Android - Bluetooth Tutorial

Previous Page Next Page

Among many ways, Bluetooth is a way to send or receive data between two different devices. Android platform includes support for the Bluetooth framework that allows a device to wirelessly exchange data with other Bluetooth devices.

Android provides Bluetooth API to perform these different operations.

- Scan for other Bluetooth devices
- Get a list of paired devices
- Connect to other devices through service discovery

Android provides BluetoothAdapter class to communicate with Bluetooth. Create an object of this calling by calling the static method getDefaultAdapter(). Its syntax is given below.

```
private BluetoothAdapter BA;
BA = BluetoothAdapter.getDefaultAdapter();
```

In order to enable the Bluetooth of your device, call the intent with the following Bluetooth constant ACTION_REQUEST_ENABLE. Its syntax is.

```
Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(turnOn, 0);
```

Apart from this constant, there are other constants provided the API , that supports different tasks. They are listed below.

Sr.No	Constant & description
1	**ACTION_REQUEST_DISCOVERABLE** This constant is used for turn on discovering of bluetooth

2	**ACTION_STATE_CHANGED** This constant will notify that Bluetooth state has been changed
3	**ACTION_FOUND** This constant is used for receiving information about each device that is discovered

Once you enable the Bluetooth , you can get a list of paired devices by calling getBondedDevices() method. It returns a set of bluetooth devices. Its syntax is.

```
private Set<BluetoothDevice>pairedDevices;
pairedDevices = BA.getBondedDevices();
```

Apart form the parried Devices , there are other methods in the API that gives more control over Blueetooth. They are listed below.

Sr.No	Method & description
1	**enable()** This method enables the adapter if not enabled
2	**isEnabled()** This method returns true if adapter is enabled
3	**disable()** This method disables the adapter
4	**getName()** This method returns the name of the Bluetooth adapter
5	**setName(String name)** This method changes the Bluetooth name
6	**getState()** This method returns the current state of the Bluetooth Adapter.
7	**startDiscovery()** This method starts the discovery process of the Bluetooth for 120 seconds.

Example

This example provides demonstration of BluetoothAdapter class to manipu Bluetooth and show list of paired devices by the Bluetooth.

To experiment with this example , you need to run this on an actual device.

Steps	Description
1	You will use Android studio to create an Android application 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 the code
3	Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4	Modify AndroidManifest.xml to add necessary permissions.
5	Run the application and choose a running android device and install the application on it and verify the results.

Here is the content of **src/MainActivity.java**

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

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;

import android.graphics.Color;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;

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

import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;

public class MainActivity extends Activity  {
   Button b1,b2,b3,b4;
   private BluetoothAdapter BA;
   private Set<BluetoothDevice>pairedDevices;
   ListView lv;

   @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);

      BA = BluetoothAdapter.getDefaultAdapter();
      lv = (ListView)findViewById(R.id.listView);
   }

   public void on(View v){
      if (!BA.isEnabled()) {
         Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
         startActivityForResult(turnOn, 0);
         Toast.makeText(getApplicationContext(),"Turned on",Toast.LENGTH_LONG).show();
      }
      else
      {
         Toast.makeText(getApplicationContext(),"Already on", Toast.LENGTH_LONG).show();
      }
   }

   public void off(View v){
      BA.disable();
      Toast.makeText(getApplicationContext(),"Turned off" ,Toast.LENGTH_LONG).show();
   }

   public  void visible(View v){
      Intent getVisible = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
      startActivityForResult(getVisible, 0);
   }

   public void list(View v){
      pairedDevices = BA.getBondedDevices();
      ArrayList list = new ArrayList();

      for(BluetoothDevice bt : pairedDevices)
      list.add(bt.getName());
      Toast.makeText(getApplicationContext(),"Showing Paired Devices",Toast.LENGTH_SHORT).show(

      final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.simple_list_item_1, l
      lv.setAdapter(adapter);
   }

   @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);
   }
}
```

Here is the content of **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"
   android:transitionGroup="true">

   <TextView android:text="Bluetooth Example" 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:src="@drawable/abc"
      android:layout_below="@+id/textView"
      android:layout_centerHorizontal="true"
      android:theme="@style/Base.TextAppearance.AppCompat" />
```

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Turn On"
        android:id="@+id/button"
        android:layout_below="@+id/imageView"
        android:layout_toStartOf="@+id/imageView"
        android:layout_toLeftOf="@+id/imageView"
        android:clickable="true"
        android:onClick="on" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Get visible"
        android:onClick="visible"
        android:id="@+id/button2"
        android:layout_alignBottom="@+id/button"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="List devices"
        android:onClick="list"
        android:id="@+id/button3"
        android:layout_below="@+id/imageView"
        android:layout_toRightOf="@+id/imageView"
        android:layout_toEndOf="@+id/imageView" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="turn off"
        android:onClick="off"
        android:id="@+id/button4"
        android:layout_below="@+id/button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_alignParentBottom="true"
        android:layout_alignLeft="@+id/button"
        android:layout_alignStart="@+id/button"
        android:layout_below="@+id/textView2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Paired devices:"
        android:id="@+id/textView2"
        android:textColor="#ff34ff06"
        android:textSize="25dp"
        android:layout_below="@+id/button4"
        android:layout_alignLeft="@+id/listView"
        android:layout_alignStart="@+id/listView" />

</RelativeLayout>
```

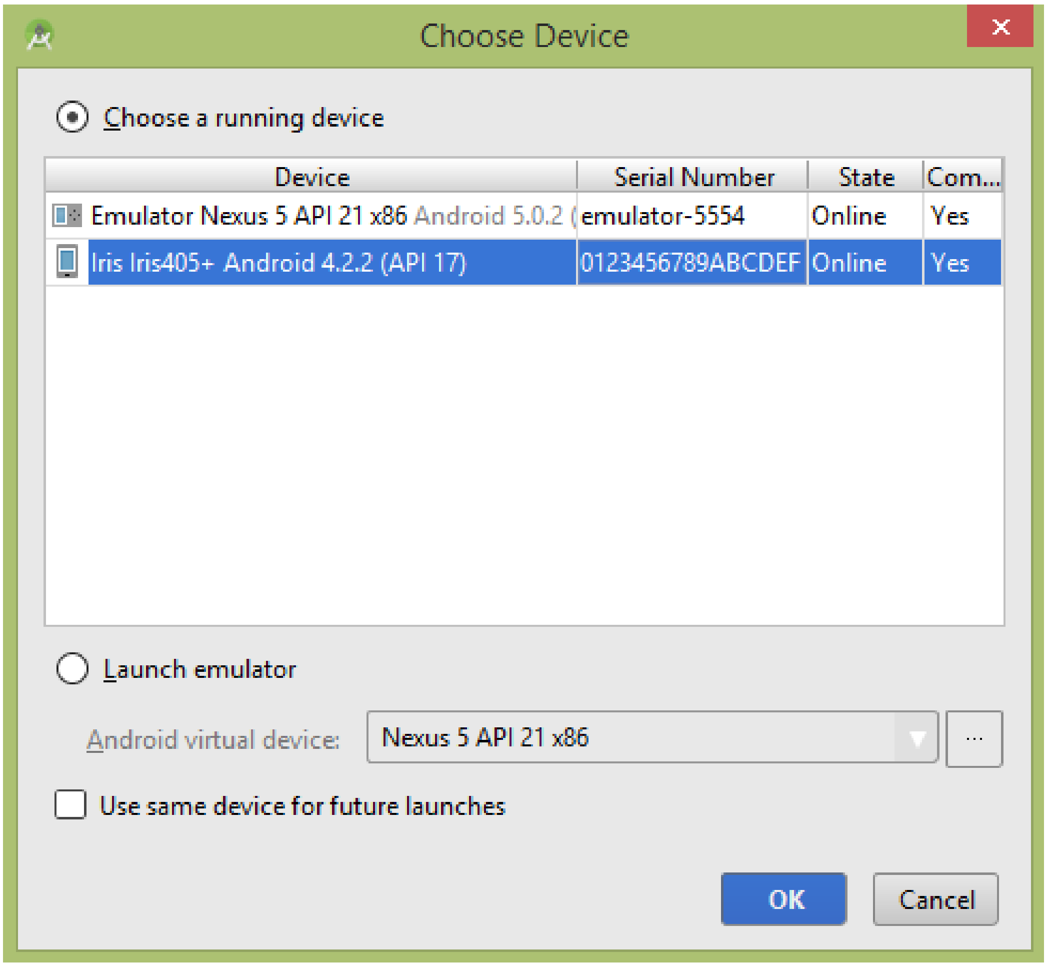
Here is the content of **Strings.xml**

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

Here is the content of **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.sairamkrishna.myapplication" >
   <uses-permission android:name="android.permission.BLUETOOTH"/>
   <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

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

      <activity
         android:name=".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 Android studio, open one of your project's activity files and click Run ▶ icon from the tool bar. Before starting your application, Android studio will display following window to select an option where you want to run your Android application.



● Choose a running device

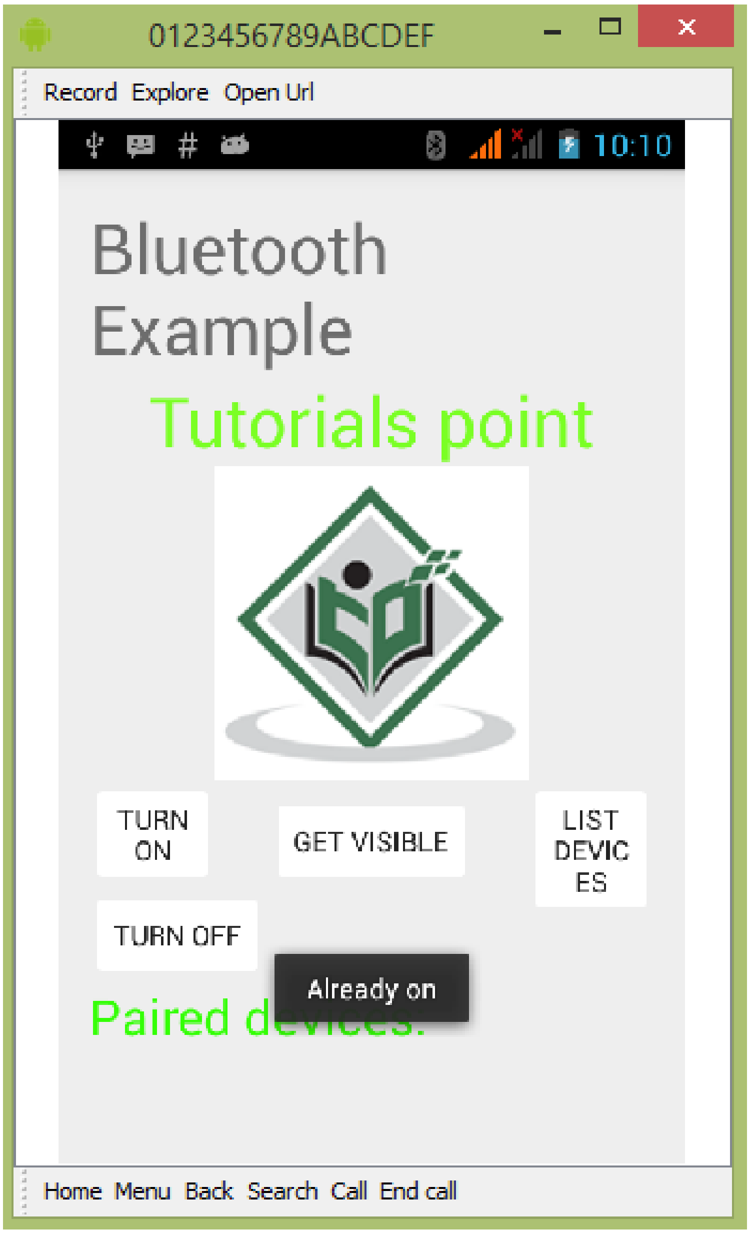
Device	Serial Number	State	Com...
Emulator Nexus 5 API 21 x86 Android 5.0.2 (	emulator-5554	Online	Yes
Iris Iris405+ Android 4.2.2 (API 17)	0123456789ABCDEF	Online	Yes

○ Launch emulator

Android virtual device: Nexus 5 API 21 x86

□ Use same device for future launches

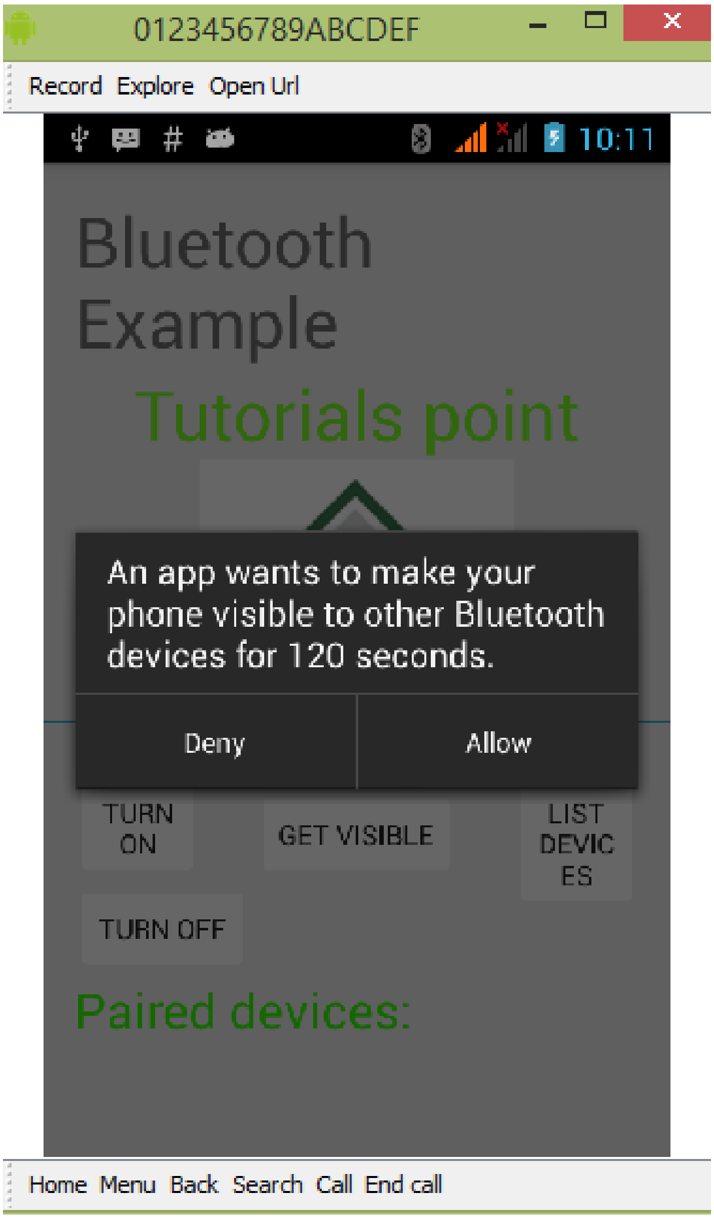
Now select Turn On to turn on the bluetooth. But as you select it , your Bluetooth will not be turned on. In fact , it will ask your permission to enable the Bluetooth.



Bluetooth Example

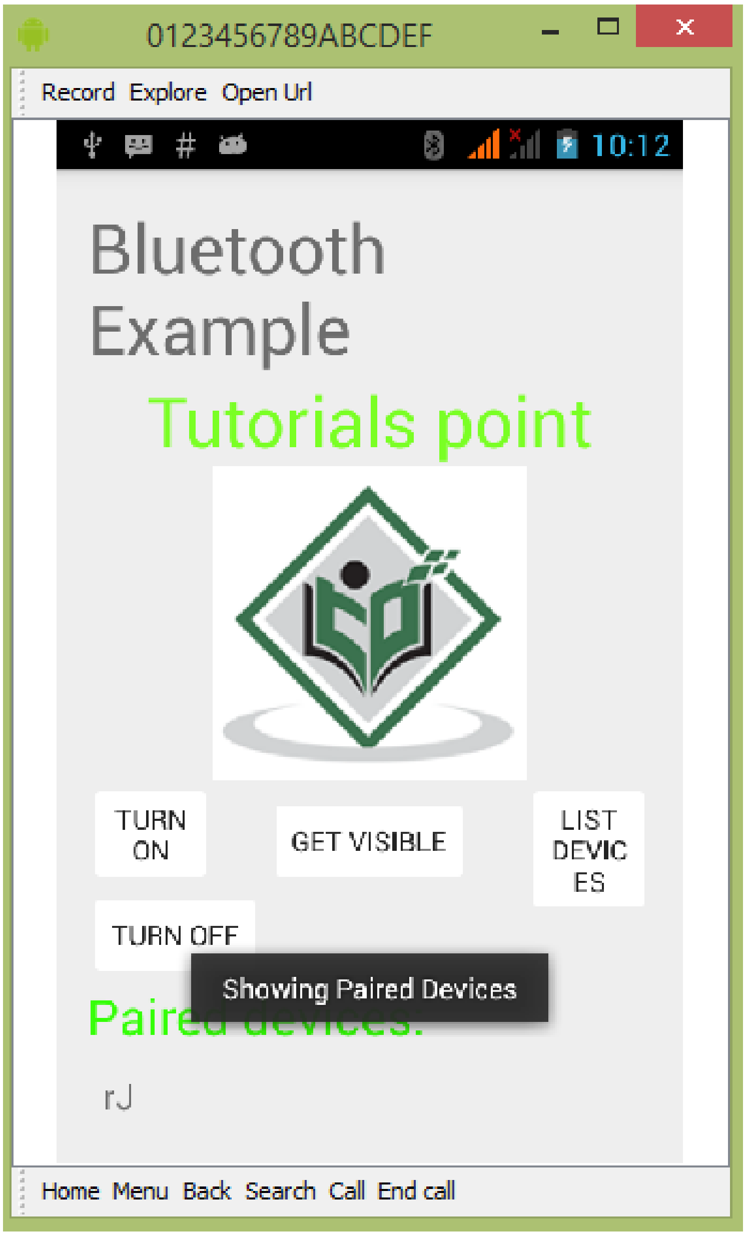
Paired devices:

Now just select the Get Visible button to turn on your visibility. The following screen would appear asking your permission to turn on discovery for 120 seconds.



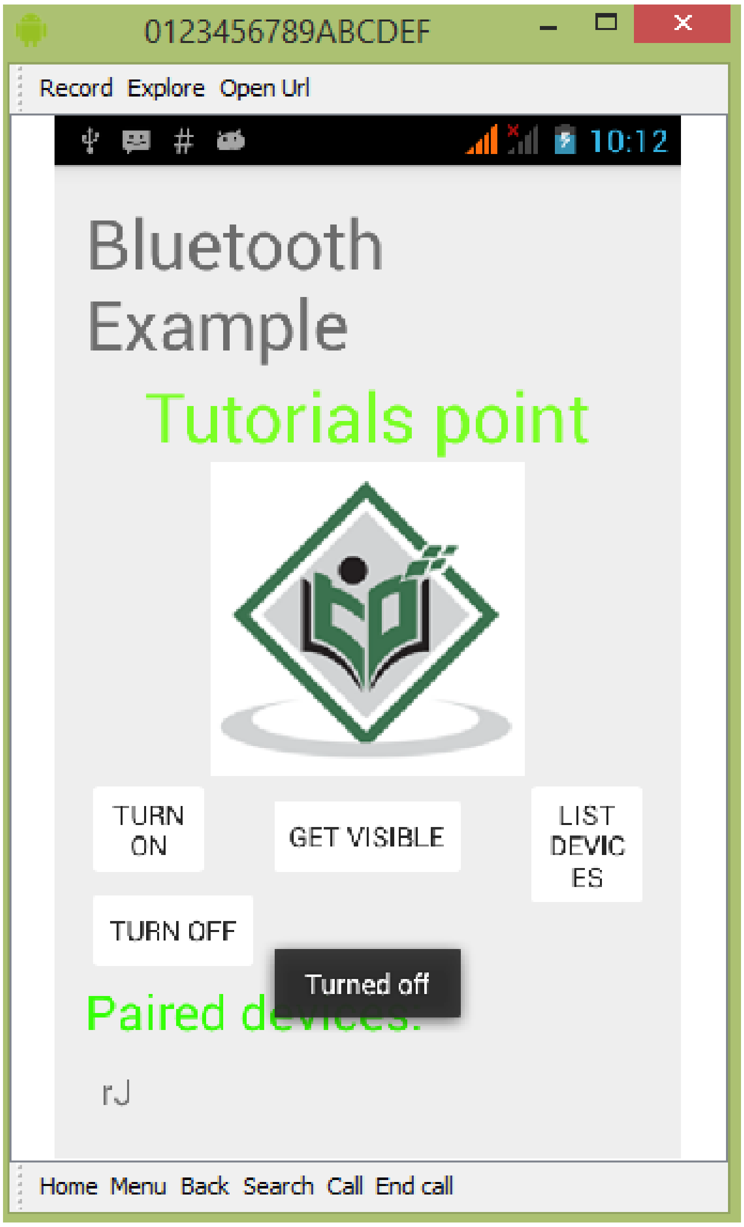
An app wants to make your phone visible to other Bluetooth devices for 120 seconds.

Now just select the List Devices option. It will list down the paired devices in the list view. In my case , I have only one paired device. It is shown below.



Showing Paired Devices

Now just select the Turn off button to switch off the Bluetooth. Following message would appear when you switch off the bluetooth indicating the successful switching off of Bluetooth.



Bluetooth Example

Tutorials point

Turned off

Paired devices:

rJ

Previous Page

Next Page

Write for us FAQ's Helping Contact

Enter email for newsletter go