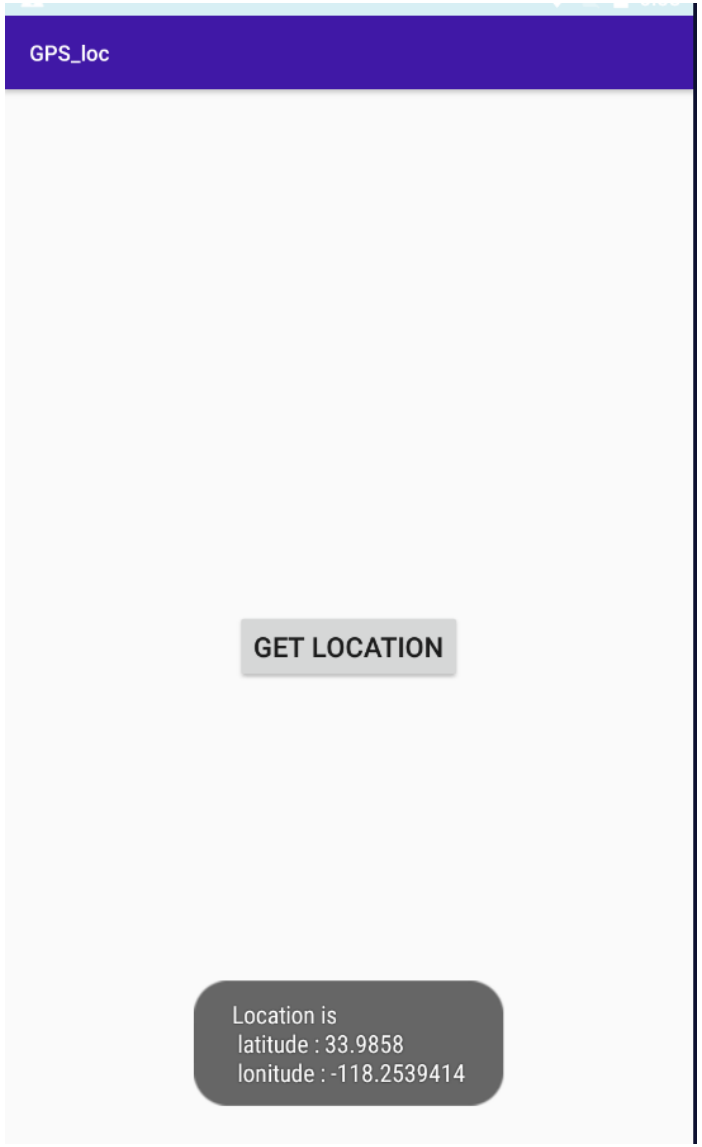
Global Positioning System (GPS)


GPS_loc
GET LOCATION
Location is
latitude : 33.9858
lonitude : -118.2539414

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="GPS_loc"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GPS_loc"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:theme="@style/FullscreenTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

GpsTracker.java

```java
package com.example.gps_loc;

import android.Manifest;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
```

GpsTracker.java

```java
public class GpsTracker extends Service implements LocationListener {

    5 usages
    private Context context;
    11 usages
    private LocationManager locationManager;
    16 usages
    private Location location;
    4 usages
    private double latitude = 0.0;
    4 usages
    private double longitude = 0.0;
    3 usages
    private boolean isGpsEnabled = false;
    3 usages
    private boolean isNetworkEnabled = false;
    3 usages
    private boolean canGetLocation = false;

    2 usages
    public static final int TIME_BW_UPDATES = 1000 * 10; // 10 seconds
    2 usages
    public static final int MIN_DISTANCE_FOR_UPDATE = 10;     // 10 meter


    //Constructor
    1 usage
    public GpsTracker(Context context) {
        this.context = context;
        getLocation();// Define below
    }//GpsTracker
```

GpsTracker.java

```
42      public Location getLocation() throws SecurityException{
43
44          locationManager = (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
45          isGpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
46          isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
47
48          if (!isGpsEnabled && !isNetworkEnabled) {
49              canGetLocation = false;
50          } else //isGpsEnabled || isNetworkEnabled
51          {
52              canGetLocation = true;
53              if(!hasPermissions()){
54                  location = null;
55                  return location;
56              }
57              if (isNetworkEnabled) {
58                  locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,TIME_BW_UPDATES, MIN_DISTANCE_FOR_UPDATE,  listener: this);
59                  if (locationManager != null) {
60                      location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
61                      if (location != null) {
62                          latitude = location.getLatitude();
63                          longitude = location.getLongitude();
64                      }
65                  }
66              }//isNetworkEnabled
```

GpsTracker.java

```
68              if (isGpsEnabled && location == null) {
69                  locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,TIME_BW_UPDATES, MIN_DISTANCE_FOR_UPDATE, listener: this);
70                  if (locationManager != null) {
71                      location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
72                      if (location != null) {
73                          latitude = location.getLatitude();
74                          longitude = location.getLongitude();
75                      }
76                  }
77              }
78          }//isGpsEnabled || isNetworkEnabled
79
80          return location;
81      }//getLocation
82
        1 usage
83      public double getLatitude() {
84          if (location != null) {
85              latitude = location.getLatitude();
86          }
87          return latitude;
88      }//getLatitude
89
        1 usage
90      public double getLongitude() {
91          if (location != null) {
92              longitude = location.getLongitude();
93          }
94          return longitude;
95      }//getLongitude
```

GpsTracker.java

```
no usages
97    public boolean canGetLocation() { return this.canGetLocation; }//canGetLocation
100

      2 usages
101   public boolean hasPermissions(){
102       return (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED);
103   }//hasPermissions
104

      1 usage
105   public void showGpsAlertDialog() {
106       AlertDialog.Builder builder = new AlertDialog.Builder(context);
107       builder.setTitle("GPS")
108               .setMessage("GPS is not enabled. Do you want to go to Settings menu?")
109               .setPositiveButton( text: "Settings", new DialogInterface.OnClickListener() {
110                   @Override
111                   public void onClick(DialogInterface dialog, int i) {
112                       Intent intent = new Intent();
113                       intent.setAction(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
114                       context.startActivity(intent);
115                   }//onClick+
116               })//setPositiveButton
117               .setNegativeButton( text: "Cancel", new DialogInterface.OnClickListener() {
118                   @Override
119                   public void onClick(DialogInterface dialog, int i) {
120                       dialog.cancel();
121                   }//onClick-
122               });//setNegativeButton
123       builder.show();
124   }//showGpsAlertDialog
```

GpsTracker.java

```
126     public void stopUsingGps() throws SecurityException{
127         if (locationManager != null && hasPermissions()) {
128             locationManager.removeUpdates( listener: this);
129         }
130     }//stopUsingGps
131
        6 usages
132     @Override
133     public void onLocationChanged(Location location) {
134         if(location != null){
135             Log.i(GpsTracker.class.getSimpleName(),
136                     msg: "lat : " + location.getLatitude() + ", lon : " + location.getLongitude());
137         } else {
138             Log.i(GpsTracker.class.getSimpleName(), msg: "location = null");
139         }
140     }
141
142     @Override
143     public void onStatusChanged(String s, int i, Bundle bundle) {
144
145     }
146
        1 usage
147     @Override
148     public void onProviderEnabled(String provider) {
149         Log.i(GpsTracker.class.getSimpleName(), msg: "provider enabled : " + provider);
150     }
```

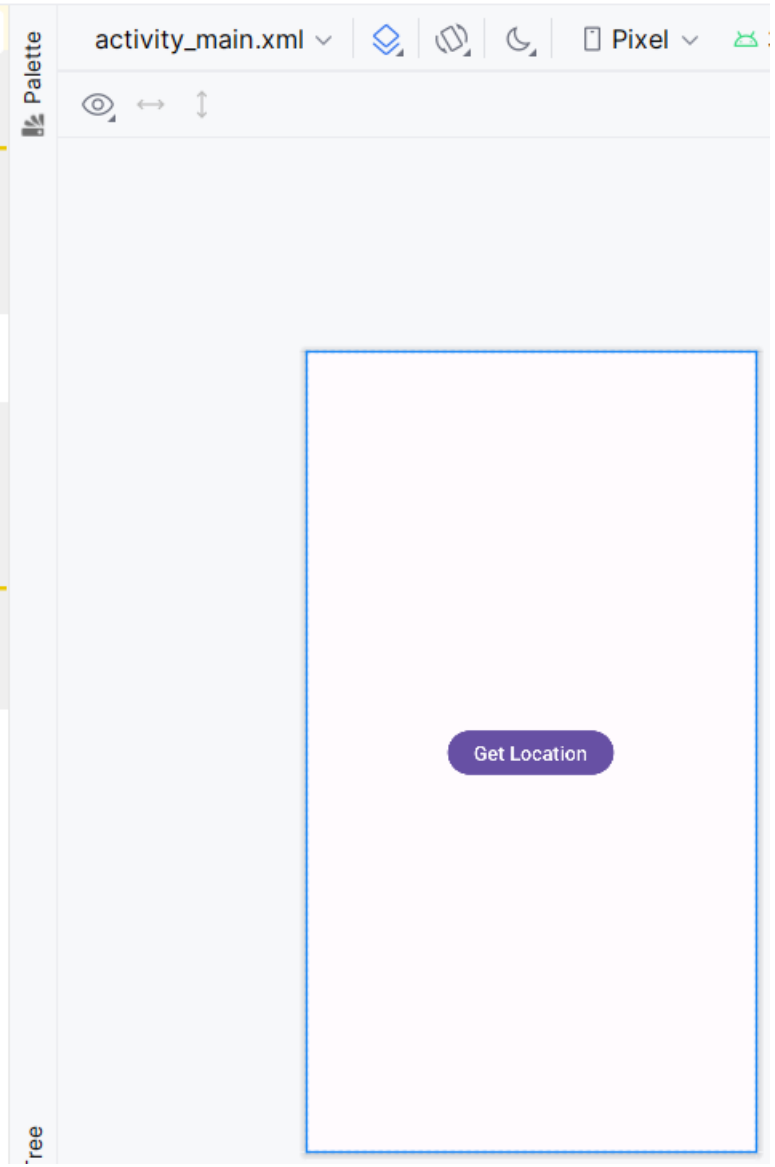
```
152         @Override
153         public void onProviderDisabled(String provider) {
154             Log.i(GpsTracker.class.getSimpleName(), msg: "provider disabled : " + provider);
155
156         }
157
158         @Nullable
159         @Override
160         public IBinder onBind(Intent intent) { return null; }
163     }//GpsTracker
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Get Location"
        android:textSize="18sp"
        android:layout_gravity="center"
        android:onClick="onclick"
        />


</FrameLayout>
```



MainActivity.java

```java
package com.example.gps_loc;

import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.Manifest;
import android.content.pm.PackageManager;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import android.view.View;
import android.widget.Toast;


public class MainActivity extends AppCompatActivity {

    2 usages
    private static final int PERMISSION_REQ_CODE = 1234;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        checkPermissions();
    }//onCreate

    @Override
    protected void onResume() { super.onResume(); }//onResume
```

MainActivity.java

```
1 usage
29  private void checkPermissions() {
30      if(ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
31              != PackageManager.PERMISSION_GRANTED){
32
33          ActivityCompat.requestPermissions( activity: this,
34                  new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
35                  PERMISSION_REQ_CODE);
36      }
37
38  }//checkPermissions
39
40  @Override
41  public void onRequestPermissionsResult(int requestCode,
42                                         @NonNull String[] permissions,
43                                         @NonNull int[] grantResults) {
44      super.onRequestPermissionsResult(requestCode, permissions, grantResults);
45      if (requestCode == PERMISSION_REQ_CODE) {
46          if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
47              Toast.makeText( context: this,  text: "Permission granted", Toast.LENGTH_SHORT).show();
48          } else {
49              Toast.makeText( context: this,  text: "Permission denied to access location", Toast.LENGTH_SHORT).show();
50          }
51      }
52
53  }//onRequestPermissionsResult
```

```
55          public void onclick(View v){
56              GpsTracker gpstracker =  new GpsTracker( context: this);
57              if(gpstracker.canGetLocation()){
58                  double lat = gpstracker.getLatitude();
59                  double lon = gpstracker.getLongitude();
60                  Toast.makeText( context: this,
61                          text: "Location is \n latitude : " + lat + "\n lonitude : " + lon, Toast.LENGTH_SHORT).show();
62              } else {
63                  gpstracker.showGpsAlertDialog();
64              }
65          }//onclick
66
67      }//MainActivity
```