Bluetooth

Bluetooth overview

The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices.
The app framework provides access to the Bluetooth functionality through Bluetooth APIs.
These APIs let apps connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

Using the Bluetooth APIs, an app can perform the following:

- Scan for other Bluetooth devices.
- Query the local **Bluetooth adapter** for paired Bluetooth devices.
- Establish RFCOMM channels.
- Connect to other devices through service discovery.
- Transfer data to and from other devices.
- Manage multiple connections.

Bluetooth APIs to accomplish the four major tasks necessary to communicate using Bluetooth:

- Setting up Bluetooth.
- Finding devices that are either paired or available in the local area.
- Connecting devices.
- Transferring data between devices.

Key classes and interfaces

All of the Bluetooth APIs are available in the android.bluetooth package.

BluetoothAdapter

Represents the local Bluetooth adapter (Bluetooth radio).

The BluetoothAdapter is the **entry-point for all Bluetooth interaction**.

Using this, you can discover other Bluetooth devices, query a list of bonded (paired) devices, instantiate a BluetoothDevice using a known MAC address, and **create a BluetoothServerSocket to listen for communications from other devices.**

BluetoothDevice

Represents a remote Bluetooth device.

Use this to request a connection with a remote device through a BluetoothSocket or query information about the device such as its name, address, class, and bonding state.

BluetoothClass

Describes the general characteristics and capabilities of a Bluetooth device. This is a read-only set of properties that defines the device's classes and services.

AndroidManifest.xml

```
<manifest ... >
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
...
</manifest>
```

Android BluetoothAdapter Class

- By using **BluetoothAdapter** object, we can interact with device's Bluetooth adapter to perform Bluetooth related operations. In case, if device does not contain any Bluetooth adapter, then it will return null.

```
BluetoothAdapter bAdapter = BluetoothAdapter.getDefaultAdapter();

if(bAdapter==null)

{ // Device won't support Bluetooth          }
```

Android Enable or Disable Bluetooth

- If Bluetooth is supported but disabled, then the **isEnabled()** method will return **false** and we can request the user to enable Bluetooth without leaving our application by using **startActivityForResult()** method with **ACTION_REQUEST_ENABLE** intent action parameter.
- To enable a Bluetooth by using **BluetoothAdapter** parameter **ACTION_REQUEST_ENABLE**.

```
if(!bAdapter.isEnabled())
{
    Intent eintent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(eintent, intVal);
}
```

- **Create an android application that enables and disables Bluetooth of the phone.**

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">

    <!--    button to turn bluetooth on or off    -->
    <Button
        android:id="@+id/btn_en_dis_bluetooth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enable Bluetooth" />
</LinearLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Bluetooth">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

MainActivity.java

```java
package com.example.bluetooth;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button btnEnableDisable;
    BluetoothAdapter bluetooth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnEnableDisable = findViewById(R.id.btn_en_dis_bluetooth);
        // fetching bluetooth adapter
        bluetooth = BluetoothAdapter.getDefaultAdapter();

        // if bluetooth not supported in device
        if(bluetooth == null) {
            Toast.makeText(this, "Bluetooth not Supported!", Toast.LENGTH_SHORT).show();
            btnEnableDisable.setEnabled(false);
            return;
        }
```

```
// if bluetooth supported in device
        // set button label according to bluetooth status on application start
        setButtonLabel();

        btnEnableDisable.setOnClickListener(view -> {
            // if bluetooth is enable then disable it
            if(bluetooth.isEnabled()) {
                bluetooth.disable();
                Toast.makeText(this, "Bluetooth Turned Off...",
Toast.LENGTH_SHORT).show();
            }
            // if bluetooth is disable then enable it
            else {
                bluetooth.enable();
                Toast.makeText(this, "Bluetooth Turned On...",
Toast.LENGTH_SHORT).show();
            }
            // updating button label
            setButtonLabel();
        });
    }

    // method to change button label according to bluetooth status
    private void setButtonLabel() {
        if(bluetooth.isEnabled())
            btnEnableDisable.setText("Disable Bluetooth");
        else
            btnEnableDisable.setText("Enable Bluetooth");
    }
}
```