

Android RatingBar with Examples

In android, **RatingBar** is a UI control that is used to get the rating from the user. The **RatingBar** is an extension of [SeekBar](#) and [ProgressBar](#) that shows a rating in stars and it allows users to set the rating value by touch or click on the stars.

The android **RatingBar** will always return a rating value as a floating-point number such as 1.0, 2.0, 2.5, 3.0, 3.5, etc.

Following is the pictorial representation of using a RatingBar in android applications.



In android, by using **android:numStars** attribute we can define the number of stars to display in **RatingBar**. An example of using RatingBar is in movie sites or product sites to collect the user rating about the movies or products, etc.

In android, by using **android.widget.RatingBar** component we can display the rating bar with star icons.

Create Android RatingBar in XML Layout File

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

```
<RatingBar
    android:id="@+id/ratingBar1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:rating="3.5"/>
```

If you observe above code snippet, we defined a rating bar (**<RatingBar>**) with different attributes, those are

Attribute	Description
android:id	It is used to uniquely identify the control
android:numStars	It is used to define the number of stars to display.
android:rating	It is used to set the default rating value for the rating bar.

Now we will see how to get the rating value from RatingBar control in android applications.

Get Android RatingBar Value

In android, by using **RatingBar** methods (**getNumStars()**, **getRating()**) we can get the number of stars and the rating value which was selected.

Following is the code snippet to get the rating details from RatingBar in android applications.

```
int noofstars = rBar.getNumStars();
float getrating = rBar.getRating();
tView.setText("Rating: "+getrating+"/"+noofstars);
```

This is how we can get the number of stars in RatingBar control and the selected rating value from RatingBar control in android applications.

Android RatingBar Control Attributes

Following are some of the commonly used attributes related to **RatingBar** control in android applications.

Attribute	Description
android:id	It is used to uniquely identify the control
android:numStars	It is used to define a number of stars to display.
android:rating	It is used to set the default rating value for the rating bar.
android:background	It is used to set the background color for a progress bar.
android:padding	It is used to set the padding for left, right, top or bottom of a progress bar.

Android RatingBar Control Example

Following is the example of defining a **RatingBar** control, [Button](#) control and [TextView](#) control in [RelativeLayout](#) to get the selected rating value from RatingBar on [Button](#) click.

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

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">
    <RatingBar
        android:id="@+id/ratingBar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="80dp"
        android:layout_marginTop="200dp"
        android:numStars="5"
        android:rating="3.5"/>
    <Button
        android:id="@+id/btnGet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

    android:layout_alignLeft="@+id/ratingBar1"
    android:layout_below="@+id/ratingBar1"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="60dp"
    android:text="Get Rating"/>
<TextView
    android:id="@+id/textview1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/btnGet"
    android:layout_below="@+id/btnGet"
    android:layout_marginTop="20dp"
    android:textSize="20dp"
    android:textStyle="bold"/>
</RelativeLayout>

```

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

Once we are done with 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.ratingbarexample` path and write the code like as shown below.

MainActivity.java

```

package com.tutlane.ratingbarexample;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.RatingBar;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private RatingBar rBar;
    private TextView tView;
    private Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rBar = (RatingBar) findViewById(R.id.ratingBar1);
        tView = (TextView) findViewById(R.id.textview1);
        btn = (Button) findViewById(R.id.btnGet);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int noofstars = rBar.getNumStars();
                float getrating = rBar.getRating();
                tView.setText("Rating: "+getrating+"/"+noofstars);
            }
        });
    }
}

```

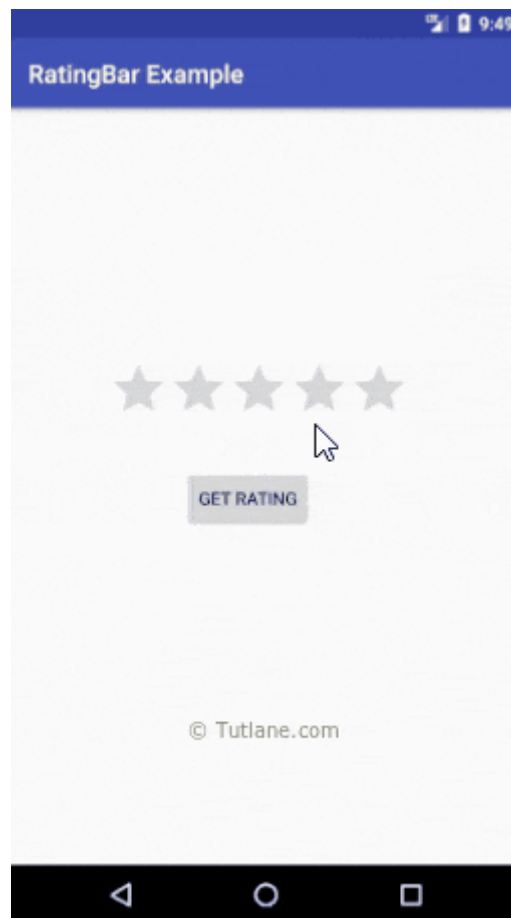
```
    });  
  }  
}
```

If you observe above code we are calling our layout using **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 we are trying to get the number of stars in RatingBar and the selected rating value from RatingBar control.

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

Output of Android RatingBar Example

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



If you observe above result, we are able to get the rating value from the **RatingBar** control when we click on [Button](#) in android application.

This is how we can use **RatingBar** control in android applications to show the ratings based on our requirements.