Mobile Applications for Business

Master SIA/SDBIS

Octavian Dospinescu

2021

Maps in Android

Steps and methodology

Preparing the environment to support applications with **maps**

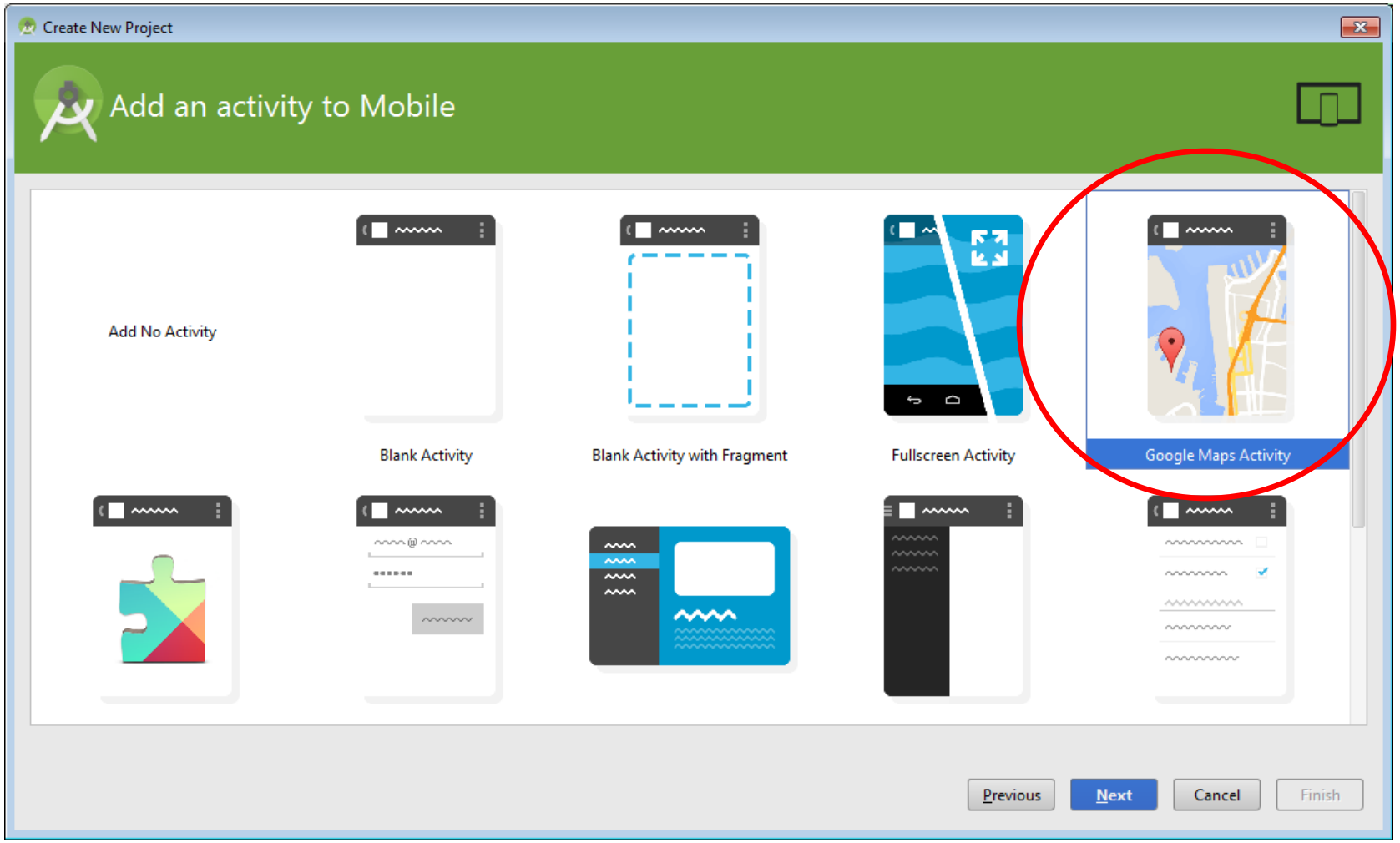
1. Create a new Android project in AndroidStudio
2. Install Google Play Services in Android SDK
3. Set the permissions in AndroidManifest.xml
4. Obtain GoogleMaps API V2 KEY (the key)
5. Include the key in the project (google_maps_api_key.xml)
6. Design the layout and implement the activity
7. Run! ☺

(more details: https://developers.google.com/maps/documentation/android-api/start)

Some very useful resources

- https://developers.google.com/maps/gmp-get-started#create-project
- https://developers.google.com/maps/gmp-get-started#enable-api-sdk
- https://developers.google.com/maps/gmp-get-started#api-key

1. Create a new Android project

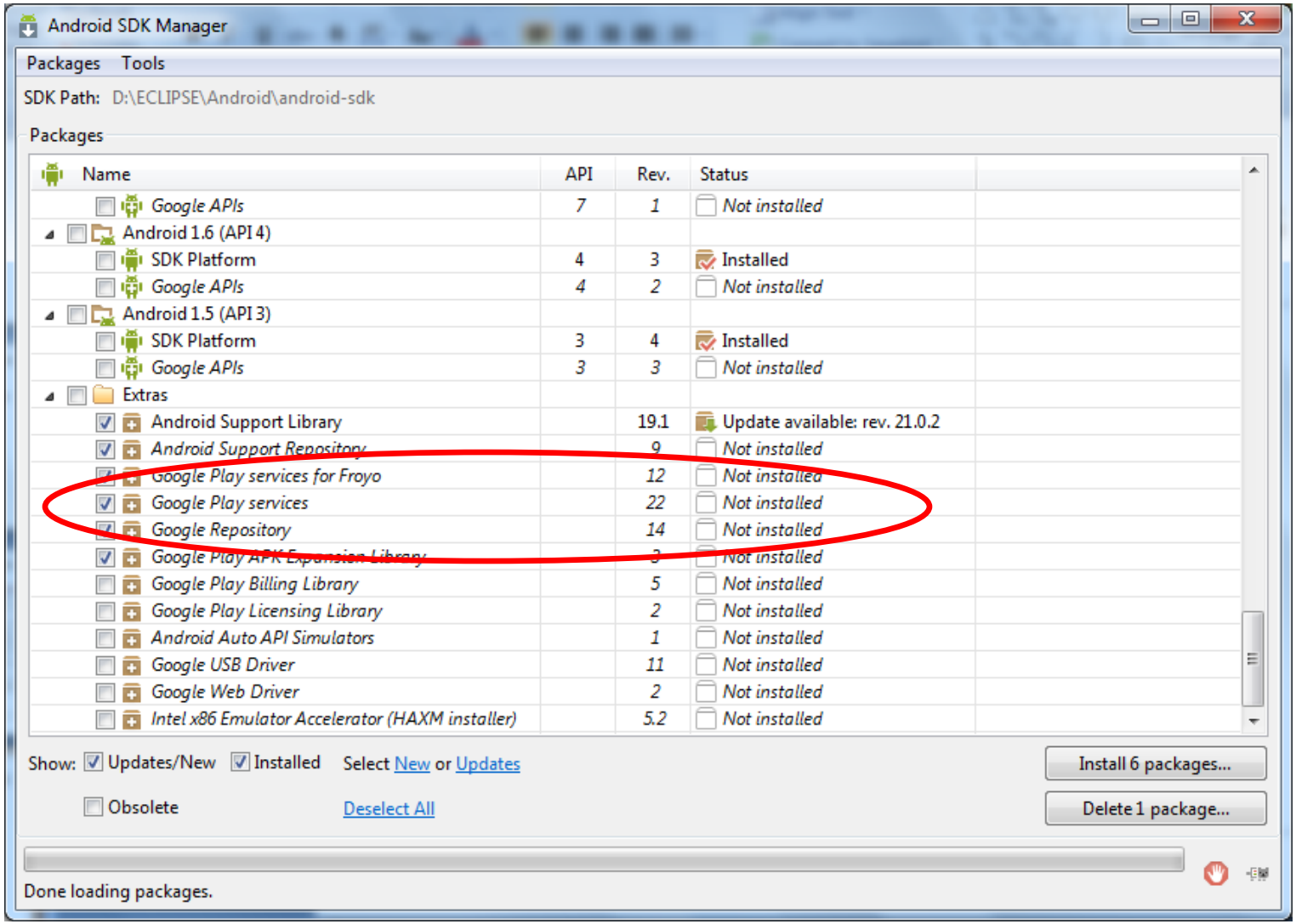


2. Install Google Play Services and Google Repository from Android SDK Manager

- Update/Install Google Play Services using Android SDK Manager
  - Android Studio: Tools/Android/SDK Manager (SDK Tools tab)
  - Select from **Extras** the options **Google Play services** and **Google Repository** and install the packages

  (more details: http://developer.android.com/sdk/installing/adding-packages.html)

2. Install Google Play Services and Google Repository from Android SDK Manager



Install Google Play Services

- !!! **Attention**: it is recommended to close and restart Android Studio after the installation of Google Play Services

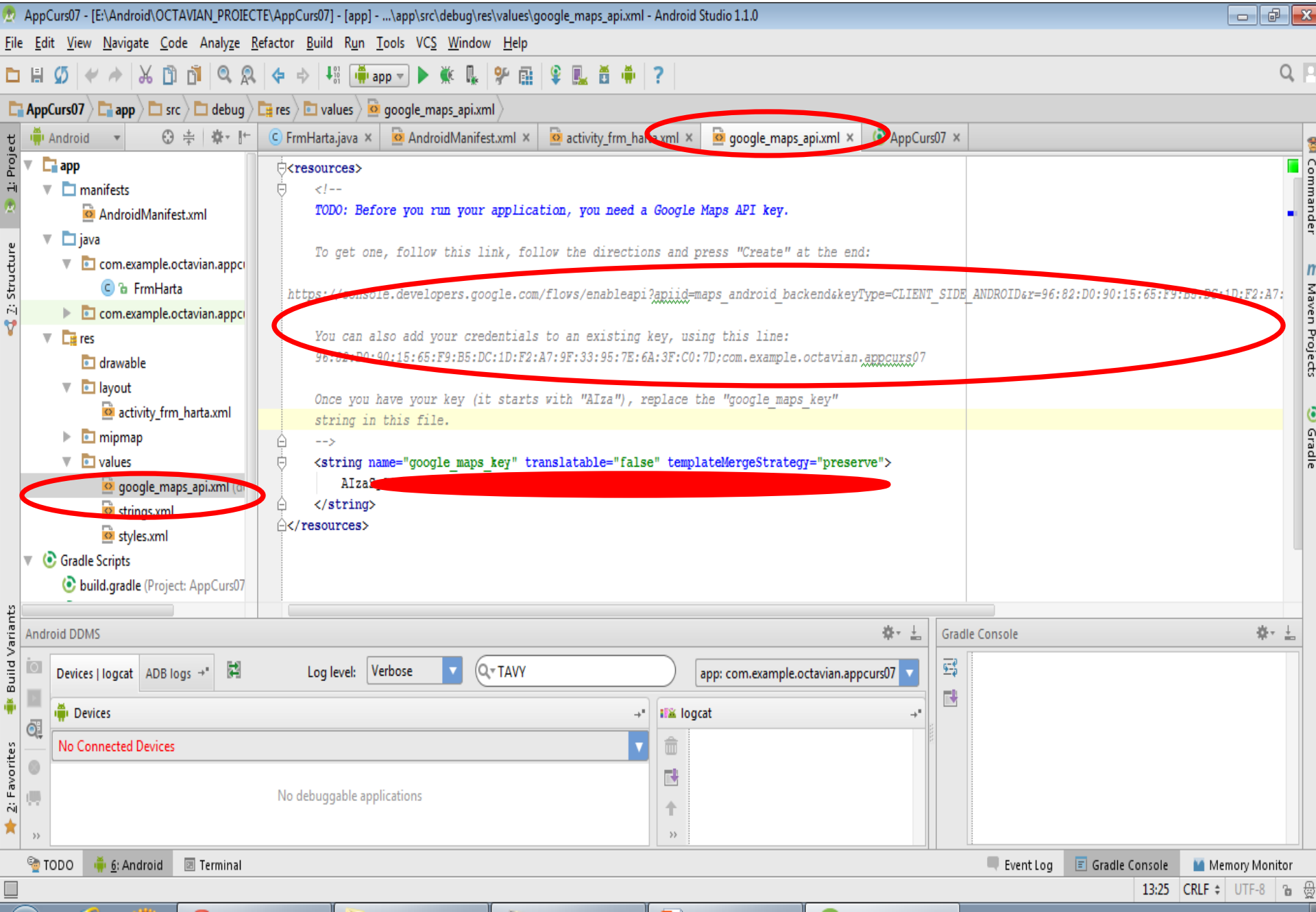
3. Set the permissions in AndroidManifest.xml

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

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

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

<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
```

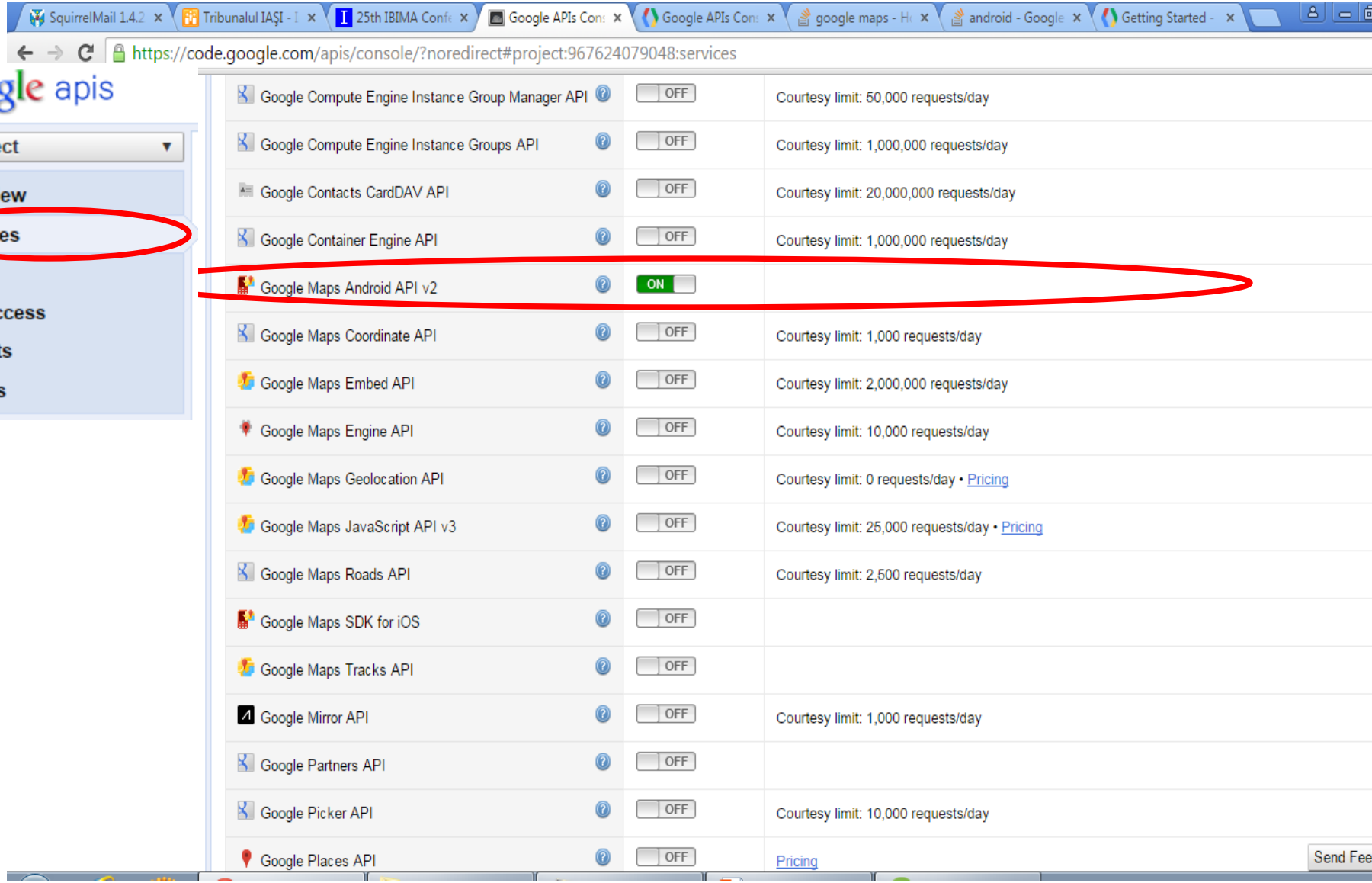
4. Obtain GoogleMaps API V2 KEY (the key)

- Read the file google_maps_api.xml and follow the instructions.
- You should have a Google Console Developer Account (https://code.google.com/apis/console/?noredirect).
- Activate from the left menu **Services** the option **Google Maps Android API v2 / Maps SDK for Android**

google_maps_api.xml



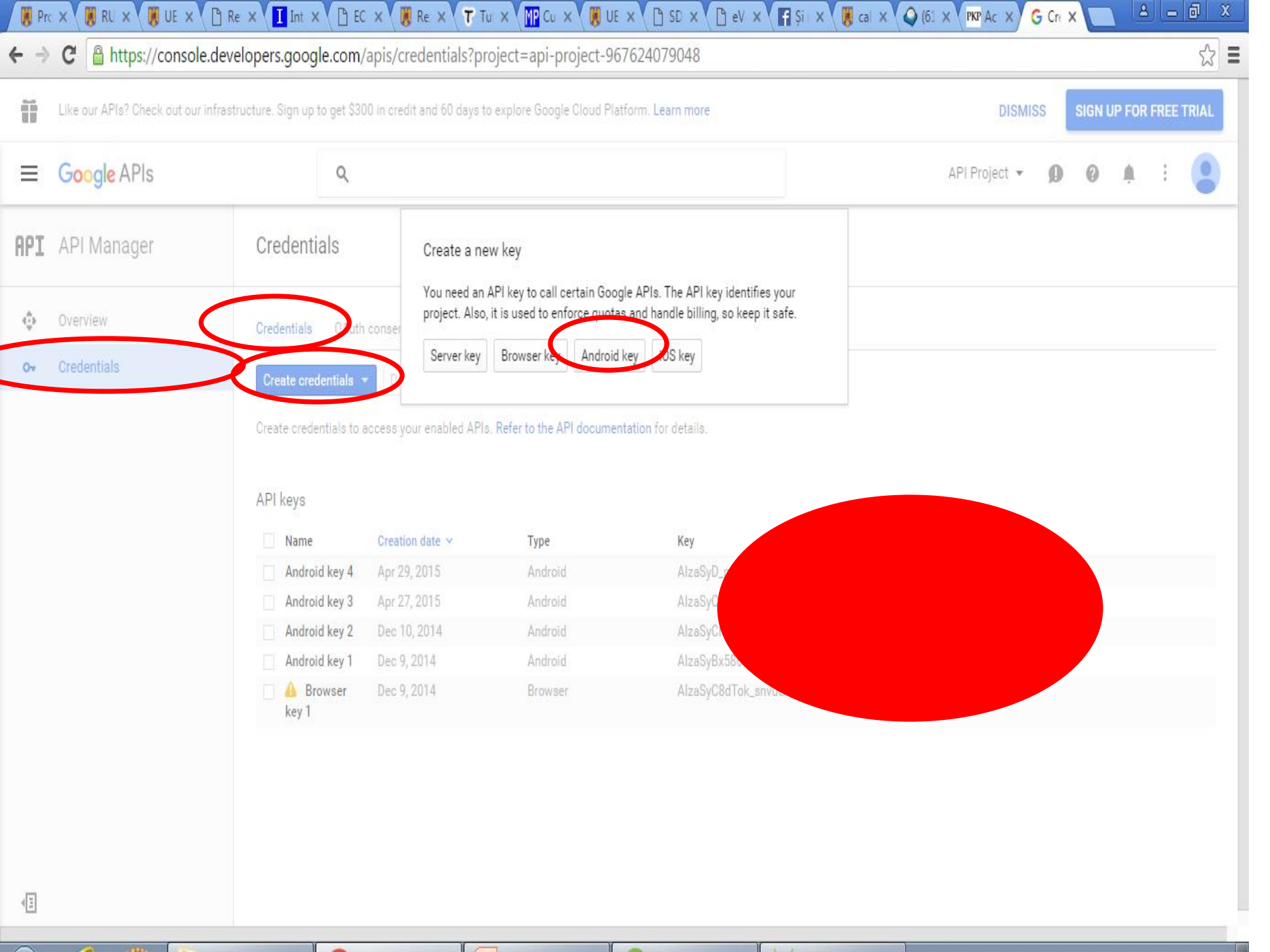
Activate the service Google Maps Android API v2

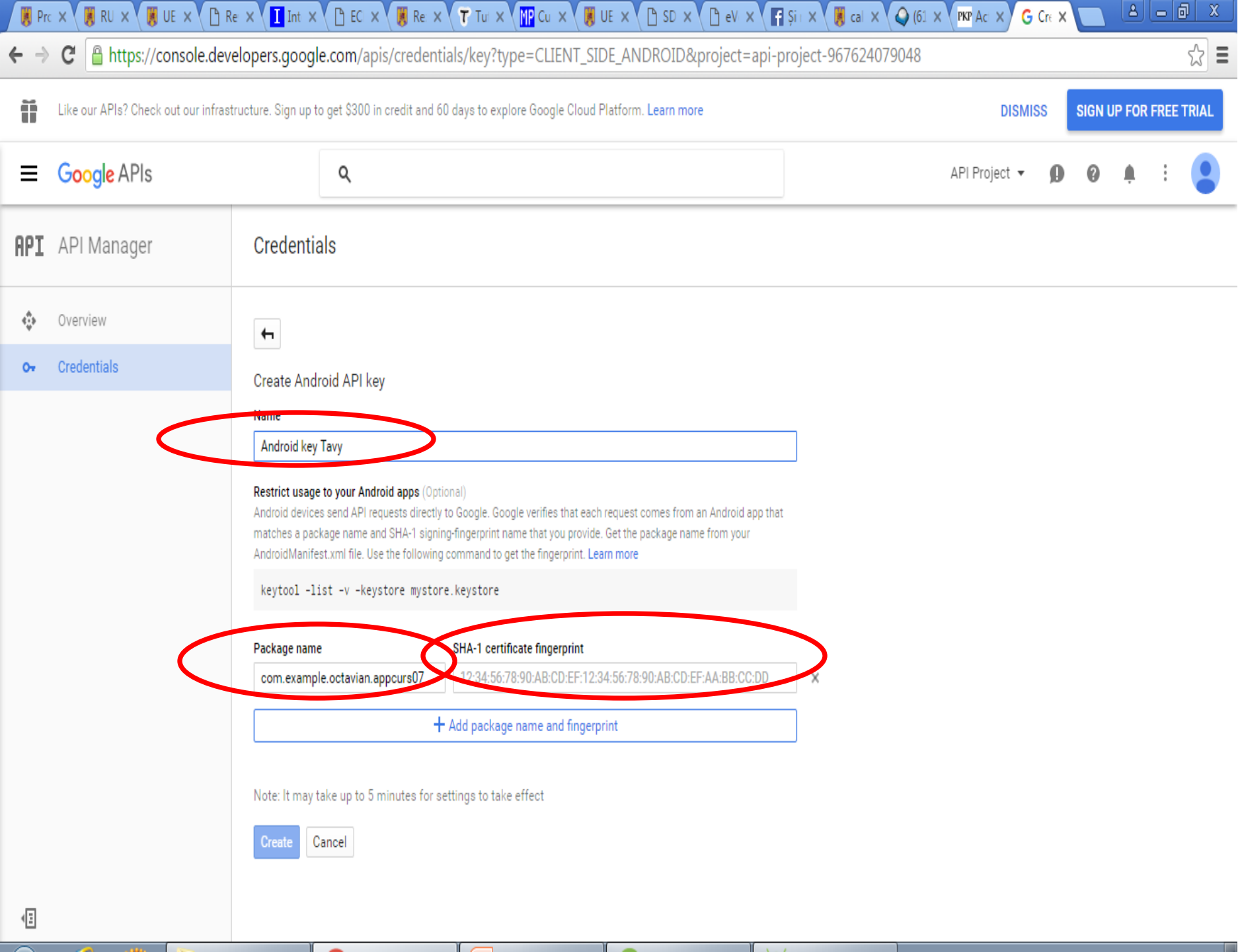


5. Obtain the key and include it in the project

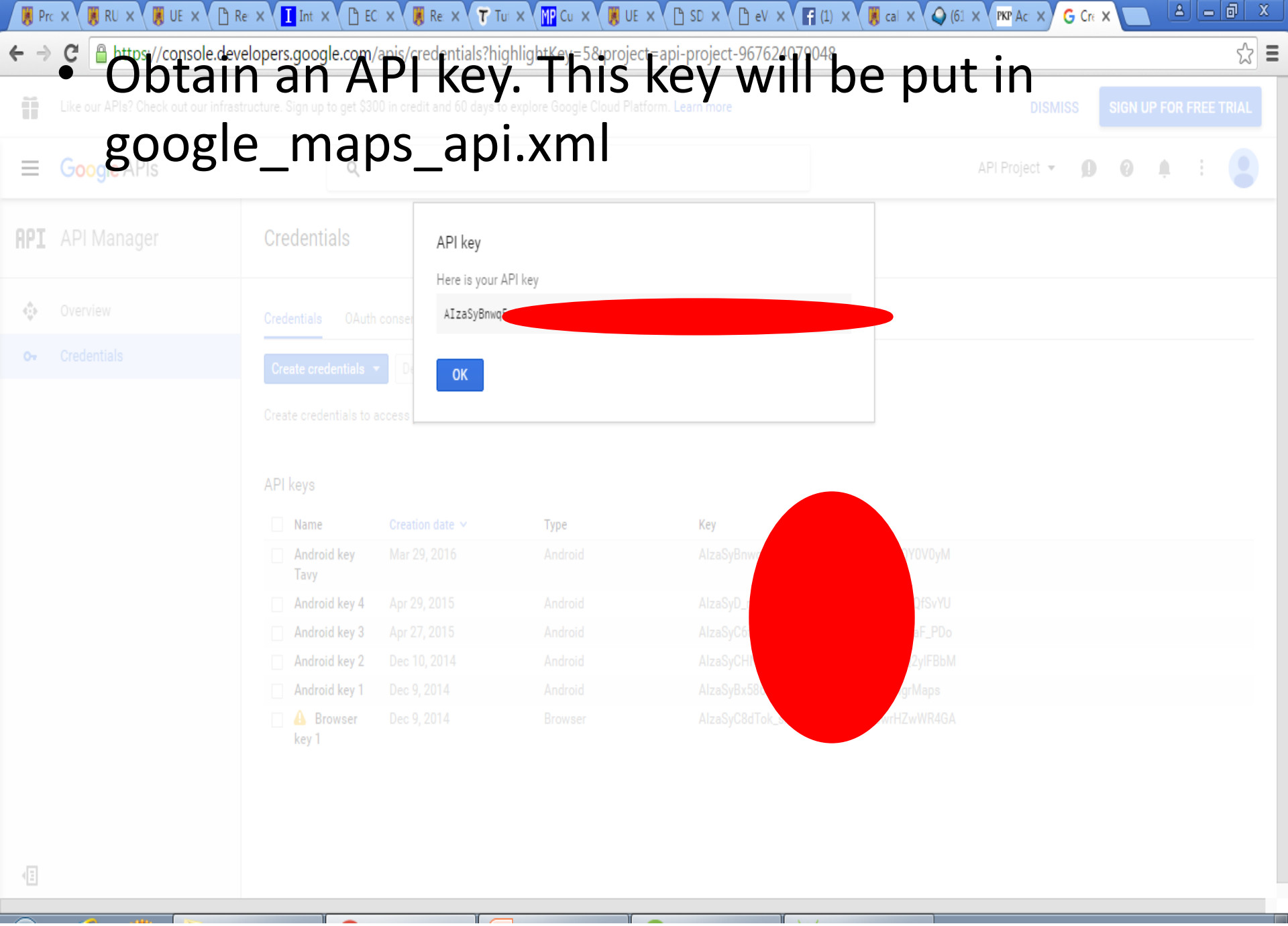
- In the left menu: **API Access/Credentials**
- From the menu: **Create credentials**
- Select **Android key**
- Insert the package name and the line from google_maps_api.xml (something like: 96:82:D0:90:15:F9:B5:DG:1D:F2:A7:8F:33:95:7E:6A:3F:C0:7D;com.example.octavian.appcurs07)
- Get the api key(something like AIzaSyC6tyaR9SjzPNFagMHVsf5dJVVaF_Pdo)
- Put the api key in google_maps_api.xml

(more details here: http://stackoverflow.com/questions/27609442/how-to-get-the-sha1-fingerprint-certificate-in-android-studio-for-debug-mode)


https://console.developers.google.com/apis/credentials?project=api-project-967624079048
Like our APIs? Check out our infrastructure. Sign up to get $300 in credit and 60 days to explore Google Cloud Platform. Learn more
DISMISS
SIGN UP FOR FREE TRIAL
Google APIs
API Project
API Manager
Credentials
Overview
Credentials
Credentials
Create credentials
Create a new key
You need an API key to call certain Google APIs. The API key identifies your project. Also, it is used to enforce quotas and handle billing, so keep it safe.
Server key
Browser key
Android key
iOS key
Create credentials to access your enabled APIs. Refer to the API documentation for details.
API keys
Name
Creation date
Type
Key
Android key 4
Apr 29, 2015
Android
Android key 3
Apr 27, 2015
Android
Android key 2
Dec 10, 2014
Android
Android key 1
Dec 9, 2014
Android
Browser key 1
Dec 9, 2014
Browser


https://console.developers.google.com/apis/credentials/key?type=CLIENT_SIDE_ANDROID&project=api-project-967624079048
Like our APIs? Check out our infrastructure. Sign up to get $300 in credit and 60 days to explore Google Cloud Platform. Learn more
DISMISS
SIGN UP FOR FREE TRIAL
Google APIs
API Project
API Manager
Credentials
Overview
Credentials
Create Android API key
Name
Android key Tavy
Restrict usage to your Android apps (Optional)
Android devices send API requests directly to Google. Google verifies that each request comes from an Android app that matches a package name and SHA-1 signing-fingerprint name that you provide. Get the package name from your AndroidManifest.xml file. Use the following command to get the fingerprint. Learn more
keytool -list -v -keystore mystore.keystore
Package name
SHA-1 certificate fingerprint
com.example.octavian.appcurs07
12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD
Add package name and fingerprint
Note: It may take up to 5 minutes for settings to take effect
Create
Cancel

- Obtain an API key. This key will be put in google_maps_api.xml



6. Design the layout and implement the activity

- Just look at the code generated by the new activity. ☺
- The main object (*mMap*) is obtained from **GoogleMap** class.
- A map can be obtained with the method **getMap()** from the class **SupportMapFragment**

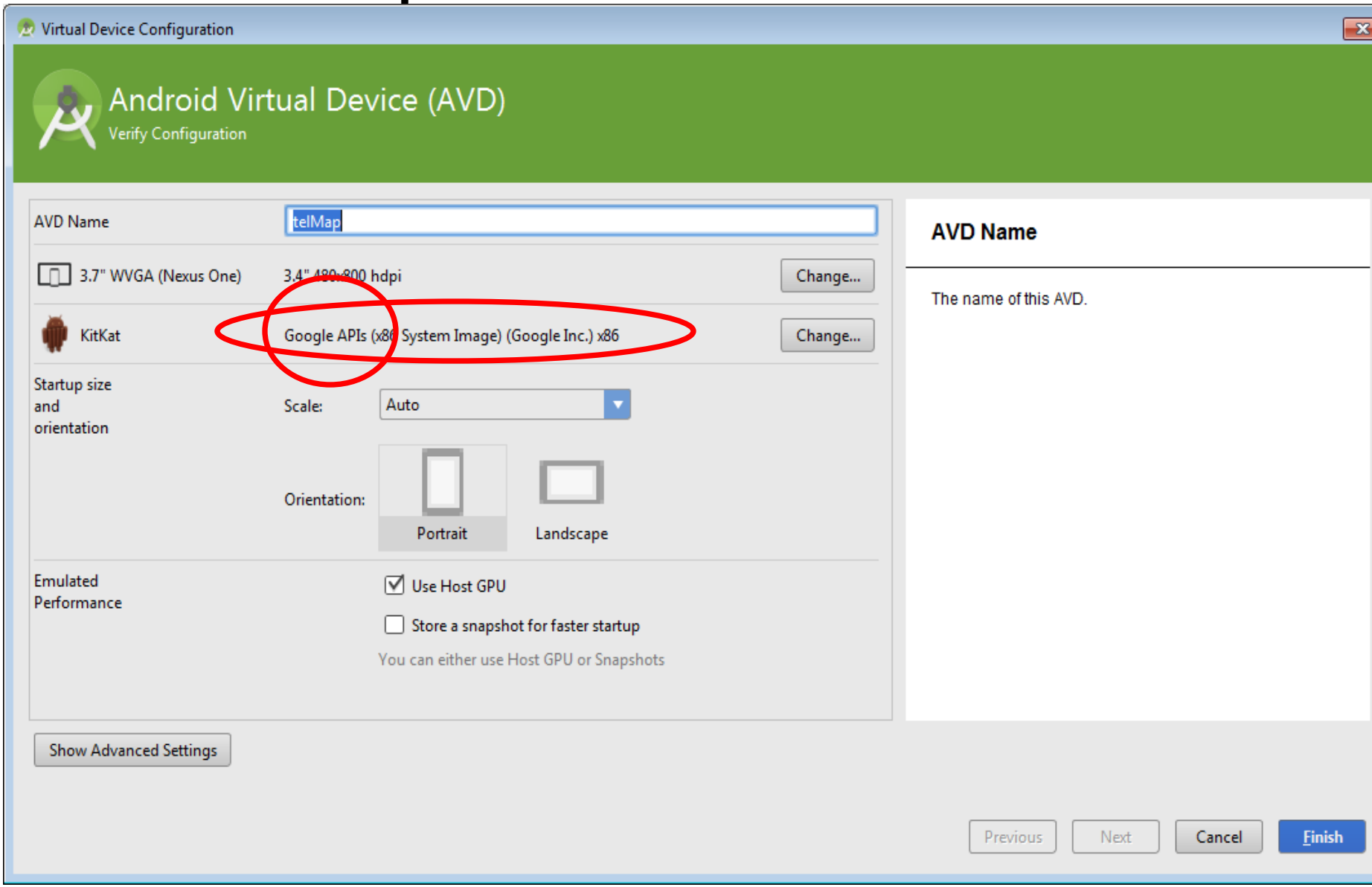
6. Design the layout and implement the activity

```
private GoogleMap mMap;

protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_frm_harta);
      setUpMapIfNeeded();
   }

 private void setUpMapIfNeeded() {
      // Do a null check to confirm that we have not already instantiated the map.
      if (mMap == null) {
         // Try to obtain the map from the SupportMapFragment.
         mMap = ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map)).getMap();
         // Check if we were successful in obtaining the map.
         if (mMap != null) {
            setUpMap();
         }
      }
   }

private void setUpMap() {
      mMap.addMarker(new MarkerOptions().position(new LatLng(45, 23)).title("Marker"));
      mMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(45, 24)));
     // mMap.animateCamera(CameraUpdateFactory.zoomBy(5));
   }
```

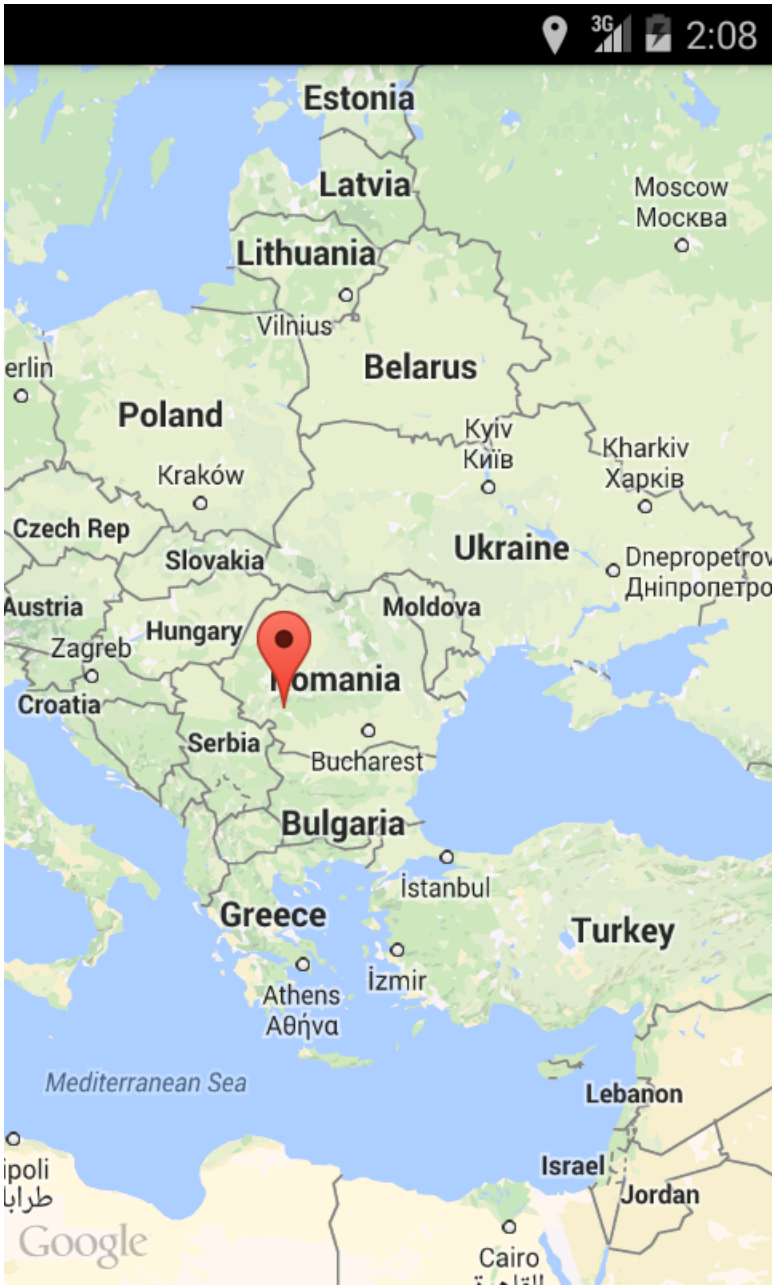
7. Run! ☺

- Prepare an emulator
- Launch the emulator
- Run the application
- Enjoy the result! ☺

Prepare the emulator



Enjoy the result ☺



Homework

- Read and implement about the **GoogleMap** class:

https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap

A (little) more sophisticated code example

- Implement the update of the map according to the user move;
- Use the Android Device Monitor to generate new locations;
- Test the applications and see the whole world!
- ☺

Code example

```
package com.example.appcursnoua_maps;
import com.google.android.gms.maps.GoogleMap;
//…
public class FrmHarta extends Activity {
	GoogleMap harta;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_frm_harta);
	}
	@Override
	protected void onResume() {
		// TODO Auto-generated method stub
		super.onResume();
		obtineHarta();
	}

	private void obtineHarta()
	{
		MapFragment fragmentHarta= (MapFragment) getFragmentManager().findFragmentById(R.id.map);
		harta=fragmentHarta.getMap();
		if(harta!=null)
		{
			harta.setMyLocationEnabled(true);
			Log.i("TAVY", "am setat enabled pentru locatie");
			harta.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(20.00, 20.00)));
		}
	}
```

Real time map update

```
package com.example.appcursnoua_maps;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.LatLng;

public class FrmHarta extends Activity implements android.location.LocationListener {

	GoogleMap harta;
	LocationManager locManager;

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

		locManager= (LocationManager) getSystemService(LOCATION_SERVICE);
	}
```

The implemenation goes on…

```
@Override
protected void onResume() {
	// TODO Auto-generated method stub
	super.onResume();
	obtineHarta();

	locManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,1000, 10,this);
}

private void obtineHarta()
{

	MapFragment fragmentHarta= (MapFragment) getFragmentManager().findFragmentById(R.id.map);
	harta=fragmentHarta.getMap();
	if(harta!=null)
	{
			harta.setMyLocationEnabled(true);
			Log.i("TAVY", "am setat enabled pentru locatie");
			harta.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(20.00, 20.00)));
	}
}
```

…and goes on…

```
    @Override
    public void onLocationChanged(Location location) {
            // TODO Auto-generated method stub
            //harta.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(location.getLatitude(), location.getLongitude())));
            harta.animateCamera(CameraUpdateFactory.newLatLng(new LatLng(location.getLatitude(), location.getLongitude())));
            Log.i("TAVY", "S-a schimbat locatia pe harta");
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
            // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String provider) {
            // TODO Auto-generated method stub
    }

    @Override
    public void onProviderDisabled(String provider) {
            // TODO Auto-generated method stub
    }

}
```

Useful links and tutorials

http://developer.android.com/google/play-services/setup.html

https://developers.google.com/maps/documentation/android/start#specify_app_settings_in_the_application_manifest

https://developers.google.com/maps/documentation/android-sdk/map

https://developers.google.com/maps/documentation/android/views#updating_the_camera_view

Link demo movie

https://www.youtube.com/watch?v=lchyOhPREh4&spfreload=10

A more sophisticated implementation

```
package com.example.user.apphartasdbissia;

    import android.Manifest;
    import android.content.pm.PackageManager;
    import android.graphics.Color;
    import android.location.Address;
    import android.location.Geocoder;
    import android.location.Location;
    import android.location.LocationListener;
    import android.location.LocationManager;
    import android.support.v4.app.ActivityCompat;
    import android.support.v4.app.FragmentActivity;
    import android.os.Bundle;
    import android.util.Log;
    import android.widget.Toast;

    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.google.android.gms.maps.model.PolylineOptions;

    import java.io.IOException;
    import java.util.List;
    import java.util.Locale;
```

```
public class FrmHarta extends FragmentActivity implements OnMapReadyCallback, GoogleMap.OnMapClickListener,
        LocationListener {

    private GoogleMap mMap;
    private static PolylineOptions linie;
    private double totalDistance = 0;
    LocationManager locManager;

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

      locManager = (LocationManager) getSystemService(LOCATION_SERVICE);

      linie = new PolylineOptions();

      // 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);
    }
```

```
@Override
    public void onMapReady (GoogleMap googleMap) {
      mMap = googleMap;
      mMap.setOnMapClickListener(this);

      // Add a marker in Sydney and move the camera
      LatLng startingPoint = new LatLng(47.13, 27.53);
      mMap.addMarker(new MarkerOptions().position(startingPoint).title("Marker in Iasi"));

      linie.add(startingPoint);

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

      if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
        //                          int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
      Log.i("TAVY","Permission not granted");
        return;
      }
      Log.i("TAVY","Permission granted");
      locManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 10, this);
      Location loc=locManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    }
```

```
@Override
        public void onMapClick (LatLng newPoint) {
           //here we implement the behaviour of the map
           //when the user just clicks on it
           //first of all, let's get the new location (the new point on the map)
           computeAndDisplayPoints(newPoint);
        }
```

```
public void computeAndDisplayPoints (LatLng newPoint)
        {
          Log.i("TAVY"," The new point is at: " + newPoint.latitude + " " + newPoint.longitude);
          //we add ne the Point on the existing polyline
          //then we draw the new polyline on the map

          //calculate the distance
          Location p1,p2;
          p1=new Location("");
          LatLng lastAddress = linie.getPoints().get( linie.getPoints().size() - 1 );

          p1.setLatitude(lastAddress.latitude); p1.setLongitude(lastAddress.longitude);
          p2=new Location("");
          p2.setLatitude(newPoint.latitude);p2.setLongitude(newPoint.longitude);
          //check if Cristian is very close to Miruna ☺
          Location Miruna=new Location("");
          Miruna.setLatitude(47.1702);
          Miruna.setLongitude(27.5758001);
          if(p1.distanceTo(Miruna)<1000)
          {
            //yes! I'm very close to Miruna
            Toast.makeText(this, "Hello Miruna!!!", Toast.LENGTH_LONG).show();
          }
          //let's geocode the point p1
          Geocoder codificator;
          codificator = new Geocoder(this, Locale.getDefault());
          List<Address> listaAdresePosibile = null;
          try {
            listaAdresePosibile = codificator.getFromLocation(p2.getLatitude(),p2.getLongitude(),1);
            for(int i=0;i<listaAdresePosibile.size();i++)
            {
              Log.i("TAVY","Adresa curenta este: " + listaAdresePosibile.get(i).getCountryName() + " " + listaAdresePosibile.get(i).getLocality());
            }
          } catch (IOException e) {
            e.printStackTrace();
          }
          totalDistance = totalDistance + p1.distanceTo(p2);
          Toast.makeText(this, "Distance=" + totalDistance/1000, Toast.LENGTH_LONG).show();
          linie.add(newPoint);
          linie.width(20);
          linie.color(Color.RED);
          mMap.addPolyline(linie);
          mMap.moveCamera(CameraUpdateFactory.newLatLng(newPoint));
        }
```

```
@Override
        public void onLocationChanged (Location location) {
           //if the location changes, we call the method that was defined by us
           Log.i("TAVY","Locatie modificata");
           LatLng newLocation= new LatLng(location.getLatitude(),location.getLongitude());
           computeAndDisplayPoints(newLocation);
        }

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

        }

        @Override
        public void onProviderEnabled(String provider) {

        }

        @Override
        public void onProviderDisabled(String provider) {

        }
    }
```