




Android UI Control

[RatingBar]

Android – UI Control

RatingBar

- **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.
- 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.
- By using **android.widget.RatingBar** component we can display the rating bar with star icons.

Android – UI Control

Create Android RatingBar in XML Layout File

- 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.

3

Android – UI Control

Get Android RatingBar Value

- 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();
textView.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.

4

Android – UI Control

Android RatingBar Control Attributes

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 rating bar.
android:padding	It is used to set the padding for left, right, top or bottom of a rating bar.

5

Android – UI Control

Android RatingBar Control Example

- In this example we define a **RatingBar** control, one **Button** 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

6

Android – UI Control

Android RatingBar Control Example

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>
```

7

Android – UI Control

Android RatingBar Control Example

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);
```

8

Android – UI Control

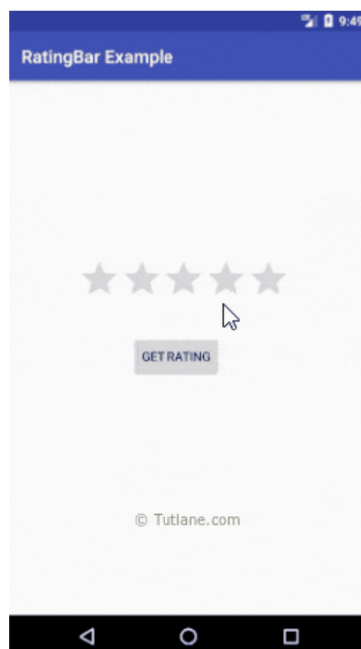
Android RatingBar Control Example

```
btn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        int noofstars = rBar.getNumStars();  
        float getrating = rBar.getRating();  
        textView.setText("Rating: "+getrating+"/"+noofstars);  
    }  
});  
}
```

9

Android – UI Control

Output of Android RatingBar Example



10

