Android - MediaPlayer Tutorial

Previous Page Next Page

Android provides many ways to control playback of audio/video files and streams. One of this way is through a class called **MediaPlayer**.

Android is providing MediaPlayer class to access built-in mediaplayer services like playing audio,video e.t.c. In order to use MediaPlayer , we have to call a static Method **create()** of this class. This method returns an instance of MediaPlayer class. Its syntax is as follows –

```
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.song);
```

The second parameter is the name of the song that you want to play. You have to make a new folder under your project with name **raw** and place the music file into it.

Once you have created the Mediaplayer object you can call some methods to start or stop the music. These methods are listed below.

```
mediaPlayer.start();
mediaPlayer.pause();
```

On call to **start()** method, the music will start playing from the beginning. If this method is called again after the **pause()** method , the music would start playing from where it is left and not from the beginning.

In order to start music from the beginning , you have to call **reset()** method. Its syntax is given below.

```
mediaPlayer.reset();
```

Apart from the start and pause method, there are other methods provided by this class for better dealing with audio/video files. These methods are listed below –

Sr.No	Method & description
1	**isPlaying()** This method just returns true/false indicating the song is playing or not

2	**seekTo(position)** This method takes an integer, and move song to that particular second
3	**getCurrentDuration()** This method returns the current position of song in milliseconds
4	**getDuration()** This method returns the total time duration of song in milliseconds
5	**reset()** This method resets the media player
6	**release()** This method releases any resource attached with MediaPlayer object
7	**setVolume(float leftVolume, float rightVolume)** This method sets the up down volume for this player
8	**setDataSource(FileDescriptor fd)** This method sets the data source of audio/video file
9	**selectTrack(int index)** This method takes an integer, and select the track from the list on that particular index
10	**getTrackInfo()** This method returns an array of track information

Example

Here is an example demonstrating the use of MediaPlayer class. It creates a basic media player that allows you to forward, backward , play and pause a song.

To experiment with this example, you need to run this on an actual device to hear the audio sound.

Steps	Description
1	You will use Android studio IDE to create an Android application under a package com.example.sairamkrishna.myapplication;. While creating this project, make sure

	you Target SDK and Compile With at the latest version of Android SDK to use higher levels of APIs.
2	Modify src/MainActivity.java file to add MediaPlayer code.
3	Modify the res/layout/activity_main to add respective XML components
4	Create a new folder under MediaPlayer with name as raw and place an mp3 music file in it with name as song.mp3
5	Run the application and choose a running android device and install the application on it and verify the results

Following is the content of the modified main activity file **src/MainActivity.java**.

```
package com.example.sairamkrishna.myapplication;

import android.app.Activity;
import android.app.Activity;

import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.media.MediaPlayer;

import android.os.Bundle;
import android.os.Handler;

import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import java.util.concurrent.TimeUnit;


public class MainActivity extends Activity {
   private Button b1,b2,b3,b4;
   private ImageView iv;
   private MediaPlayer mediaPlayer;
   private double startTime = 0;
   private double finalTime = 0;
   private Handler myHandler = new Handler();;
   private int forwardTime = 5000;
   private int backwardTime = 5000;
   private SeekBar seekbar;
   private TextView tx1,tx2,tx3;

   public static int oneTimeOnly = 0;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      b1 = (Button) findViewById(R.id.button);
      b2 = (Button) findViewById(R.id.button2);
      b3=(Button)findViewById(R.id.button3);
```

```
b4=(Button)findViewById(R.id.button4);
iv=(ImageView)findViewById(R.id.imageView);

tx1=(TextView)findViewById(R.id.textView2);
tx2=(TextView)findViewById(R.id.textView3);
tx3=(TextView)findViewById(R.id.textView4);
tx3.setText("Song.mp3");

mediaPlayer = MediaPlayer.create(this, R.raw.song);
seekbar=(SeekBar)findViewById(R.id.seekBar);
seekbar.setClickable(false);
b2.setEnabled(false);

b3.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Toast.makeText(getApplicationContext(), "Playing sound",Toast.LENGTH_SHORT).show()
      mediaPlayer.start();

      finalTime = mediaPlayer.getDuration();
      startTime = mediaPlayer.getCurrentPosition();

      if (oneTimeOnly == 0) {
         seekbar.setMax((int) finalTime);
         oneTimeOnly = 1;
      }
      tx2.setText(String.format("%d min, %d sec",
      TimeUnit.MILLISECONDS.toMinutes((long) finalTime),
      TimeUnit.MILLISECONDS.toSeconds((long) finalTime) -
      TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes((long) finalTime)))
      );

      tx1.setText(String.format("%d min, %d sec",
      TimeUnit.MILLISECONDS.toMinutes((long) startTime),
      TimeUnit.MILLISECONDS.toSeconds((long) startTime) -
      TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes((long) startTime)))
      );

      seekbar.setProgress((int)startTime);
      myHandler.postDelayed(UpdateSongTime,100);
      b2.setEnabled(true);
      b3.setEnabled(false);
   }
});

b2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Toast.makeText(getApplicationContext(), "Pausing sound",Toast.LENGTH_SHORT).show()
      mediaPlayer.pause();
      b2.setEnabled(false);
      b3.setEnabled(true);
   }
});

b1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      int temp = (int)startTime;

      if((temp+forwardTime)<=finalTime){
         startTime = startTime + forwardTime;
         mediaPlayer.seekTo((int) startTime);
         Toast.makeText(getApplicationContext(),"You have Jumped forward 5 seconds",Toas
      }
```

```
            else{
               Toast.makeText(getApplicationContext(),"Cannot jump forward 5 seconds",Toast.LE
            }
         }
      });

      b4.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            int temp = (int)startTime;

            if((temp-backwardTime)>0){
               startTime = startTime - backwardTime;
               mediaPlayer.seekTo((int) startTime);
               Toast.makeText(getApplicationContext(),"You have Jumped backward 5 seconds",Toa
            }
            else{
               Toast.makeText(getApplicationContext(),"Cannot jump backward 5 seconds",Toast.L
            }
         }
      });
   }

   private Runnable UpdateSongTime = new Runnable() {
      public void run() {
         startTime = mediaPlayer.getCurrentPosition();
         tx1.setText(String.format("%d min, %d sec",

         TimeUnit.MILLISECONDS.toMinutes((long) startTime),
         TimeUnit.MILLISECONDS.toSeconds((long) startTime) -
         TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.
         toMinutes((long) startTime)))
         );
         seekbar.setProgress((int)startTime);
         myHandler.postDelayed(this, 100);
      }
   };

   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      // Inflate the menu; this adds items to the action bar if it is present.
      getMenuInflater().inflate(R.menu.menu_main, menu);
      return true;
   }

   @Override
   public boolean onOptionsItemSelected(MenuItem item) {
      // Handle action bar item clicks here. The action bar will
      // automatically handle clicks on the Home/Up button, so long
      // as you specify a parent activity in AndroidManifest.xml.

      int id = item.getItemId();

      //noinspection SimplifiableIfStatement

      if (id == R.id.action_settings) {
         return true;
      }
      return super.onOptionsItemSelected(item);
   }
}
```

Following is the modified content of the xml **res/layout/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" android:paddingLeft="@dimen/activity_horizontal_margin
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

   <TextView android:text="Music Palyer" android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/textview"
      android:textSize="35dp"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Tutorials point"
      android:id="@+id/textView"
      android:layout_below="@+id/textview"
      android:layout_centerHorizontal="true"
      android:textColor="#ff7aff24"
      android:textSize="35dp" />

   <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/imageView"
      android:layout_below="@+id/textView"
      android:layout_centerHorizontal="true"
      android:src="@drawable/abc"/>

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text=">>"
      android:id="@+id/button"
      android:layout_alignParentBottom="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true" />

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="||"
      android:id="@+id/button2"
      android:layout_alignParentBottom="true"
      android:layout_alignLeft="@+id/imageView"
      android:layout_alignStart="@+id/imageView" />

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="<"
      android:id="@+id/button3"
      android:layout_alignTop="@+id/button2"
      android:layout_toRightOf="@+id/button2"
      android:layout_toEndOf="@+id/button2" />

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="<<"
      android:id="@+id/button4"
```

```
      android:layout_alignTop="@+id/button3"
      android:layout_toRightOf="@+id/button3"
      android:layout_toEndOf="@+id/button3" />

   <SeekBar
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/seekBar"
      android:layout_alignLeft="@+id/textview"
      android:layout_alignStart="@+id/textview"
      android:layout_alignRight="@+id/textview"
      android:layout_alignEnd="@+id/textview"
      android:layout_above="@+id/button" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceSmall"
      android:text="Small Text"
      android:id="@+id/textView2"
      android:layout_above="@+id/seekBar"
      android:layout_toLeftOf="@+id/textView"
      android:layout_toStartOf="@+id/textView" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceSmall"
      android:text="Small Text"
      android:id="@+id/textView3"
      android:layout_above="@+id/seekBar"
      android:layout_alignRight="@+id/button4"
      android:layout_alignEnd="@+id/button4" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceMedium"
      android:text="Medium Text"
      android:id="@+id/textView4"
      android:layout_alignBaseline="@+id/textView2"
      android:layout_alignBottom="@+id/textView2"
      android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Following is the content of the **res/values/string.xml**.

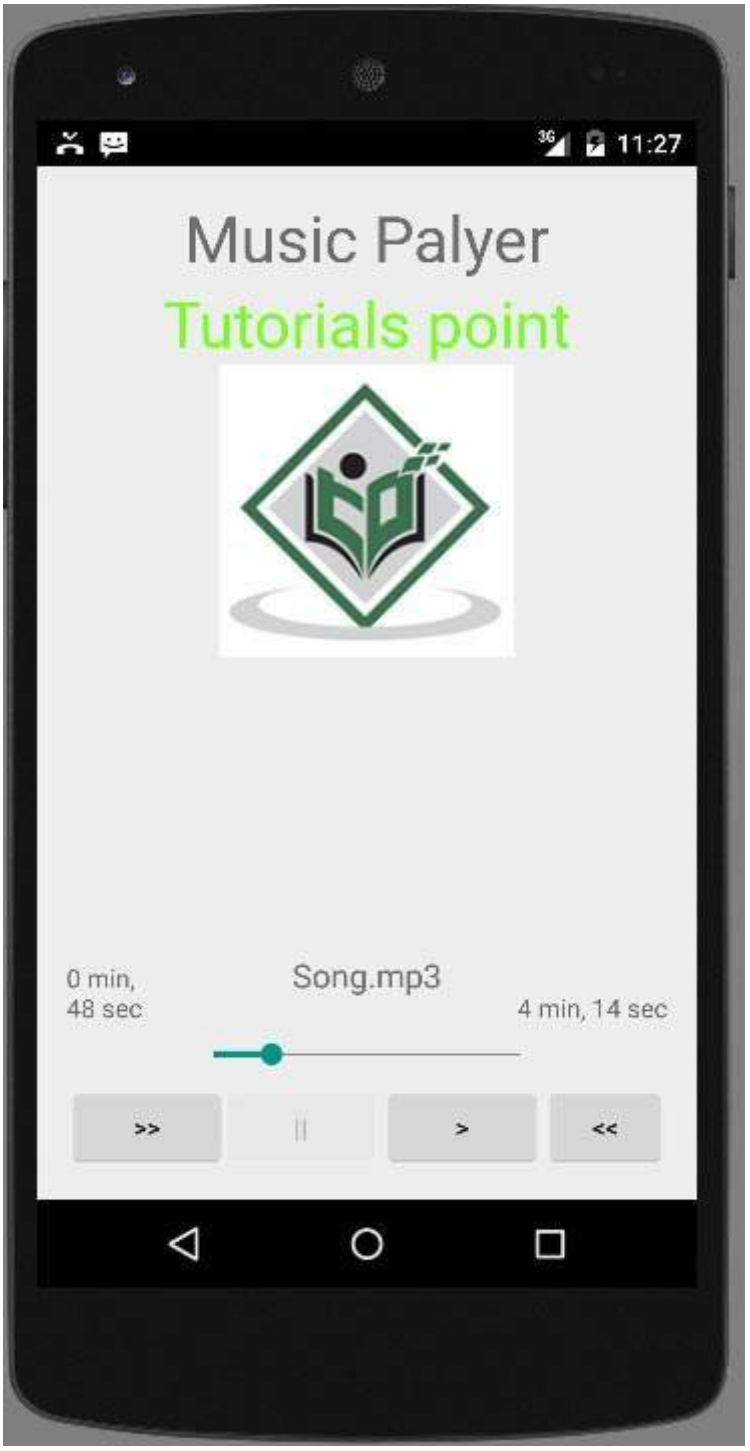
```
<resources>
   <string name="app_name">My Application</string>
   <string name="hello_world">Hello world!</string>
   <string name="action_settings">Settings</string>
</resources>
```

Following is the content of **AndroidManifest.xml** file.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.mediaplayer"
   android:versionCode="1"
   android:versionName="1.0" >
```

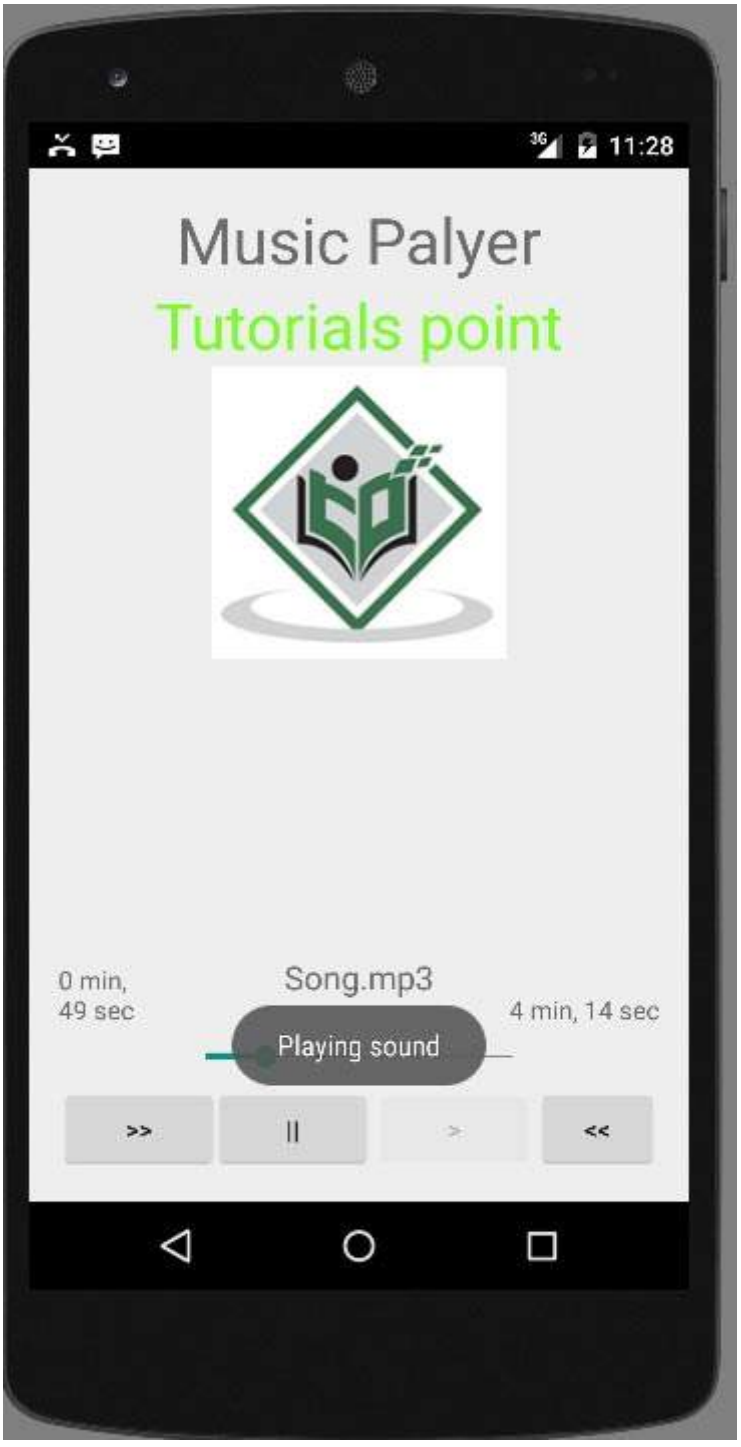
```
   <uses-sdk
      android:minSdkVersion="13"
      android:targetSdkVersion="22" />

   <application
      android:allowBackup="true"
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >

      <activity
         android:name="com.example.sairamkrishna.myapplication.MainActivity"
         android:label="@string/app_name" >

         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>

      </activity>

   </application>
</manifest>
```

Let's try to run your application. I assume you have connected your actual Android Mobile device with your computer. To run the app from Eclipse, open one of your project's activity files and click Run icon from the toolbar. Before starting your application, Android studio will display following screens

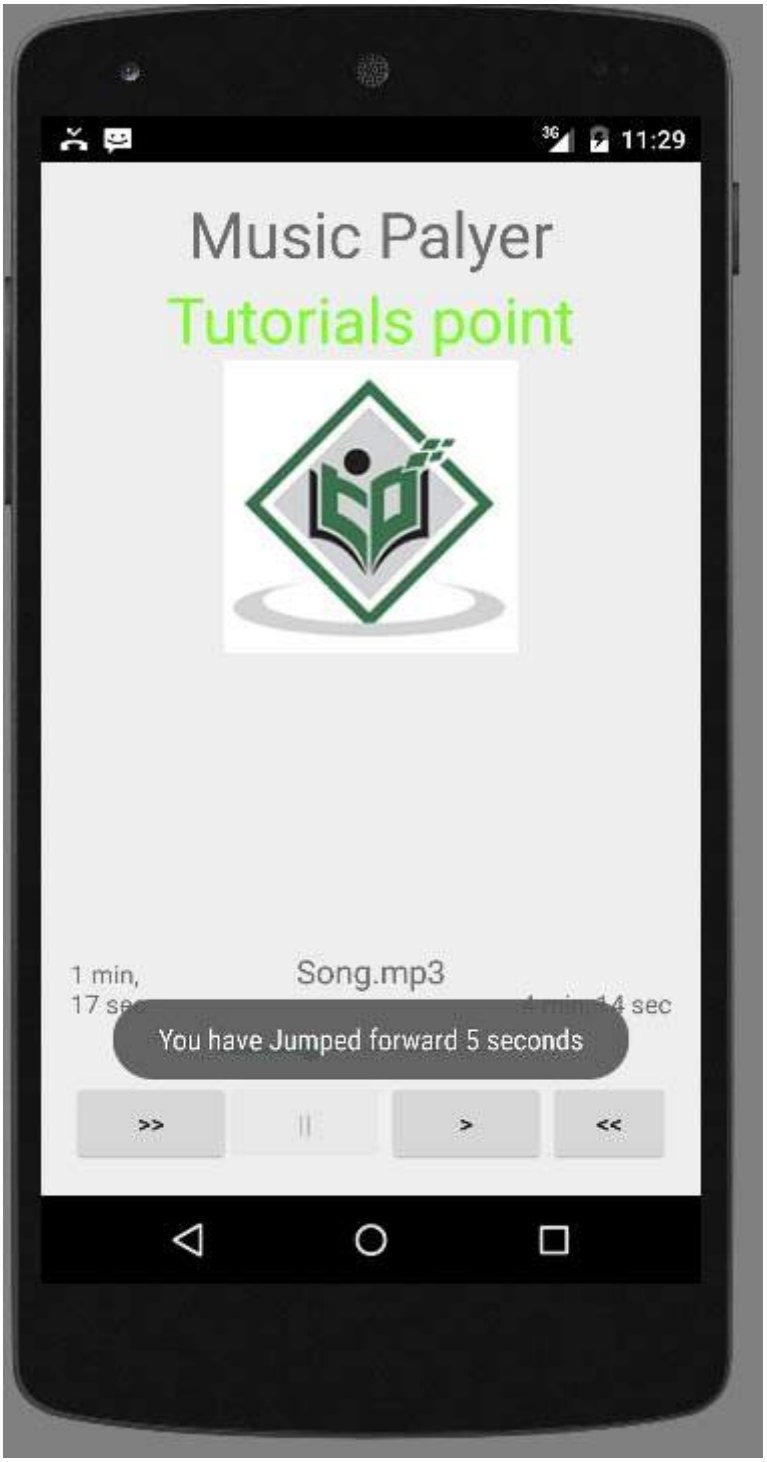


By default you would see the pause button disabled. Now press play button and it would become disable and pause button become enable. It is shown in the picture below –

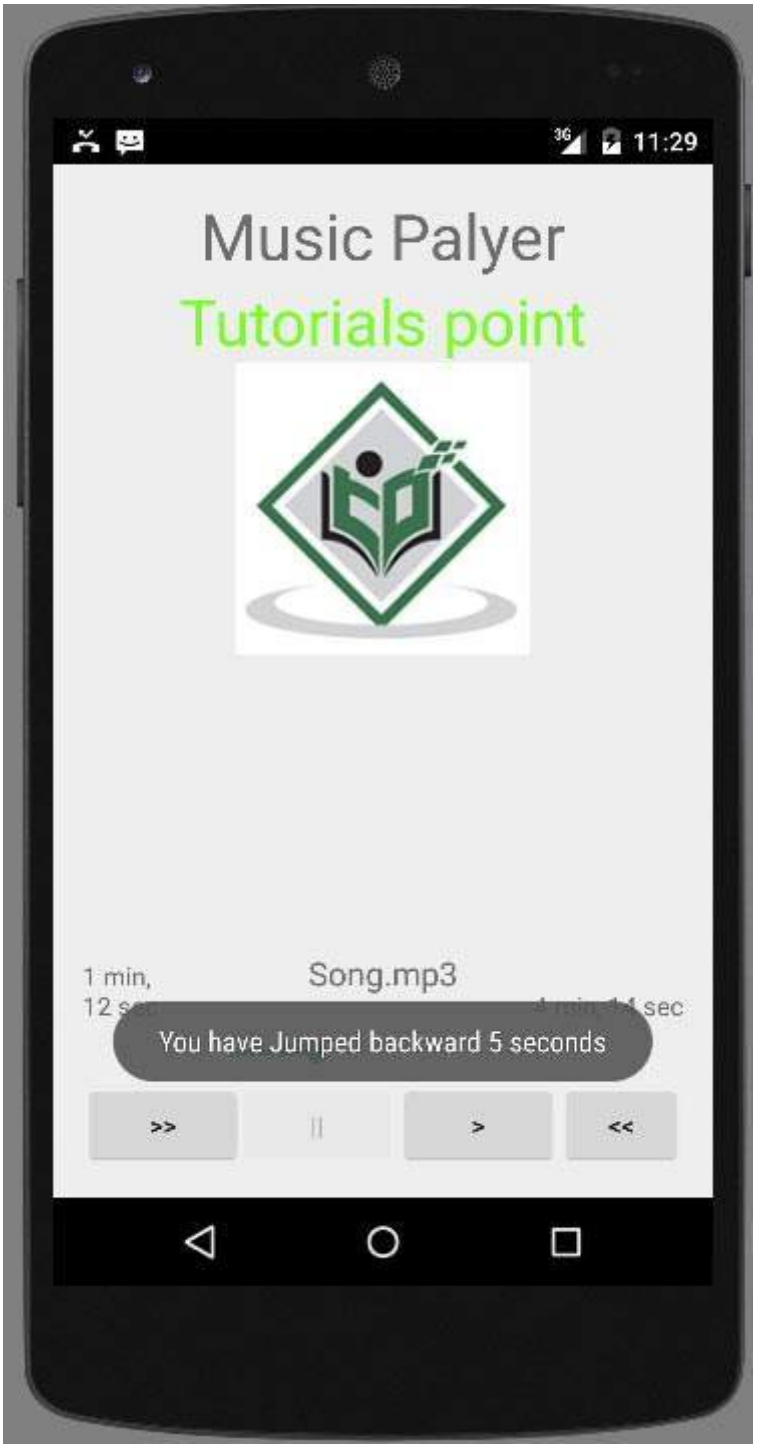
Up till now, the music has been playing. Now press the pause button and see the pause notification. This is shown below –



Now when you press the play button again, the song will not play from the beginning but from where it was paused. Now press the fast forward or backward button to jump the song forward or backward 5 seconds. A time came when the song cannot be jump forward. At this point , the notification would appear which would be something like this –



Your music would remain playing in the background while you are doing other tasks in your mobile. In order to stop it , you have to exit this application from background activities.



Above image shows when you pick rewind button.

Previous Page Next Page

Enter email for newsletter go