

# Lab bài 8

## 1. MỤC TIÊU

Sau khi hoàn thành bài thực hành, bạn có thể nắm được các vấn đề sau:

- Biết sử dụng Location API để lấy vị trí hiện tại
- Biết cách triển khai Location Based Service trong ứng dụng Android

## 2. THỜI GIAN THỰC HIỆN: 20'

## 3. YÊU CẦU

Viết một ứng dụng Android đơn giản và sử dụng Location API để lấy và hiển thị vị trí hiện tại của thiết bị.

## 4. CÁC BƯỚC THỰC HIỆN

### Bước 1

Tạo project với các thông tin sau:

- Project Name: lab8
- Package Name: location.tenv.lab

### Bước 2

Tạo một service có tên GPSTrackerService để thực hiện logic lấy vị trí hiện tại của thiết bị.

```

package funix.prm.lab;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
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;

public class GPSTrackerService extends Service implements LocationListener {
    private final Context mContext;
    private boolean isGPSEnabled = false; // flag for GPS status
    private boolean isNetworkEnabled = false; // flag for network status
    private boolean canGetLocation = false; // flag for GPS status
    private Location location; // location
    private double latitude; // latitude
    private double longitude; // longitude
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10 meters
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute
    protected LocationManager locationManager; // Declaring a Location Manager

    public GPSTrackerService(Context context) {
        this.mContext = context;
        getLocation();
    }
}

```

```

@SuppressLint("MissingPermission")
public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);
        // getting GPS status
        isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        // getting network status
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {
            this.canGetLocation = true;
            // First get location from Network Provider
            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }
    }
}

```

```

        // if GPS Enabled get lat/long using GPS Services
        if (isGPSEnabled) {
            if (location == null) {
                locationManager.requestLocationUpdates(
                    locationManager.GPS_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, listener: this);
                Log.d( tag: "GPS Enabled", msg: "GPS Enabled");
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return location;
}

```

```

public void stopUsingGPS() {
    if (locationManager != null) {
        locationManager.removeUpdates( listener: GPSTrackerService.this);
    }
}

public double getLatitude() {
    if (location != null) {
        latitude = location.getLatitude();
    }

    return latitude;
}

public double getLongitude() {
    if (location != null) {
        longitude = location.getLongitude();
    }
    return longitude;
}

public boolean canGetLocation() {
    return this.canGetLocation;
}

```

```

public void showSettingsAlert() {
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);
    // Setting Dialog Title
    alertDialog.setTitle("GPS is settings");
    // Setting Dialog Message
    alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");
    // On pressing Settings button
    alertDialog.setPositiveButton( text: "Settings", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            mContext.startActivity(intent);
        }
    });
    // on pressing cancel button
    alertDialog.setNegativeButton( text: "Cancel", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    // Showing Alert Message
    alertDialog.show();
}

```

```

@Override
public void onLocationChanged(Location location) {
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}

@Override
public IBinder onBind(Intent arg0) {
    return null;
}
}

```

### Bước 3

Tạo activity có tên MainActivity để xử lý thao tác người dùng khi lấy thông tin vị trí

```

package funix.prm.lab;

import android.Manifest;
import android.app.Activity;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.test.mock.MockPackageManager;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {
    private Button mBtnShowLocation;
    private static final int REQUEST_CODE_PERMISSION = 2;
    private String mPermission = Manifest.permission.ACCESS_FINE_LOCATION;
    // GPSTrackerService class
    private GPSTrackerService mGPSTracker;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        try {
            if (ActivityCompat.checkSelfPermission(context: this, mPermission)
                != MockPackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(activity: this, new String[]{mPermission},
                    REQUEST_CODE_PERMISSION);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        mBtnShowLocation = (Button) findViewById(R.id.btnGetLocation);
    }
}

```

```

        mBtnShowLocation.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                // create class object
                mGPSTracker = new GPSTrackerService(context: MainActivity.this);

                // check if GPS enabled
                if (mGPSTracker.canGetLocation()) {

                    double latitude = mGPSTracker.getLatitude();
                    double longitude = mGPSTracker.getLongitude();

                    // \n is for new line
                    Toast.makeText(getApplicationContext(), text: "Your Location is - \nLat: "
                        + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
                } else {
                    // can't get location
                    // GPS or Network is not enabled
                    // Ask user to enable GPS/network in settings
                    mGPSTracker.showSettingsAlert();
                }
            }
        });
    }
}

```

## Bước 4

Tạo file layout có tên activity\_main để làm giao diện cho màn hình MainActivity

```

<?xml version = "1.0" encoding = "utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/btnGetLocation"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="GET LOCATION" />

</LinearLayout>

```

## Bước 5

Cấu hình activity và khai báo các permissions cần thiết trên AndroidManifest.xml

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="funix.prm.lab">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="PRM391xLabs"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

## 5. KIỂM TRA KẾT QUẢ

Chạy ứng dụng trên AVD hoặc bất cứ thiết bị Android nào mà bạn có. Màn hình sau xuất hiện, yêu cầu thiết bị cho phép ứng dụng truy cập vào location của thiết bị.



100% 16:47

GET LOCATION



Allow **PRM391xLabs**  
to access this device's  
location?


DENY

ALLOW

Bạn chọn Allow, màn hình sau xuất hiện



GET LOCATION



Bạn chọn vào nút "GET LOCATION", màn hình sau xuất hiện và tắt hiển thị thông tin về vị trí của thiết bị như sau



100% 16:47

GET LOCATION



Your Location is -  
Lat: 21.0682098  
Long: 105.7625727

**KẾT THÚC**