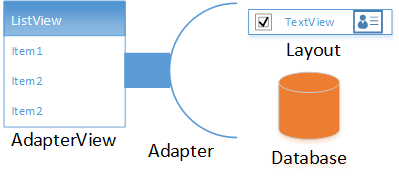
# Android Adapters (and AdapterViews)

Android’s Adapter is described in the API documentation, as “a bridge between an AdapterView and the underlying data for that view/.  An AdapterView is a group of widgets (aka view) components in Android that include the ListView, Spinner, and GridView.  In general, these are the widgets that provide the selecting capability in the user interface.  What is not mentioned in the documentation is that the AdapterView also provides the layout of the underlying data for the view.  The AdapterView really brings together the data and the layout (potentially in a complex collection of views) for each of the rows that make up the AdapterView.



**BaseAdapter**

The source of much confusion comes from the fact that the BaseAdapter is the abstract class that implements the interfaces – all of them!  It implements ListAdapter, SpinnerAdapter and Adatper!  Java does not allow for multiple inheritance among classes, but a class can implement multiple interfaces which is exactly what BaseAdapter does.

BaseAdapter is provided as a convenience to Android developers.  Rather than creating a class the implements Adapter, SpinnerAdapter or ListAdapter, just extend BaseAdapter and implement the specialized methods for either a ListView or Spinner.  What’s more, the BaseAdapter already provides common implementations for many of the interfaces so you don’t have to override these unless there is a specific need.

At a minimum, you will need to implement four methods.  These four methods are called by Android to build your AdapterView and to return the correct information when one of the items in the AdapterView is selected.

* getCount( ):  indicates to Android how many items (or rows) are in the data set that will be presented in the AdapterView.
* getItem(int pos):  get the data item associated with the item (or row) from the AdapterView passed as a parameter to the method.  This method will be used by Android to fetch the appropriate data to build the item/row in the AdapterView.
* getItemId(int pos):  This method returns the data set’s id for a item/row position of the AdapterView.  Typically, the data set id matches the AdapterView rows so this method just returns the same value.
* getView(int position, View convertView, ViewGroup parent):  This method creates the View (which may be a single View component like a TextView or a complex set of widgets in a layout) that displays the data for the specified (by position) item/row in the AdapterView.

Android ListView

Android **ListView** is a view which contains the group of items and displays in a scrollable list. ListView is implemented by importing *android.widget.ListView* class. ListView is a default scrollable which does not use other scroll view.

ListView uses Adapter classes which add the content from data source (such as string array, array, database etc) to ListView. Adapter bridges data between an *AdapterViews* and other Views (ListView, ScrollView etc).

Example of ListView

Let's implement a simple listview example.

***activity\_main.xml***

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

**<android.support.constraint.ConstraintLayout** xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context="listview.example.com.listview.MainActivity"**>**

**<ListView**

        android:id="@+id/listView"

        android:layout\_width="match\_parent"

        android:layout\_height="fill\_parent"

**/>**

**</android.support.constraint.ConstraintLayout>**

mylist.xml

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

<TextView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Medium Text"

android:textStyle="bold"

android:textAppearance="?android:attr/textAppearanceMedium"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="5dp"

android:padding="2dp"

android:textColor="#4d4d4d"

/>

Now place the list of data in strings.xml file by creating string-array.

strings.xml

File:strings.xml

<resources>

<string name="app\_name">ListView</string>

<string-array name="array\_technology">

<item>Android</item>

<item>Java</item>

<item>Php</item>

<item>Hadoop</item>

<item>Sap</item>

<item>Python</item>

<item>Ajax</item>

<item>C++</item>

<item>Ruby</item>

<item>Rails</item>

<item>.Net</item>

<item>Perl</item>

</string-array>

</resources>

Activity class

In java class we need to add adapter to listview using setAdapter() method of listview.

File: MainActivity.java

package listview.example.com.listview;

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 {

ListView listView;

TextView textView;

String[] listItem;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

listView=(ListView)findViewById(R.id.listView);

textView=(TextView)findViewById(R.id.textView);

listItem = getResources().getStringArray(R.array.array\_technology);

final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

android.R.layout.simple\_list\_item\_1, android.R.id.text1, listItem);

listView.setAdapter(adapter);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {

// TODO Auto-generated method stub

String value=adapter.getItem(position);

Toast.makeText(getApplicationContext(),value,Toast.LENGTH\_SHORT).show();

}

});

}

}