WiFi

Wi-Fi

- Android allows applications to access the state of the wireless connections.
- Application can access almost all the information of a wifi connection.
- The information that an application can access includes connected network's link speed, IP address, negotiation state, other networks information.
- Applications can also scan, add, save, terminate and initiate Wi-Fi connections.

Wi-Fi

Android provides WifiManager API to manage all aspects of WIFI connectivity.

We can instantiate this class by calling getSystemService method.

Its syntax is given below −

```
WifiManager mainWifiObj;
mainWifiObj = (WifiManager) getSystemService(Context.WIFI_SERVICE);
//
WifiManager mainWifiObj;
mainWifiObj = (WifiManager)
getApplicationContext().getSystemService(Context.WIFI_SERVICE);
```

Wi-Fi

In order to scan a list of wireless networks, you also need to register your BroadcastReceiver. It can be registered using registerReceiver method with argument of your receiver class object.

Its syntax is given below –

```
class WifiScanReceiver extends BroadcastReceiver
{   @override
   public void onReceive(Context c, Intent intent) {
   } }
WifiScanReceiver wifiReciever = new WifiScanReceiver();
registerReceiver(wifiReciever, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION))
;
```

Wi-Fi

The wifi scan can be start by calling the startScan method of the WifiManager class.
This method returns a list of ScanResult objects.
You can access any object by calling the get method of list.
Its syntax is given below −

```
List<ScanResult> wifiScanList =
mainWifiObj.getScanResults();

  String data = wifiScanList.get(0).toString();
```

Wi-Fi

Methods And Description

getWifiState() : This method gets the Wi-Fi enabled state

isWifiEnabled() : This method return whether Wi-Fi is enabled or disabled.

setWifiEnabled(boolean enabled) : This method enable or disable Wi-Fi.

updateNetwork(WifiConfiguration config) : This method update the network description of an existing configured network.

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

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.wifi">
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-
permission>
 <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-
permission>
    <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.WiFi">
        <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>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    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 wifi on or off    -->
    <Button
        android:id="@+id/btn_en_dis_wifi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Toggle Wifi State" />

</LinearLayout>
```

```java
package com.example.wifi;

import androidx.appcompat.app.AppCompatActivity;
import android.net.wifi.WifiManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button btnEnableDisableWifi;
    WifiManager wifiManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnEnableDisableWifi = findViewById(R.id.btn_en_dis_wifi);
        // getting wifi service
        wifiManager = (WifiManager) getApplicationContext().getSystemService(WIFI_SERVICE);
        btnEnableDisableWifi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (wifiManager.isWifiEnabled()) {
                    wifiManager.setWifiEnabled(false);
                    btnEnableDisableWifi.setText("Enable Wifi");
                    Toast.makeText(MainActivity.this, "Wifi Disabled", Toast.LENGTH_SHORT).show();
                }
                // if wifi is disable make it enable
               else{
                wifiManager.setWifiEnabled(true);
                btnEnableDisableWifi.setText("Disable Wifi");
                Toast.makeText(MainActivity.this, "Wifi Enabled", Toast.LENGTH_SHORT).show();
               }
            }});
    }
}
```

Self study

- Difference between geocoding and reverse geocoding