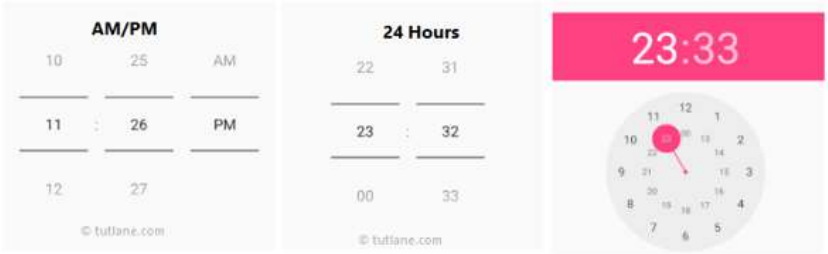
Android UI Control
[TimePicker]

Android – UI Control

TimePicker

- **TimePicker** is a widget for selecting the time of day, in either 24-hour or AM/PM mode.
- If we use **TimePicker** in our application, it will ensure that the users will select a valid time for the day.
- Following is the pictorial representation of using a timepicker control in android applications.



- TimePicker available in two modes, one is to show the time in clock mode and another one is to show the time in spinner mode.

Android – UI Control

Create Android TimePicker in XML Layout File

- We can create a TimePicker in XML layout file using **<TimePicker>** element with different attributes like as shown below

```
<TimePicker android:id="@+id/timePicker1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content" />
```


Android – UI Control

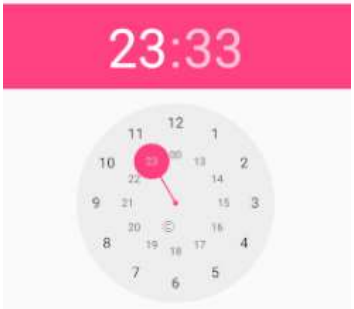
Android TimePicker with Clock Mode

- We can define android TimePicker to show time in clock format by using TimePicker android:timePickerMode attribute.

- Following is the example of showing the TimePicker in **Clock** mode.

```
<TimePicker android:id="@+id/timePicker1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:timePickerMode="clock" />
```

- The above code will return the TimePicker like as shown below.



Android – UI Control

Android TimePicker with Spinner Mode

- If we want to show the TimePicker in spinner format like showing hours and minutes separately to select the time, then by using TimePicker **android:timePickerMode** attribute we can achieve this.

- Following is the example of showing the TimePicker in spinner mode.

```
<TimePicker
  android:id="@+id/datePicker1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:timePickerMode="spinner"/>
```

- The above code will return the TimePicker like as shown below

Android – UI Control

Android TimePicker with Spinner Mode

- We can change the TimePicker in spinner mode to **AM** / **PM** format instead of **24 Hours** format by using the **setIs24HourView(true)** method in activity file like as shown below.

```
TimePicker picker=(TimePicker)findViewById(R.id.timePicker1);
picker.setIs24HourView(true);
```

- The above code will return the TimePicker like as shown below

Android – UI Control

Android TimePicker Control Attributes

Attribute	Description
android:id	It is used to uniquely identify the control
android:padding	It is used to set the padding for left, right, top or bottom of the time picker.
android:timePickerMode	It is used to specify timepicker mode, either spinner or clock
android:background	It is used to set the background color for the time picker.

Android – UI Control

Android TimePicker Example

- In this example we define one **TimePicker** control, one **TextView** control and one **Button** control in **RelativeLayout** to show the selected time in **AM/PM** format on **Button** click in the android application.

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

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

Android – UI Control

Android TimePicker 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">
   <TimePicker
      android:id="@+id/timePicker1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="20dp" />
   <Button
      android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/timePicker1"
      android:layout_marginTop="10dp"
      android:layout_marginLeft="160dp"
      android:text="Get Date" />
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/button1"
      android:layout_marginLeft="120dp"
      android:layout_marginTop="10dp"
      android:textStyle="bold"
      android:textSize="18dp"/>
</RelativeLayout>
```


Android – UI Control

Android TimePicker Example

MainActivity.java

```
package com.tutlane.timepickerexample;
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends AppCompatActivity {
   TimePicker picker;
   Button btnGet;
   TextView tvw;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      tvw=(TextView)findViewById(R.id.textView1);
      picker=(TimePicker)findViewById(R.id.timePicker1);
```

Android TimePicker Example

```
picker.setIs24HourView(true);
btnGet=(Button)findViewById(R.id.button1);
btnGet.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      int hour, minute;
      String am_pm;
      if (Build.VERSION.SDK_INT >= 23 ){
         hour = picker.getHour();
         minute = picker.getMinute();
      }
      else{
         hour = picker.getCurrentHour();
         minute = picker.getCurrentMinute();
      }
```


Android TimePicker Example

```
      if(hour > 12) {
            am_pm = "PM";
            hour = hour - 12;
         }
         else
         {
            am_pm="AM";
         }
         tvw.setText("Selected Date: "+ hour +":"+ minute+" "+am_pm);
      }
   });
  }
}
```

Output of Android TimePicker Example