

## LAB10:

### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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"/>
    <uses-feature android:name="android.hardware.location.gps"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.Lab10"
        tools:targetApi="31">

        <!--
        TODO: Before you run your application, you need a Google Maps
API key.

        To get one, follow the directions here:

            https://developers.google.com/maps/documentation/android-
sdk/get-api-key

        Once you have your API key (it starts with "AIza"), define a
new property in your
        project's local.properties file (e.g. MAPS_API_KEY=Aiza...),
and replace the
        "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyCBlap-jqb0uC3vp7eBrzJn8iiTKJpxtgM" />

        <activity
            android:name=".MapsActivity"
            android:exported="true"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>

            </intent-filter>
        </activity>
    </application>
```

```
</manifest>
```

## Mainactivity.java

```
package com.example.lab10;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;

import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.Manifest;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.example.lab10.databinding.ActivityMapsBinding;

public class MapsActivity extends FragmentActivity implements
    OnMapReadyCallback {

    private GoogleMap mMap;
    private ActivityMapsBinding binding;
    private LocationListener locationListener;
    private LocationManager locationManager;
    private final long MIN_DIST=5;
    private final long MIN_TIME=1000;
    private LatLng latLng;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMapsBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        // Obtain the SupportMapFragment and get notified when the map is
        // ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            PackageManager.PERMISSION_GRANTED);
    }
}
```

```

        ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.ACCESS_COARSE_LOCATION},
PackageManager.PERMISSION_GRANTED);
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or
move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the
user will be prompted to install
     * it inside the SupportMapFragment. This method will only be
triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady (GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(-34, 151);
        mMap.addMarker(new
MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        locationListener = new LocationListener() {
            @Override
            public void onLocationChanged(@NonNull Location location)
            {
                LatLng = new
                LatLng(location.getLatitude(),location.getLongitude());
                mMap.addMarker(new MarkerOptions().position(LatLng
                ).title("My position"));

                mMap.moveCamera(CameraUpdateFactory.newLatLng(LatLng));

            }
        };
        locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
        try{

            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TI
ME,MIN_DIST,locationListener);}
        catch (SecurityException e){
            e.printStackTrace();
        }

    }

}

```

## activity\_maps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```