

Android Spinner (Dropdown List)

In android, **Spinner** is a view that allows a user to select one value from the list of values. The spinner in android will behave same as a dropdown list in other programming languages.

Generally, the android spinners will provide a quick way to select one item from the list of values and it will show a dropdown menu with a list of all values when we click or tap on it.

By default, the android spinner will show its currently selected value and by using **Adapter** we can bind the items to spinner objects.

Following is the pictorial representation of using **spinner** in android applications.



We can populate our **Spinner** control with list of choices by defining an **ArrayAdapter** in our [Activity](#) file.

Generally, the **Adapter** pulls data from sources such as an array or database and converts each item into a result view and that's placed into the list.

Android Adapter

In android, **Adapter** will act as an intermediate between the data sources and adapter views such as **ListView**, **GridView** to fill the data into adapter views. The adapter will hold the data and iterates through an items in data set and generate the views for each item in the list.

Generally, in android we have a different type of adapters available to fetch the data from different data sources to fill the data into adapter views, those are

Adapter	Description
ArrayAdapter	It will expect an Array or List as input.
CurosrAdapter	It will accepts an instance of a cursor as an input.
SimpleAdapter	It will accept a static data defined in the resources.
BaseAdapter	It is a generic implementation for all three adapter types and it can be used for ListView, GridView or Spinners based on our requirements

Now we will see how to create spinner or dropdownlist in android applications.

Create Android Spinner in XML Layout File

In android, we can create **Spinner** in XML layout file using **<Spinner>** element with different attributes like as shown below.

```
<Spinner android:id="@+id/spinner1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Populate Android Spinner with Values

To populate spinner with list of values, we need to specify spinner adapter, such as an **ArrayAdapter** in [activity](#) file like as shown below.

```
String[] users = { "Suresh Dasari", "Trishika Dasari", "Rohini Alavala", "Praveen Kumar", "Madhav Sai" };
Spinner spin = (Spinner) findViewById(R.id.spinner1);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
users);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spin.setAdapter(adapter);
```

This is how we can define and bind data to Spinner control in android applications. Now we will see complete example of using spinner control android applications.

Android Spinner Example

Following is the example of defining a one **Spinner** control, one [TextView](#) control in [RelativeLayout](#) to show the list of user details in android application.

Create a new android application using android studio and give names as **SpinnerExample**.

Now open an **activity_main.xml** file from **\res\layout** path and write the code like as shown below

activity_main.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="match_parent">
    <TextView
        android:id="@+id/txtVw"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="150dp"
        android:text="Select User:"
        android:textStyle="bold"
        android:textSize="15dp" />
    <Spinner
        android:id="@+id/spinner1"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/txtVw"
        android:layout_toRightOf="@+id/txtVw" />
</RelativeLayout>

```

If you observe above code we created a one **Spinner** control and one [TextView](#) control in XML Layout file.

Once we are done with the creation of layout with required controls, we need to load the XML layout resource from our [activity onCreate\(\)](#) callback method, for that open main [activity](#) file **MainActivity.java** from `\java\com.tutlane.spinnerexample` path and write the code like as shown below.

MainActivity.java

```

package com.tutlane.spinnerexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {
    String[] users = { "Suresh Dasari", "Trishika Dasari", "Rohini Alavala", "Praveen Kumar", "Madhav Sai" };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spin = (Spinner) findViewById(R.id.spinner1);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
users);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spin.setAdapter(adapter);
        spin.setOnItemSelectedListener(this);
    }
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int position, long id) {
        Toast.makeText(getApplicationContext(), "Selected User: "+users[position]
,Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO - Custom Code
    }
}

```

If you observe the above code we are calling our layout using the **setContentView** method in the form of **R.layout.layout_file_name** in our [activity](#) file. Here our XML file name is **activity_main.xml** so we used file name **activity_main** and binding the list of values to **Spinner** control using **ArrayAdapter**.

Generally, during the launch of our [activity](#), the **onCreate()** callback method will be called by the android framework to get the required layout for an [activity](#).

Output of Android Spinner Example

When we run the above example using an android virtual device (AVD) we will get a result like as shown below.



If you observe above result, our spinner control is like dropdown list in other programming languages and we are able to get the selected user details in android application.

This is how we can use **Spinner** control in android applications to allow users to select one value from the list of values based on our requirements.