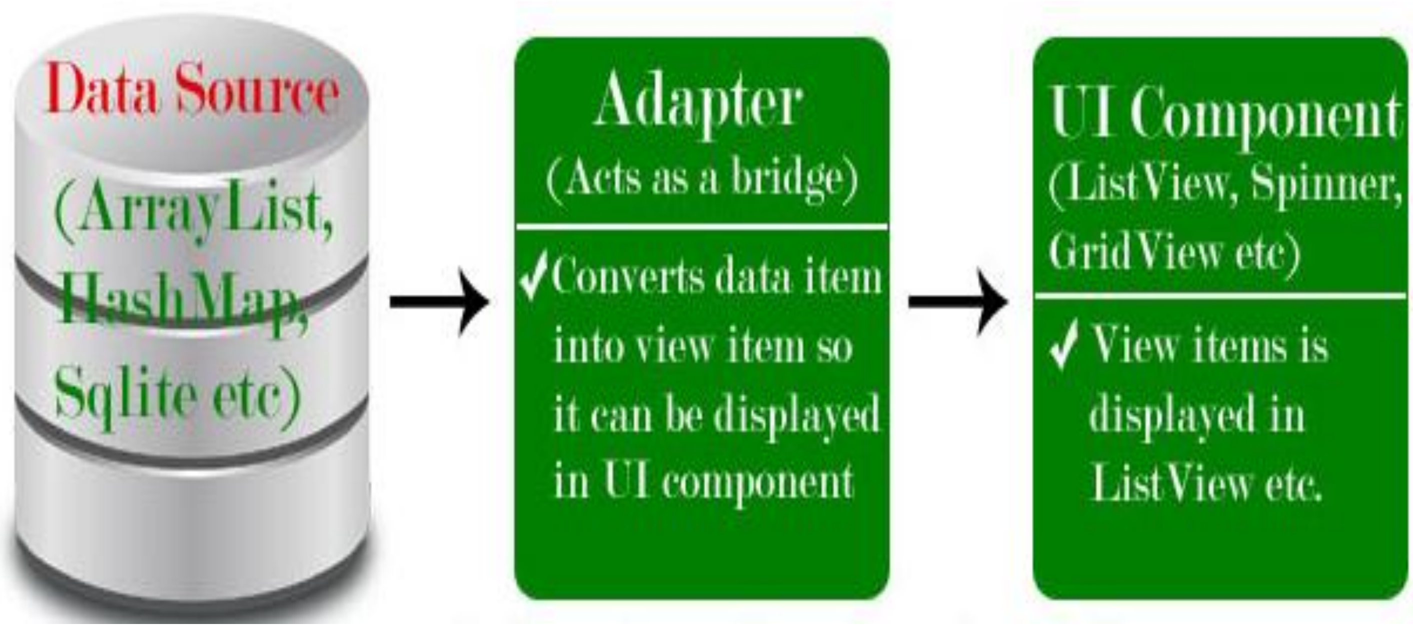
List View

- In Android, 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 an Adapter view then view can takes the data from the adapter view and shows the data on different views like as ListView, GridView, Spinner etc.
- To fill data in a list or a grid we need to implement Adapter. Adapters acts like a bridge between UI component and data source.
- Here data source is the source from where we get the data and UI components are list or grid items in which we want to display that data.

Diagrammatic Explanation



Data Source
(ArrayList, HashMap, Sqlite etc)

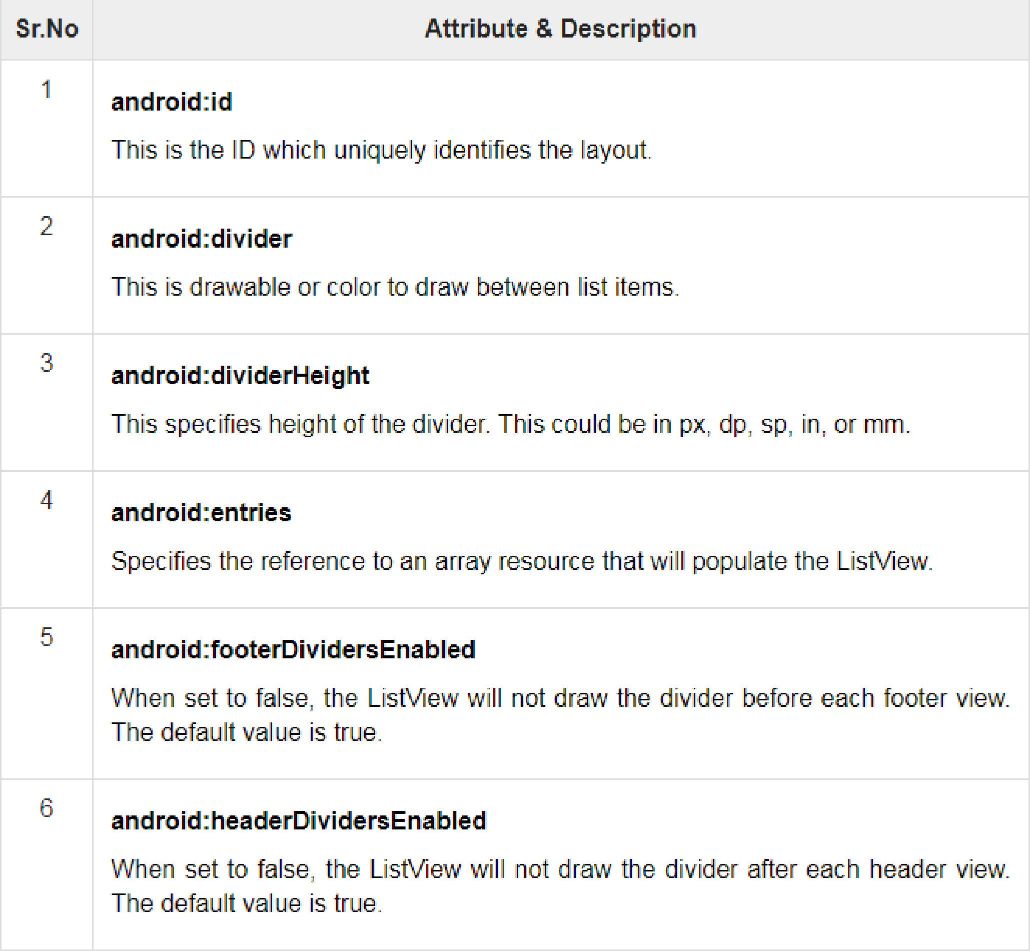
Adapter
(Acts as a bridge)

✓Converts data item into view item so it can be displayed in UI component

UI Component
(ListView, Spinner, GridView etc)

✓ View items is displayed in ListView etc.

List Attributes



Sr.No	Attribute & Description
1	**android:id** This is the ID which uniquely identifies the layout.
2	**android:divider** This is drawable or color to draw between list items.
3	**android:dividerHeight** This specifies height of the divider. This could be in px, dp, sp, in, or mm.
4	**android:entries** Specifies the reference to an array resource that will populate the ListView.
5	**android:footerDividersEnabled** When set to false, the ListView will not draw the divider before each footer view. The default value is true.
6	**android:headerDividersEnabled** When set to false, the ListView will not draw the divider after each header view. The default value is true.

Adapters In Android:

- There are the some commonly used Adapter in Android used to fill the data in the UI components.
- BaseAdapter – It is parent adapter for all other adapters
- ArrayAdapter – It is used whenever we have a list of single items which is backed by an array
- Custom ArrayAdapter – It is used whenever we need to display a custom list
- SimpleAdapter – It is an easy adapter to map static data to views defined in your XML file
- Custom SimpleAdapter – It is used whenever we need to display a customized list and needed to access the child items of the list or grid

SimpleAdapter In Android

- In Android SimpleAdapter is an easy Adapter to map static data to views defined in an XML file(layout). In Android we can specify the data backing to a list as an ArrayList of Maps(i.e. hashmap or other). Each entry in a ArrayList is corresponding to one row of a list.

- The Map contains the data for each row. Here we also specify an XML file(custom list items file) that defines the views which is used to display the row, and a mapping from keys in the Map to specific views.

- **SimpleAdapter is used for customization of list or grid items.**

Parameters used in SimpleAdapter

1. context:

The first parameter is used to pass the context means the reference of current class. Here this is a keyword used to show the current class reference

2. data:

The second parameter is data which is a List of Maps. Each entry in a List corresponds to one row in the list. The Maps contains the data of each row and should include all the entries specified in “from” string array.

3. resource:

The third parameter is resource id which is use to set the layout (xml file) for list items in which you have a text view, image view or any other view.

4. from:

The fourth parameter is **from** is an string array or called a list of column names that will be added to a Map associated with each item of a grid or list view.

5. to:

The fifth and last parameter is **to** which is an integer array used to store the Id’s of the views. The views that should display column in the “from” parameter. These should all be TextViews. The first N views in this list are given the values of the first N columns in the “from” parameter.

MainActivity.java

```
package com.example.mylist;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.HashMap;

public class MainActivity extends AppCompatActivity {


    String[] fruitsNames = {"Apple", "Mango", "Strawberry","Cherry"};
    int[] fruitsImages = {R.drawable.apple, R.drawable.mango, R.drawable.strawberry,  R.drawable.cherry};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView simpleListView=(ListView)findViewById(R.id.simpleListView);

        ArrayList<HashMap<String,String>> arrayList=new ArrayList<>();
        for (int i=0;i<fruitsNames.length;i++)
        {
            HashMap<String,String> hashMap=new HashMap<>();
            hashMap.put("name",fruitsNames[i]);
            hashMap.put("image",fruitsImages[i]+"");
            arrayList.add(hashMap);
        }
```

MainActivity.java…contd

-
  ```
String[] from={"name","image"};
        int[] to={R.id.textView,R.id.imageView};
        SimpleAdapter simpleAdapter=new SimpleAdapter(this,arrayList,R.layout.list_view_items,from,to);
        simpleListView.setAdapter(simpleAdapter);


        simpleListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                Toast.makeText(getApplicationContext(),fruitsNames[i],Toast.LENGTH_LONG).show();
            }
        });
    }


}
```

Activity_main.xml

```
<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"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/simpleListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:divider="#000"
        android:dividerHeight="2dp"
        android:listSelector="#600"/>

</RelativeLayout>
```

list_view_items.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="wrap_content"
    android:background="#fff">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:padding="5dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="10dp"
        android:src="@drawable/ic_launcher" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#8B0000" />
</RelativeLayout>
```