ListView

#### **Attributes of ListView:**

Lets see some different attributes of ListView which will be used while designing a custom list:

**1. id:** id is used to uniquely identify a ListView.

Below is the id attribute’s example code with explanation included.

<!-- Id of a list view uniquely identify it-->

<ListView

android:id="@+id/simpleListView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

/>

**2. divider:**This is a drawable or color to draw between different list items.

Below is the divider example code with explanation included, where we draw red color divider between different views.

<!--Divider code in ListView-->

<ListView

android:id="@+id/simpleListView"

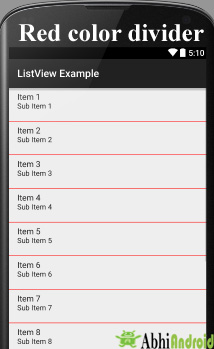
android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:divider="#f00"

android:dividerHeight="1dp"

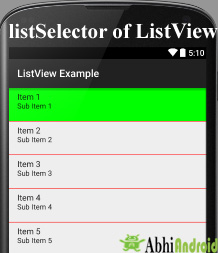
/>



**3. dividerHeight:** This specify the height of the divider between list items. This could be in dp(density pixel),sp(scale independent pixel) or px(pixel).

In above example of divider we also set the divider height 1dp between the list items. The height should be in dp,sp or px.

**4. listSelector:**listSelector property is used to set the selector of the listView. It is generally orange or Sky blue color mostly but you can also define your custom color or an image as a list selector as per your design.



Below is listSelector example code with explanation includes, where list selector color is green, when you select any list item then that item’s background color is green .

**<!-- List Selector Code in ListView -->**

<ListView

android:id="@+id/simpleListView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:divider="#f00"

android:dividerHeight="1dp"

android:listSelector="#0f0"/> <!--list selector in green color-->

#### **Adapters Use in ListView:**

An adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to adapter view then view can takes the data from the adapter view and shows the data on different views like as list view, grid view, spinner etc.

ListView is a subclass of AdapterView and it can be populated by binding  to an Adapter, which retrieves the data from an external source and creates a View that represents each data entry.

**In android commonly used adapters are:**

1. Array Adapter
2. Base Adapter

Now we explain these two adapter in detail:

**1.Array Adapter:**

Whenever you have a list of single items which is backed by an array, you can use ArrayAdapter. For instance, list of phone contacts, countries or names.

**Important Note:** By default, ArrayAdapter expects a Layout with a single TextView, If you want to use more complex views means more customization in list items, please avoid ArrayAdapter and use custom adapters.

Below is Array Adapter code:

ArrayAdapter adapter = new ArrayAdapter<String>(this,R.layout.ListView,R.id.textView,StringArray);

**Example of list view using Custom adapter(Base adapter):**

In this example we display a list of countries with flags. For this, we have to use custom adapter as shown in example:



**Step 1:** Create a new project Listebasexample and activity Main Activity. Here we will create a ListView in LinearLayout. **Below is the code of activity\_main.xml or content\_main.xml:**

<?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">

<ListView

android:id="@+id/simpleListView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:divider="@color/material\_blue\_grey\_800"

android:dividerHeight="1dp"

android:footerDividersEnabled="false" />

</LinearLayout>

**Step 2:** Create a new activity name Listview and below is the code of activity\_listview.xml

<?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="horizontal">

<ImageView

android:id="@+id/icon"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:src="@drawable/ic\_launcher" />

<TextView

android:id="@+id/textView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:padding="@dimen/activity\_horizontal\_margin"

android:textColor="@color/black" />

</LinearLayout>

**Step 3:** In third step we will use custom adapter to display the country names in UI by coding MainActivity.java. **Below is the code of MainActivity.java**

**Important Note:** Make sure flag images are stored in drawable folder present inside res folder with correct naming.

package com.abhiandroid.listbaseexample;

import android.app.Activity;

import android.os.Bundle;

import android.widget.ListView;

public class MainActivity extends Activity {

ListView simpleList;

String countryList[] = {"India", "China", "australia", "Portugle", "America", "NewZealand"};

int flags[] = {R.drawable.india, R.drawable.china, R.drawable.australia, R.drawable.portugle, R.drawable.america, R.drawable.new\_zealand};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

simpleList = (ListView) findViewById(R.id.simpleListView);

CustomAdapter customAdapter = new CustomAdapter(getApplicationContext(), countryList, flags);

simpleList.setAdapter(customAdapter);

}

}

**Step 4:**Now create another class Custom Adapter which will extend BaseAdapter. Below is the code of CustomAdapter.java

package com.abhiandroid.listbaseexample;

import android.content.Context;

import android.media.Image;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.ImageView;

import android.widget.TextView;

import java.util.zip.Inflater;

public class CustomAdapter extends BaseAdapter {

Context context;

String countryList[];

int flags[];

LayoutInflater inflter;

public CustomAdapter(Context applicationContext, String[] countryList, int[] flags) {

this.context = context;

this.countryList = countryList;

this.flags = flags;

inflter = (LayoutInflater.*from*(applicationContext));

}

@Override

public int getCount() {

return countryList.length;

}

@Override

public Object getItem(int i) {

return null;

}

@Override

public long getItemId(int i) {

return 0;

}

@Override

public View getView(int i, View view, ViewGroup viewGroup) {

view = inflter.inflate(R.layout.*activity\_listview*, null);

TextView country = (TextView)           view.findViewById(R.id.*textView*);

ImageView icon = (ImageView) view.findViewById(R.id.*icon*);

country.setText(countryList[i]);

icon.setImageResource(flags[i]);

return view;

}

**Output:**

Now run the App in Emulator and it will show you name of countries along with flags. Below is the output screen:

