources().getStringArray(R.array.adobe_products);

        ArrayAdapter adapter = new ArrayAdapter<String>(this,
R.layout.activity_listview, adobe_products);

        ListView listView = (ListView) findViewById(R.id.mobile_list);
        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View view,
                                    int position, long id) {

                String itemPosition     = Integer.toString(position);

                // When clicked, show a toast with the TextView text
                Toast.makeText(getApplicationContext(),
                        ((TextView) view).getText() + itemPosition,
Toast.LENGTH_SHORT).show();

                String product = ((TextView) view).getText().toString();

                // Launching new Activity on selecting single List Item
                Intent i = new Intent(getApplicationContext(),
SingleListItem.class);
                // sending data to new activity
                i.putExtra("product", product);
                startActivity(i);

            }
        });

    }


}
```

9) Create SingleListItem.java and add following code into it.

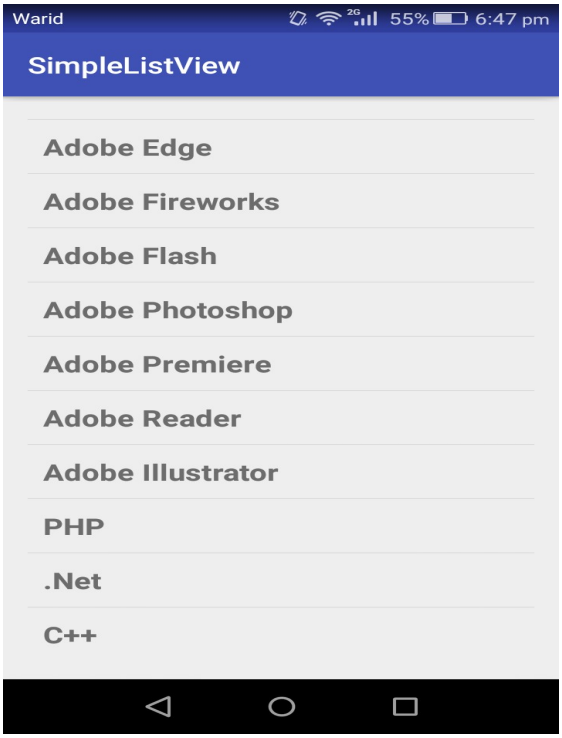
```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.widget.TextView;

/**
 * Created by Hp on 9/20/2016.
 */
public class SingleListItem extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

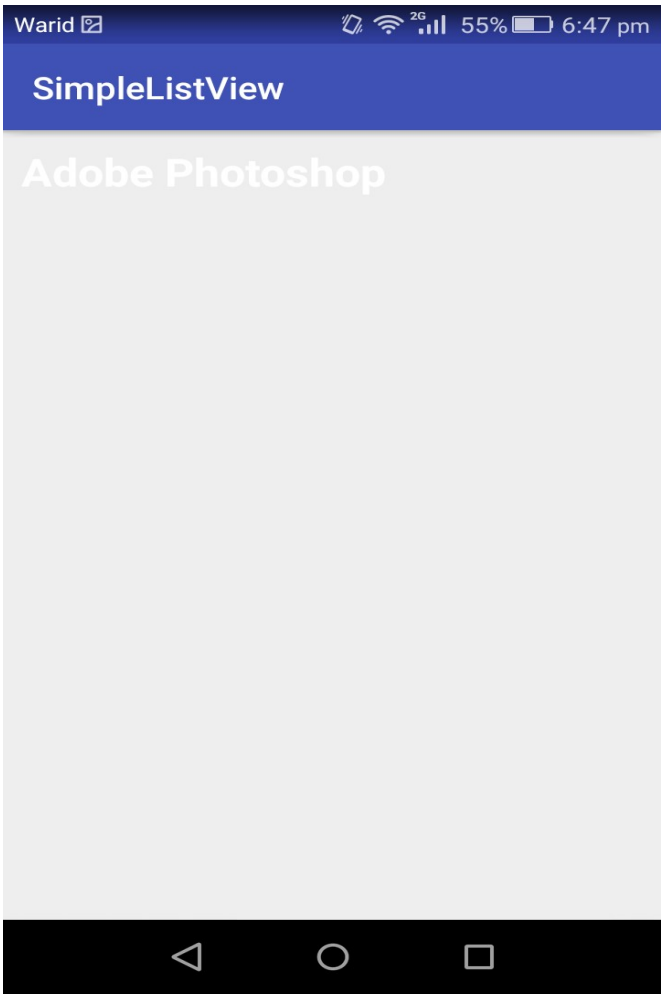
```
        this.setContentView(R.layout.single_list_item_view);

        TextView txtProduct = (TextView) findViewById(R.id.product_label);

        Intent i = getIntent();
        // getting attached intent data
        String product = i.getStringExtra("product");
        // displaying selected product name
        txtProduct.setText(product);

    }
}
```

10) Open AndroidManifest.java in manifest folder and add following line before ending of application tag which is </application>

```
<activity android:name=".SingleListItem"></activity>
```

11) You will see following output.




Warid
55%
6:47 pm
SimpleListView
Adobe Photoshop