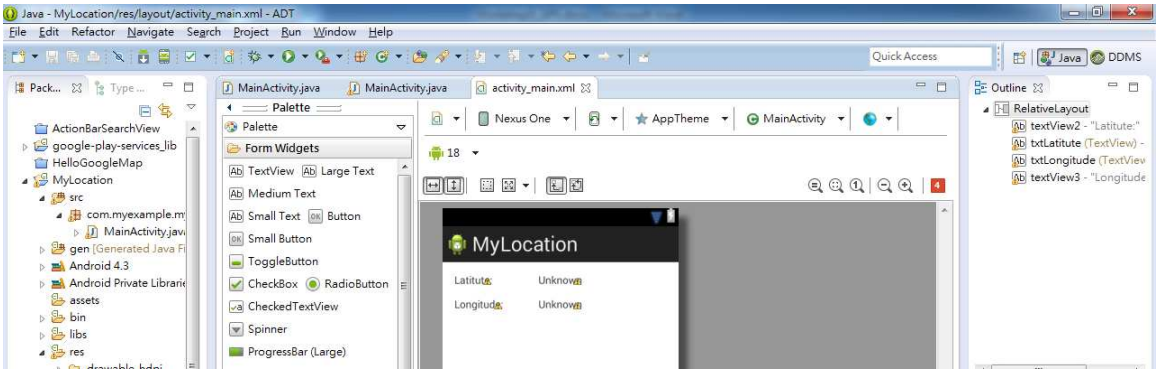
1. Location Services

1.1 GPS Location

1. Create the Android application with the following attributes.
   - Application Name: **MyLocation**
   - Project Name: **MyLocation**
   - Package Name: **com.example.mylocation**

2. Put 4 text views into the layout as follow:



3. The XML code for the layout look like:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Latitude:" />

    <TextView
        android:id="@+id/txtLatitude"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView2"
        android:layout_marginLeft="19dp"
        android:layout_toRightOf="@+id/textView2"
```

```
        android:text="Unknown" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="18dp"
        android:text="Longitude:" />

    <TextView
        android:id="@+id/txtLongitude"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView1"
        android:layout_alignBottom="@+id/textView1"
        android:layout_alignLeft="@+id/txtLatitude"
        android:text="Unknown" />
</RelativeLayout>
```

4. Add the following user permission in "**AndroidManifest.xml**":
   - android.permission.INTERNET
   - android.permission.ACCESS_FINE_LOCATION

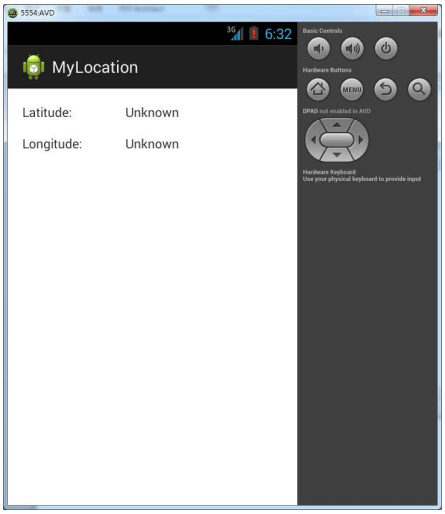
5. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mylocation;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.widget.TextView;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;

public class MainActivity extends Activity implements LocationListener {
```

```
    private double latitude, longitude;
    private TextView txtLatitude;
    private TextView txtLongitude;
    private LocationManager locationManager;
    private String provider;

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

        txtLatitude = (TextView) findViewById(R.id.txtLatitude);
        txtLongitude = (TextView) findViewById(R.id.txtLongitude);

        // Get the location manager
        locationManager = (LocationManager)
                    getSystemService(Context.LOCATION_SERVICE);

       // Define the criteria how to select the location provider
        Criteria criteria = new Criteria();
        provider = locationManager.getBestProvider(criteria, false);
        Location location = locationManager.getLastKnownLocation(provider);

        // Initialize the location fields
        if (location != null) {
            // Provider selected
            onLocationChanged(location);
        } else {
            txtLatitude.setText("Unknown");
            txtLongitude.setText("Unknown");
        }
    }

    @Override
    protected void onResume() {
        // Request updates at startup
        super.onResume();
        locationManager.requestLocationUpdates(provider, 400, 1, this);
    }
```

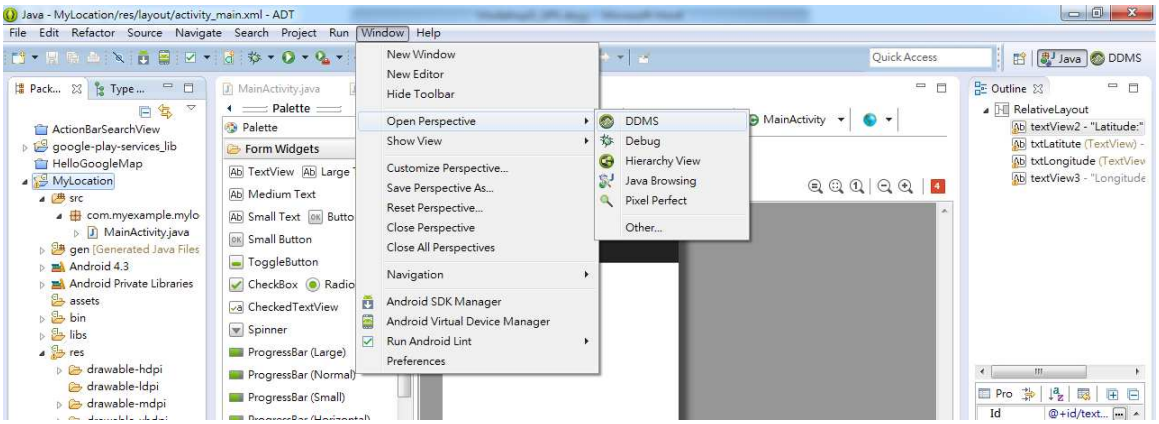
```
    @Override
    protected void onPause() {
        // Remove the location listener updates when Activity is paused
        super.onPause();
        locationManager.removeUpdates(this);
    }

    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        latitude = (double) (location.getLatitude());
        longitude = (double) (location.getLongitude());
        txtLatitude.setText(String.valueOf(latitude));
        txtLongitude.setText(String.valueOf(longitude));
    }

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

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

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

6. Save and execute the app, can you see the location information in emulator? Install this app into real mobile, and turn on GPS, can you see the location information correctly?

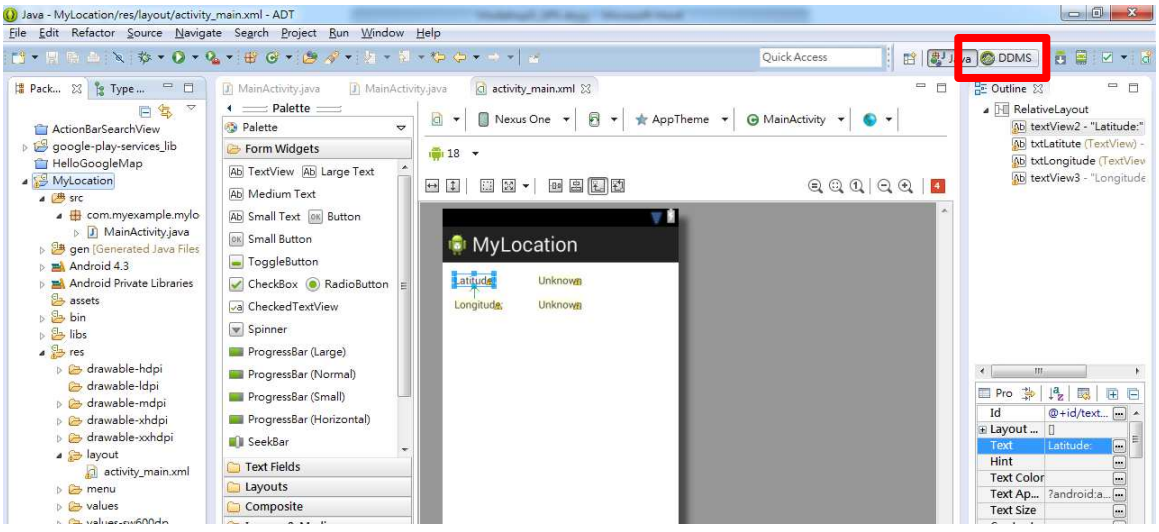


1.2 Mock Location in AVD

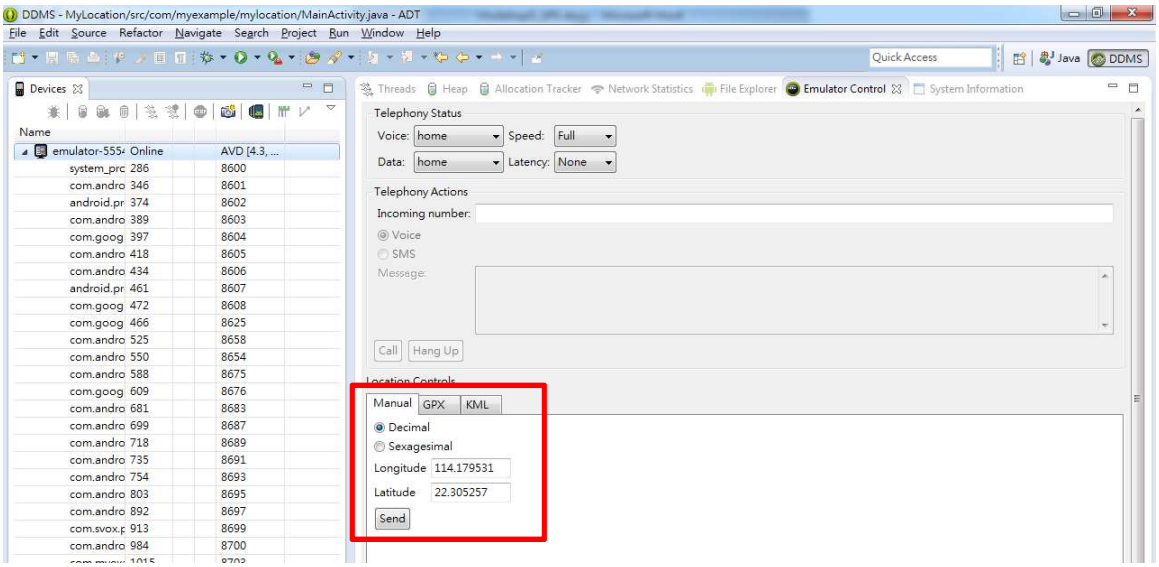
1. Select **Windows ➔ Open Perspective ➔ DDMS**.



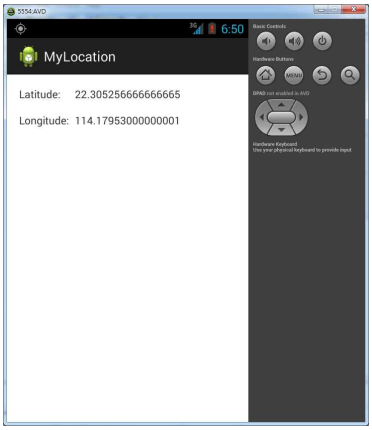
2. Select the **DDMS** tab.



3. Input "**114.179531**" and "**22.305257**" as the longitude and latitude, and then press "**Send**".

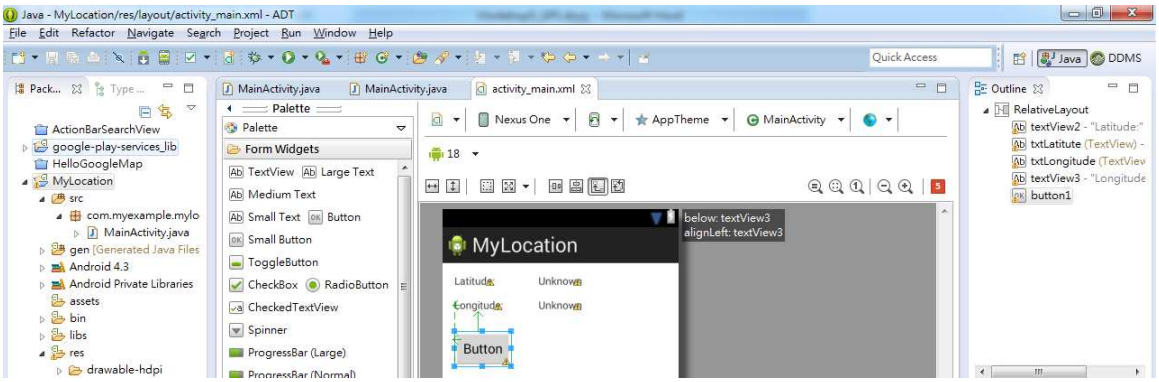


4. Switch to the app again, the location should be display correctly.



1.3 Open the Web-based Google Map

1. Drag a button to the layout



2. Modify the source file "**MainActivity.java**" as follow:

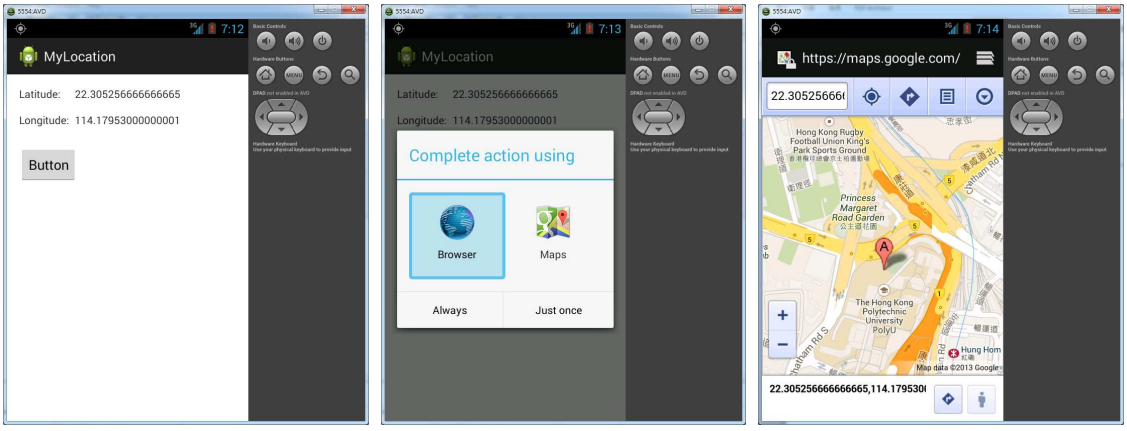
```
package com.example.mylocation;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.widget.TextView;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.widget.Button;
import android.net.Uri;
import android.view.View;
import android.view.View.OnClickListener;
import android.content.Intent;

public class MainActivity extends Activity implements LocationListener {
    private double latitude, longitude;
    private TextView txtLatitude;
    private TextView txtLongitude;
    private LocationManager locationManager;
    private String provider;
    private Button button1;

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

        txtLatitude = (TextView) findViewById(R.id.txtLatitute);
        txtLongitude = (TextView) findViewById(R.id.txtLongitude);

        // Create the button and listener
        button1 = (Button) findViewById(R.id.button1);
        button1.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                drawGoogleMap(latitude, longitude);
            }
       });
```

```
        // Get the location manager
        locationManager = (LocationManager)
                        getSystemService(Context.LOCATION_SERVICE);

        // Define the criteria how to select the location provider
        Criteria criteria = new Criteria();
        provider = locationManager.getBestProvider(criteria, false);
        Location location = locationManager.getLastKnownLocation(provider);

        // Initialize the location fields
        if (location != null) {
            // Provider selected
            onLocationChanged(location);
        } else {
            txtLatitude.setText("Unknown");
            txtLongitude.setText("Unknown");
        }
    }

    public void drawGoogleMap(double latitude, double longitude){
        String myGeoCode = "https://maps.google.com/maps?q="
            + latitude + "," + longitude + "(You are here!)&iwloc=A&hl=en";
        Intent intentViewMap = new Intent(Intent.ACTION_VIEW, Uri.parse(myGeoCode));
        startActivity(intentViewMap);
    }

    @Override
    protected void onResume() {
        // Request updates at startup
        super.onResume();
        locationManager.requestLocationUpdates(provider, 400, 1, this);
    }

    @Override
    protected void onPause() {
        // Remove the location listener updates when Activity is paused
        super.onPause();
        locationManager.removeUpdates(this);
    }
```

```
    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        latitude = (double) (location.getLatitude());
        longitude = (double) (location.getLongitude());
        txtLatitude.setText(String.valueOf(latitude));
        txtLongitude.setText(String.valueOf(longitude));
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

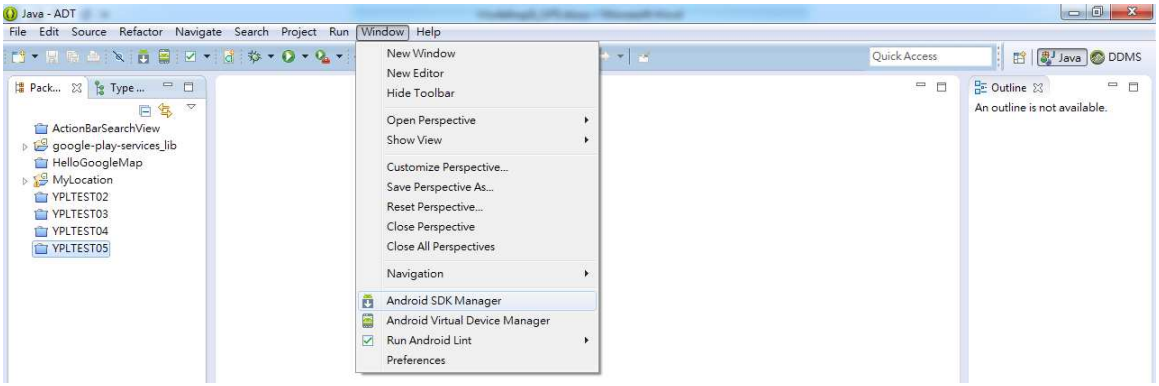
3. Save and execute the app, then use DDMS to send the GPS location to the emulator.



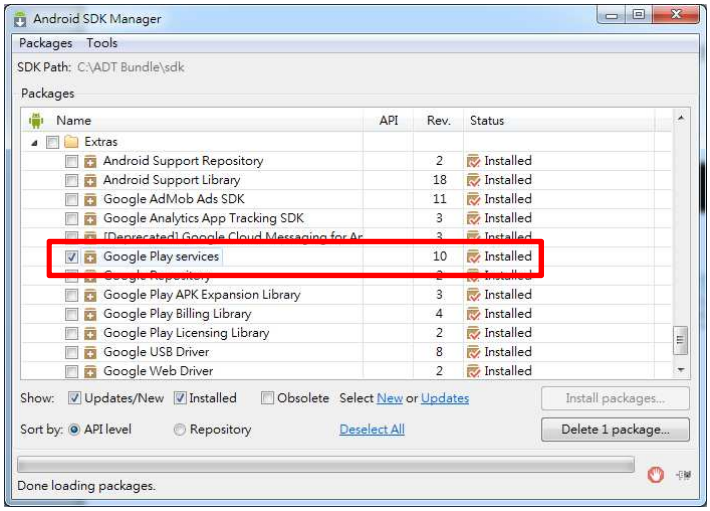
2. Google Maps

2.1 Install the Google Play Services

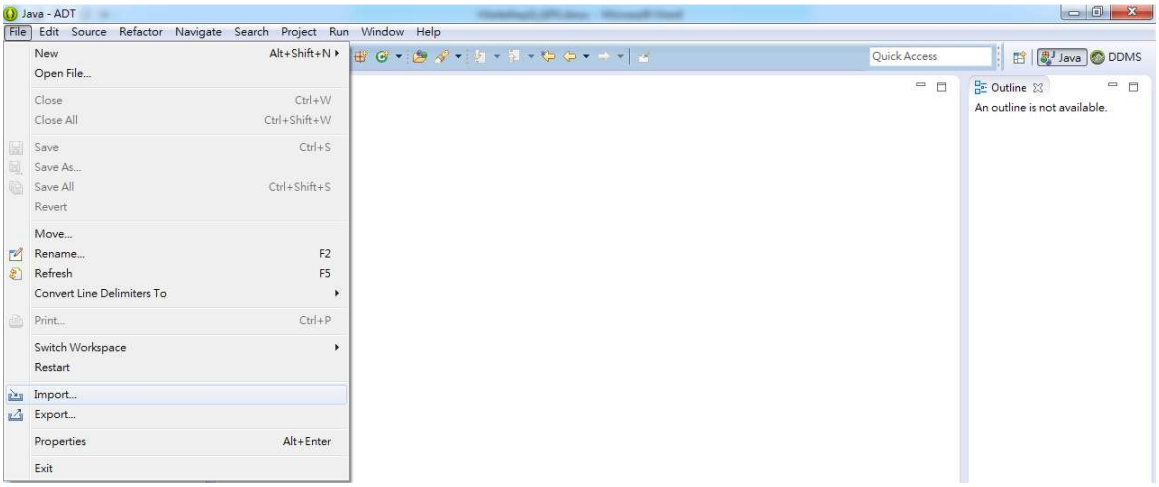
1. Select **Windows ➔ Android SDK Manager** from ADT.



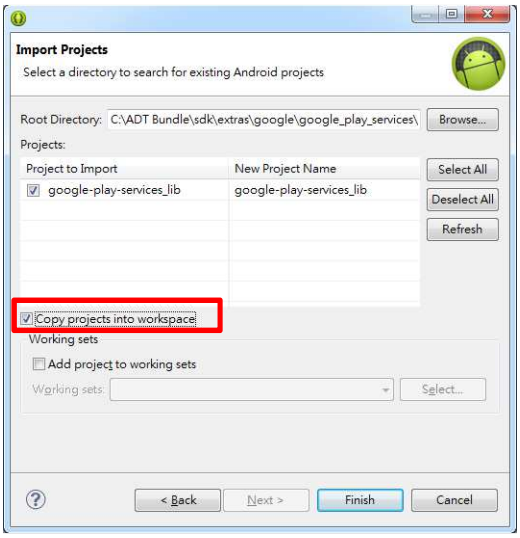
2. Scroll down to bottom, then select **Extras ➔ Google Play Services** and install it. Remember to restart your SDK when complete.



3. Select **File ➔ Import** from ADT.



4. Browse the library folder (<android-sdk>/extras/google/google_play_services/libproject/), select the project "**google-play-services_lib**" and press "**Finish**" to import. Remember to select "**Copy projects into workspace**"



2.2 Obtain the Google Maps API Key

1. Open the DOS prompt by "**cmd**", and then locate to your debug keystore file (*Normally store in C:\Users\[User Name]\.android\debug.keystore*). Then execute the corresponding keytool command (*normally located in %JAVA_HOME%\bin*):

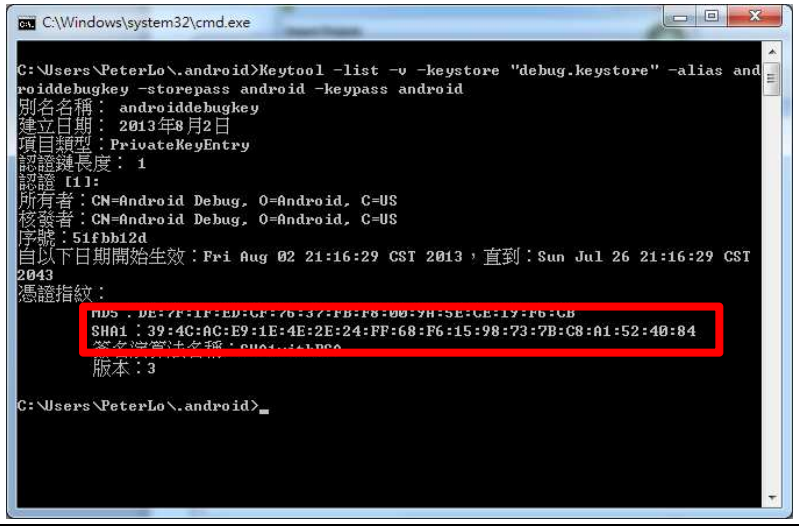
For Debug Certificate Fingerprint

```
Keytool -list -v -keystore "debug.keystore" -alias androiddebugkey -storepass android
-keypass android
```

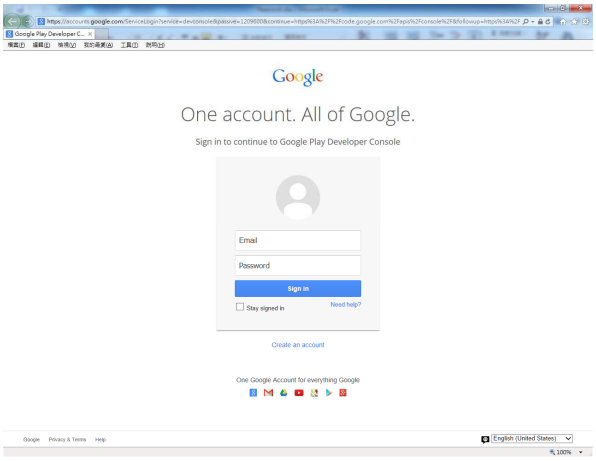
For Release Certificate Fingerprint

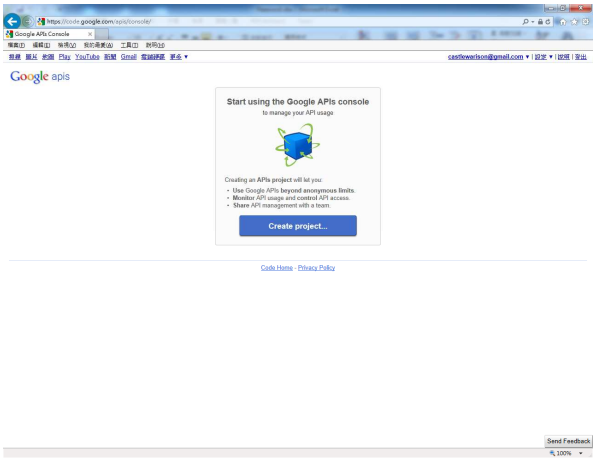
```
Keytool -list -v -keystore [keystore Name] -alias [Alias Name]
```

2. The Maps API key is based on the short form of the application's digital certificate which called SHA-1 fingerprint. Remember to copy the SHA1 fingerprint.

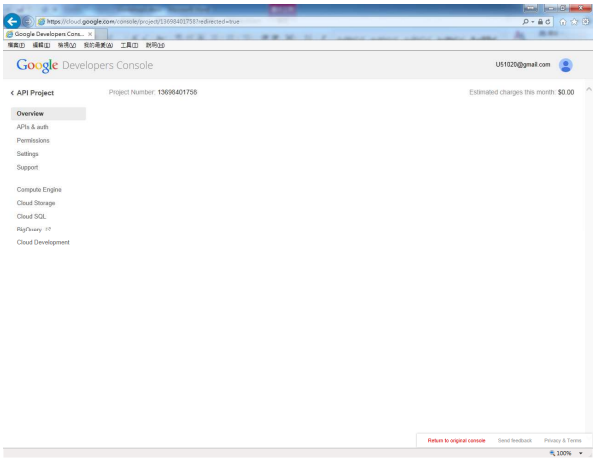


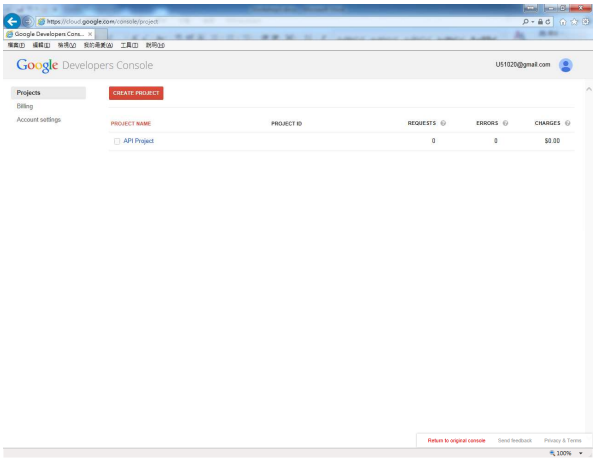
3. Go to Google API Console (https://code.google.com/apis/console) to register the project, press "**Create project**" to continue (Google ID is required to access this page).



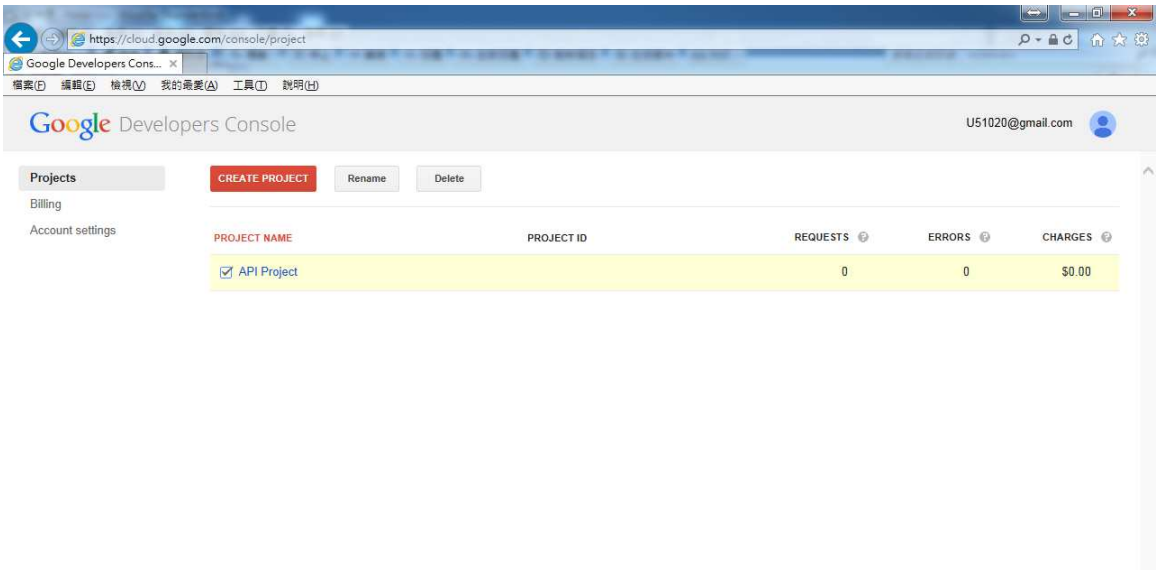


4. The new project will be created, press **API Project** to return to the project main page.

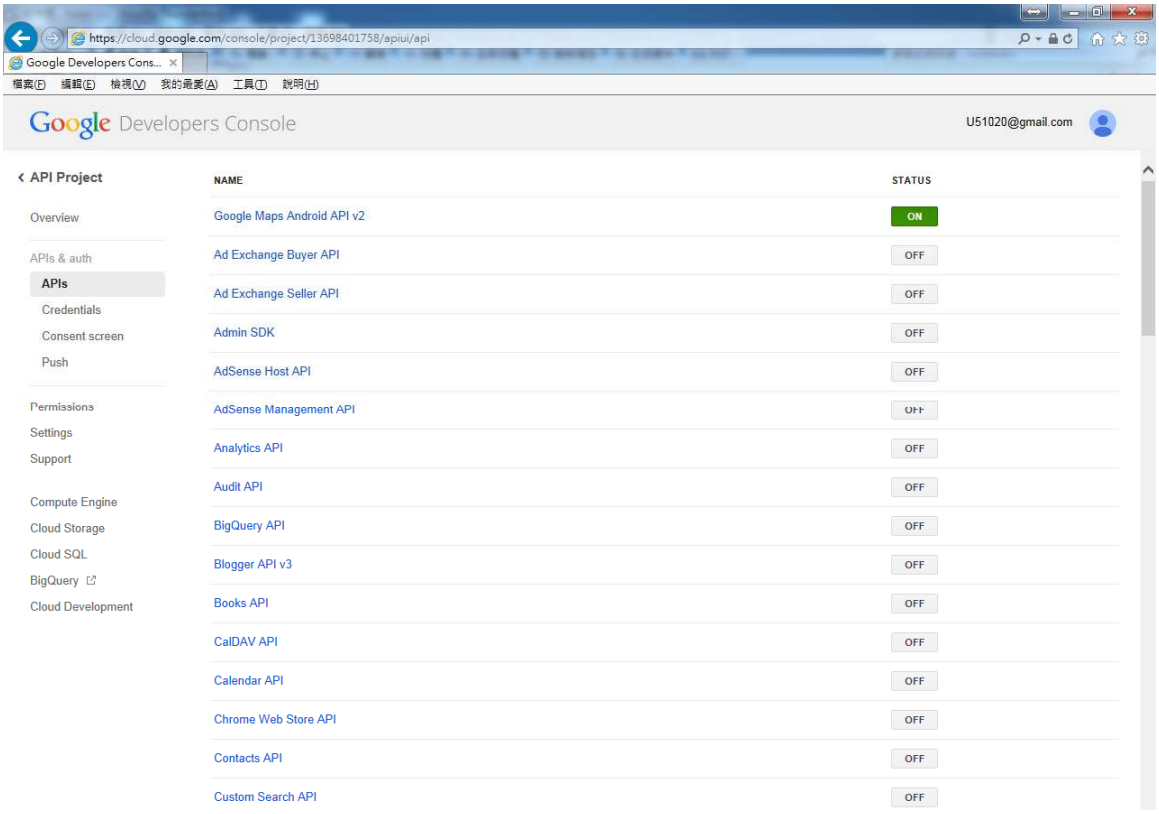




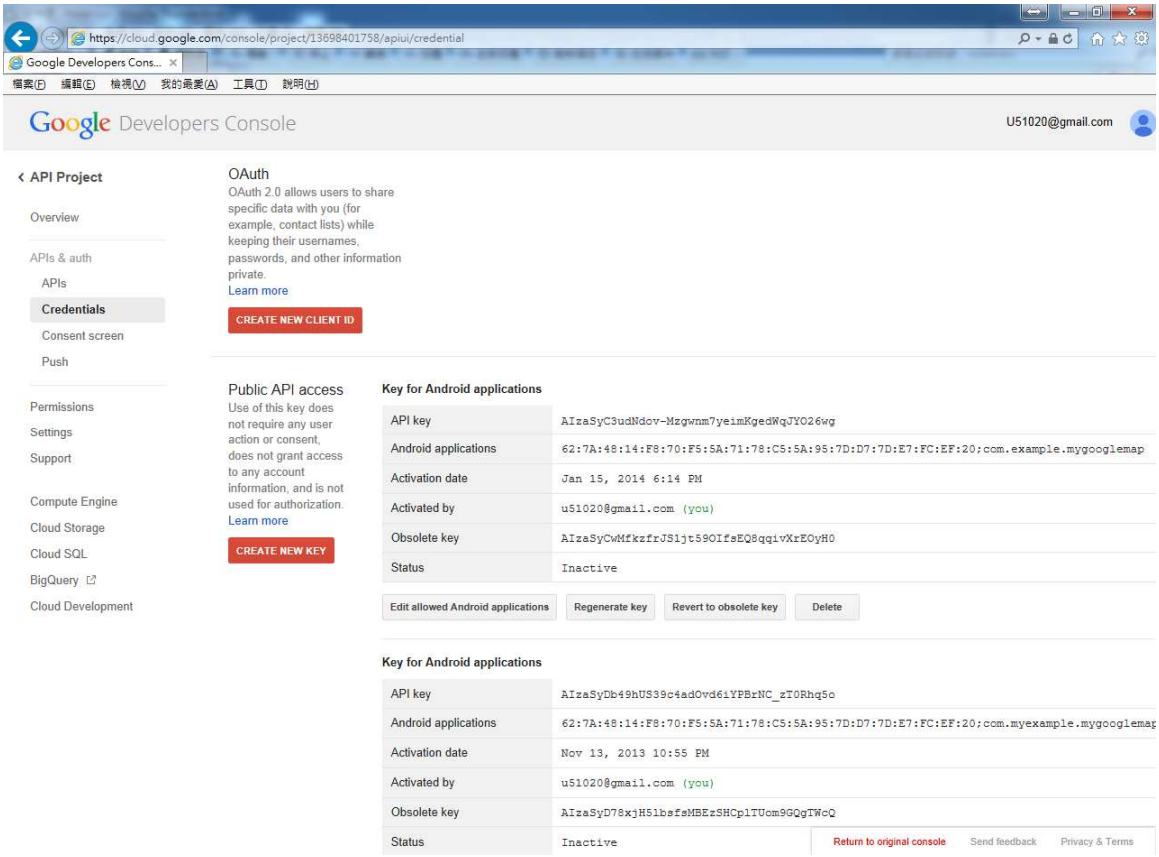
5. Select the created project and click on it.



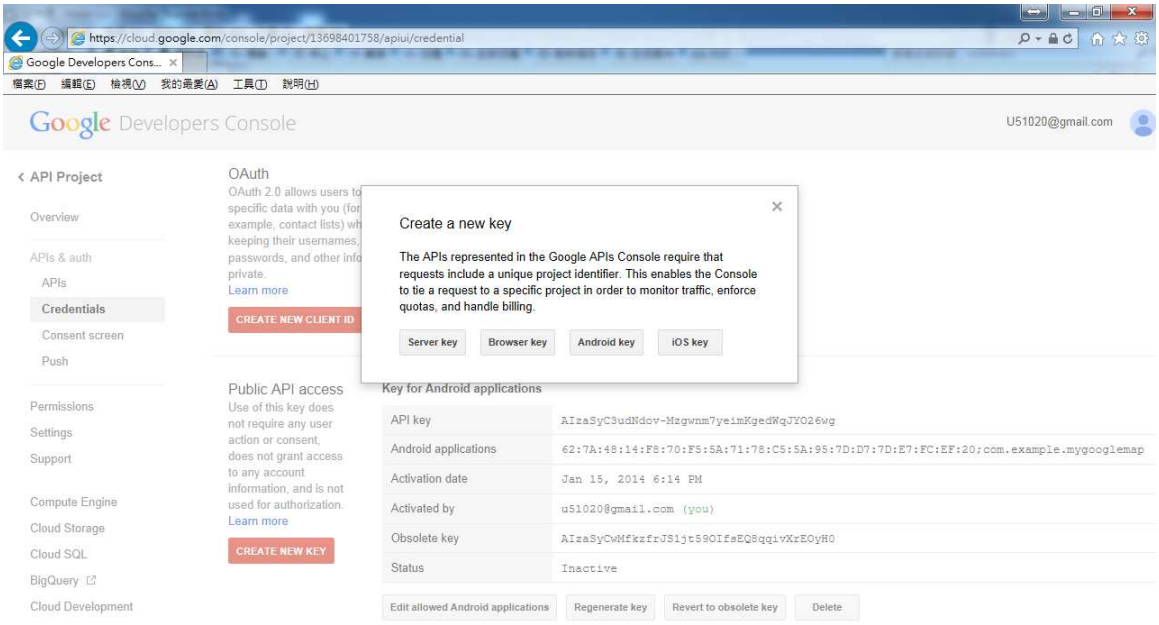
6. Select **APIs & auth ➔ APIs**, and then switch on the **Google Maps Android API v2**.



7. Select **Credentials**, and then press [CREATE NEW LEY].



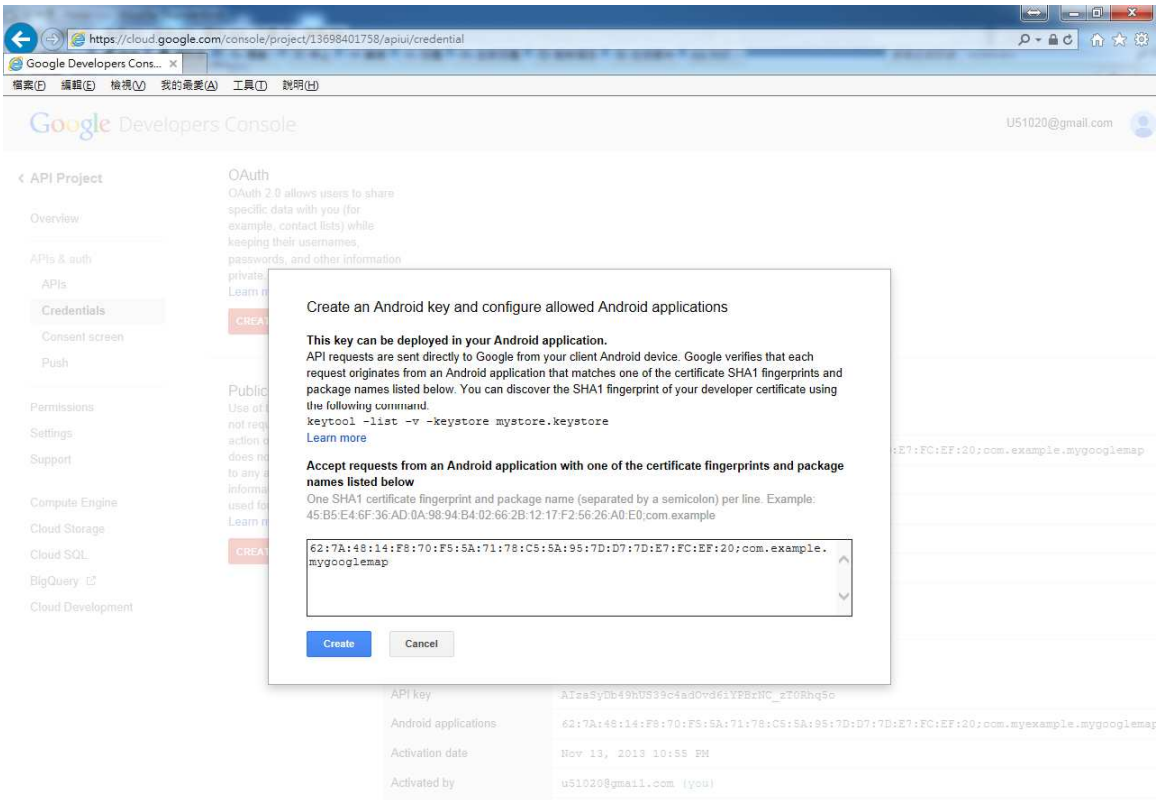
8. Select **Android key**.



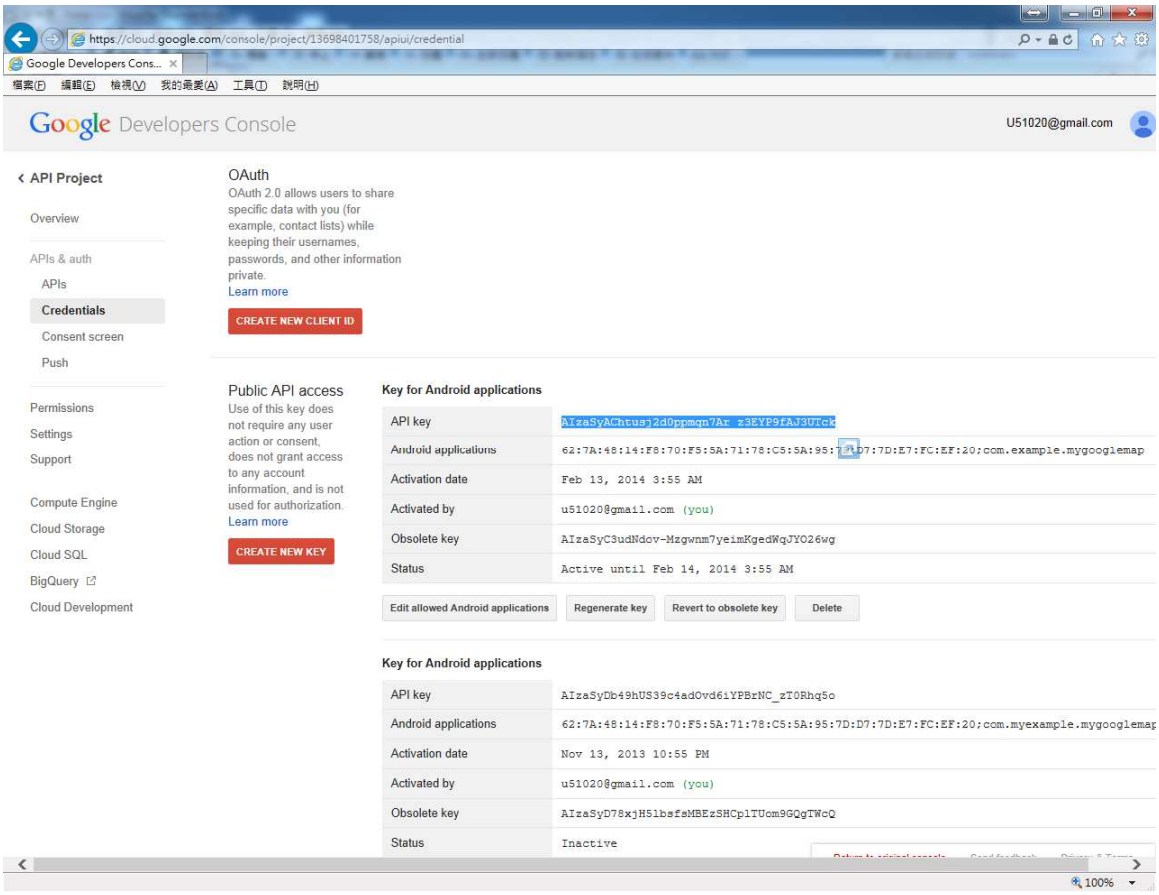
9. Generate the request key by concatenate the Certificate fingerprints and package name which separated by a semicolon:

```
[SHA1 fingerprint];[app's package name]
```

10. Input the request key and then press "**Create**".



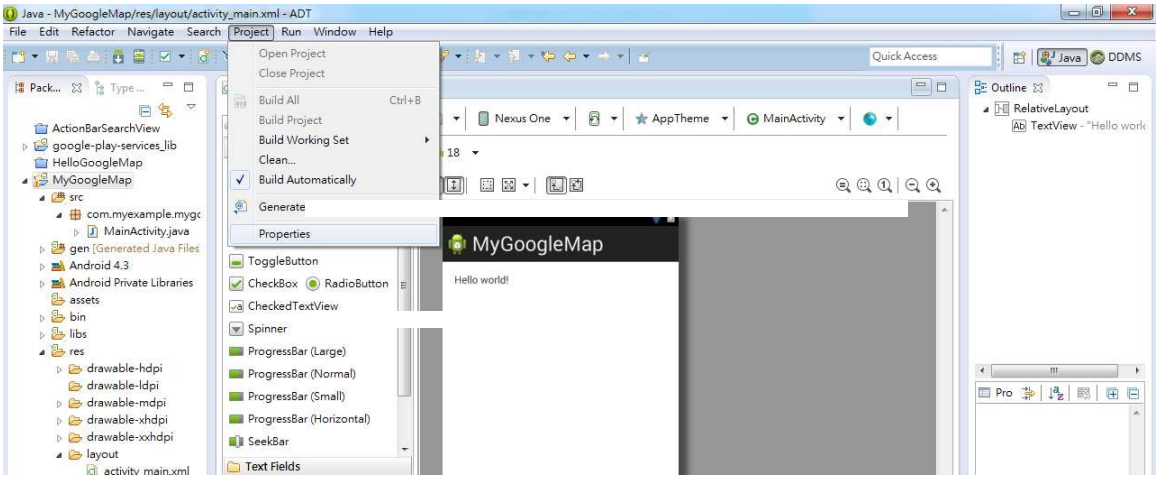
11. Copy the API key generated.



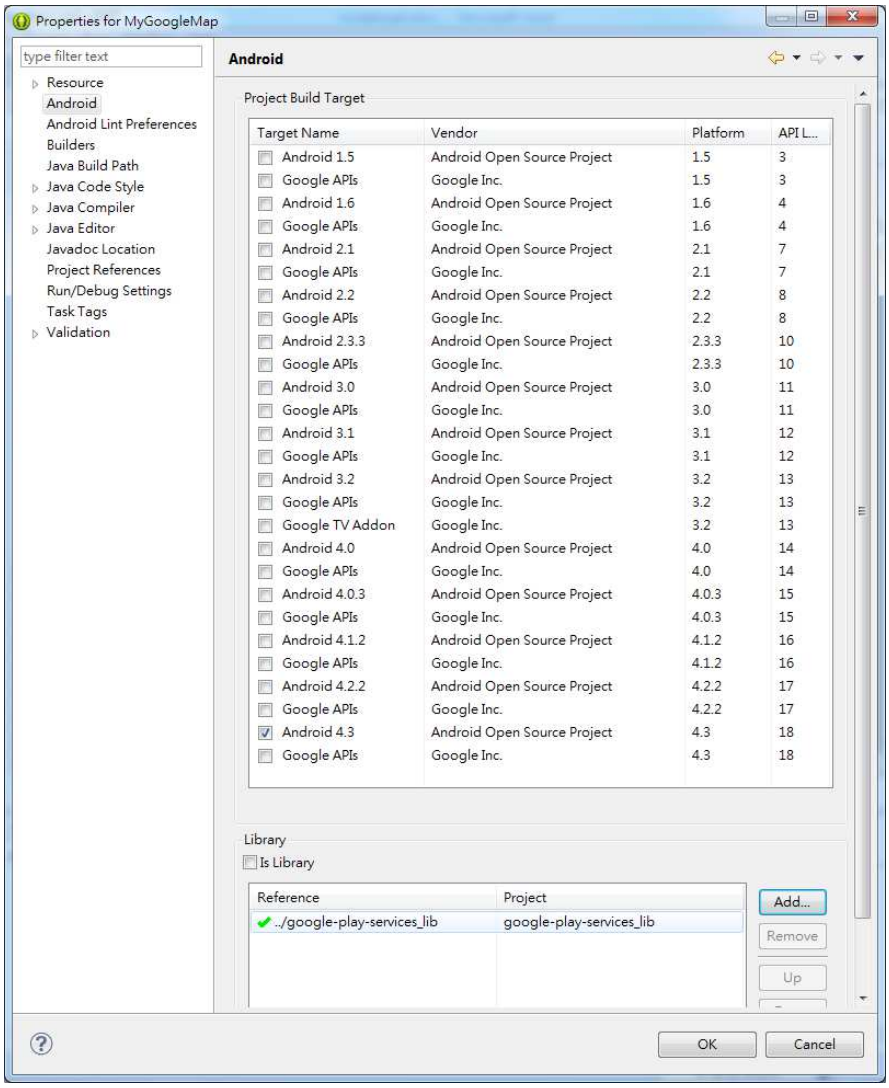
2.3 Create a Simple Google Map

1. Create the Android application with the following attributes.
   - Application Name: **MyGoogleMap**
   - Project Name: **MyGoogleMap**
   - Package Name: **com.example.mygooglemap**

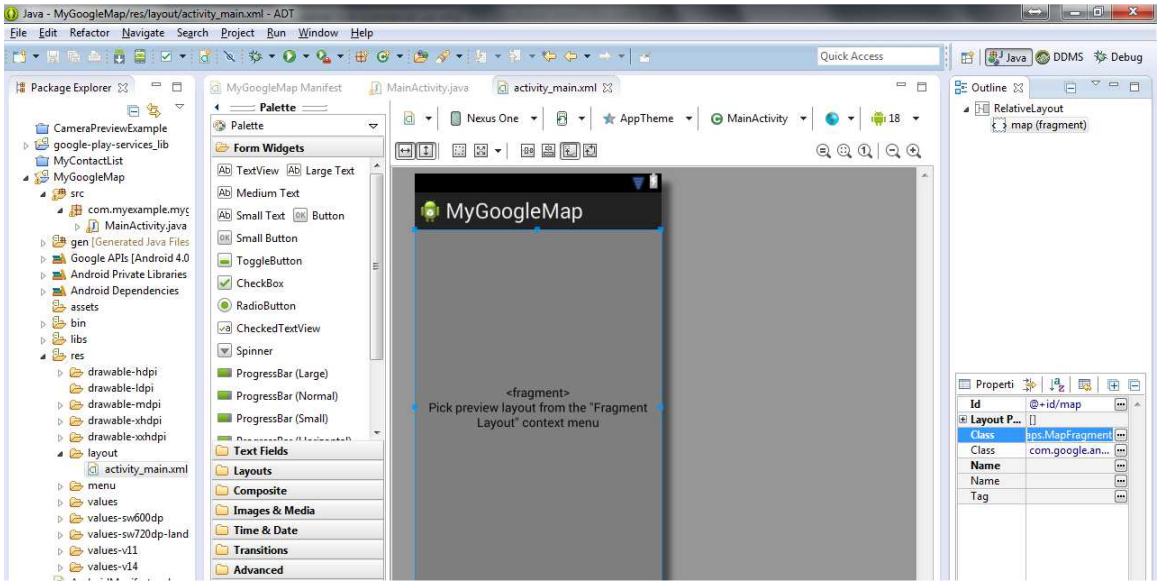
2. Select **Project ➔ Properties**.



3. Add "**google-play-services_lib**" to the project.



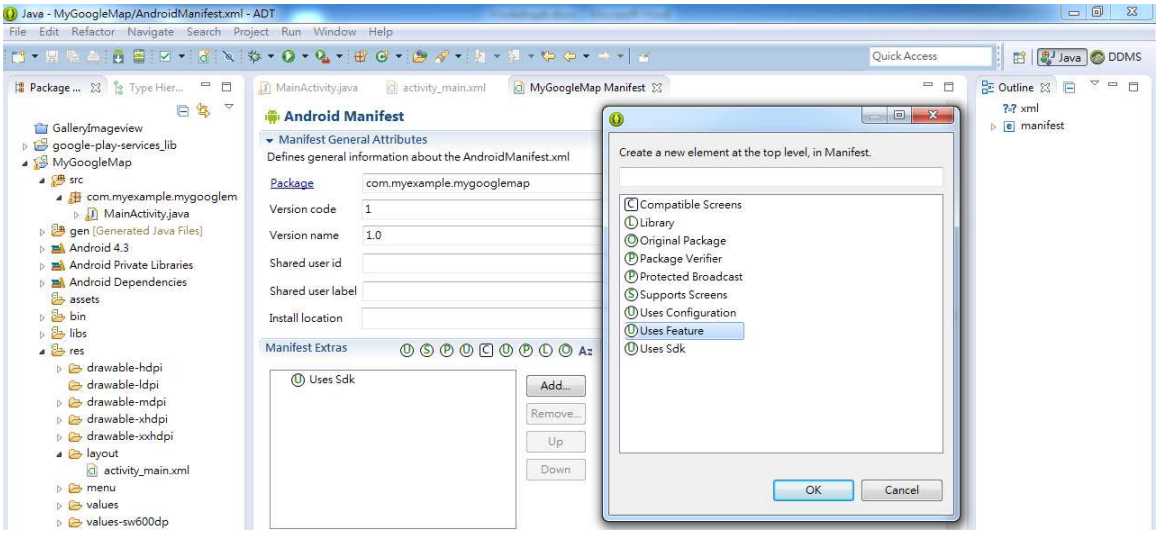
4. Remove the default Textview "Hello world!". Then create a fragment with class "**com.google.android.gms.maps.MapFragment**" in layout.



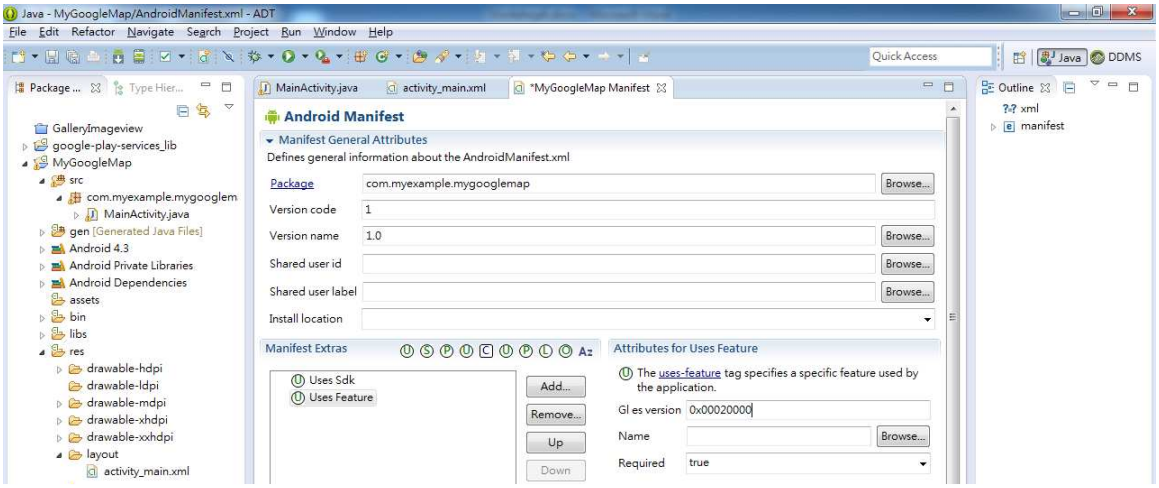
5. The XML for the layout look like:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
     xmlns:tools="http://schemas.android.com/tools"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     tools:context=".MainActivity" >

    <fragment
         android:id="@+id/map"
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         class="com.google.android.gms.maps.MapFragment" />
</RelativeLayout>
```

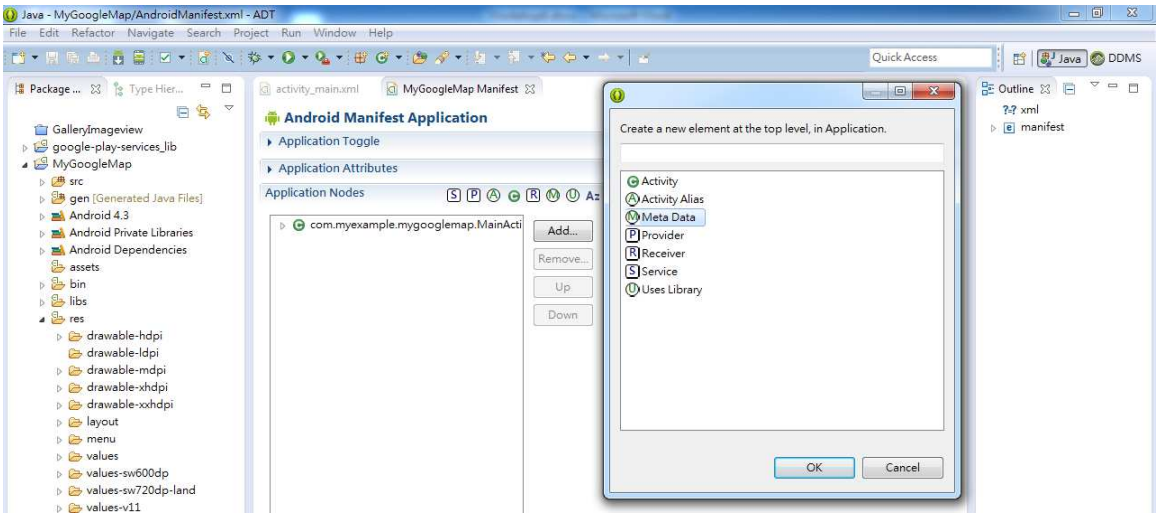
6. Open "**AndroidManifest.xml**" and switch to "**Manifest**", then create an "**Uses-feature**".



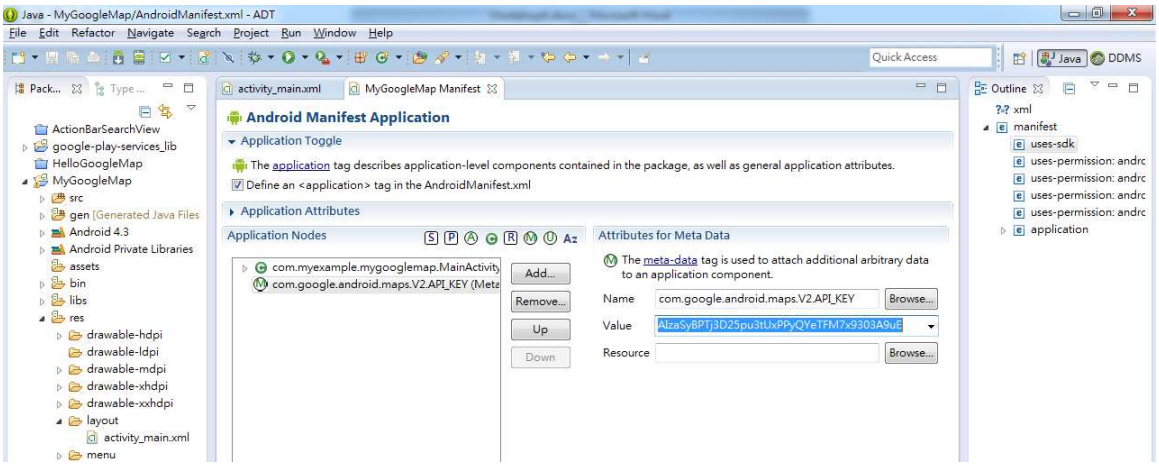
7. Set Gl es Version = "**0x00020000**" and Required = "**true**".



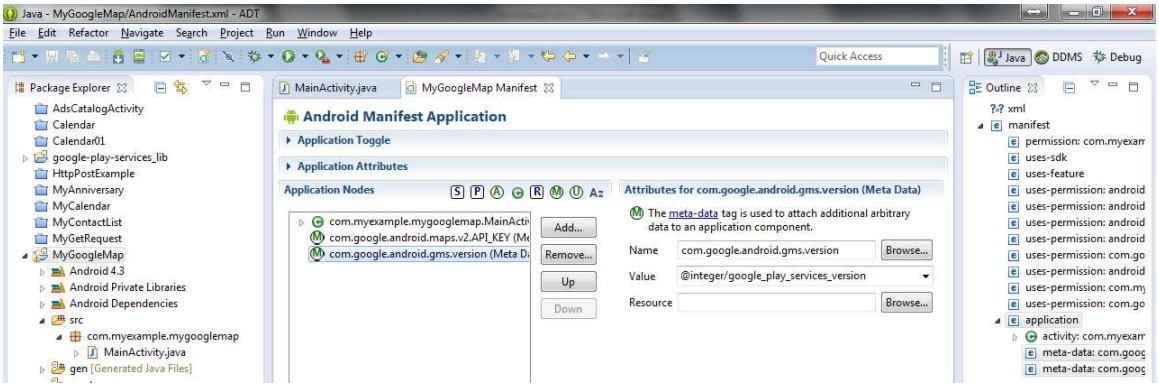
8. Open "**AndroidManifest.xml**" and switch to "**Application**", then create a "**Meta-Data**".



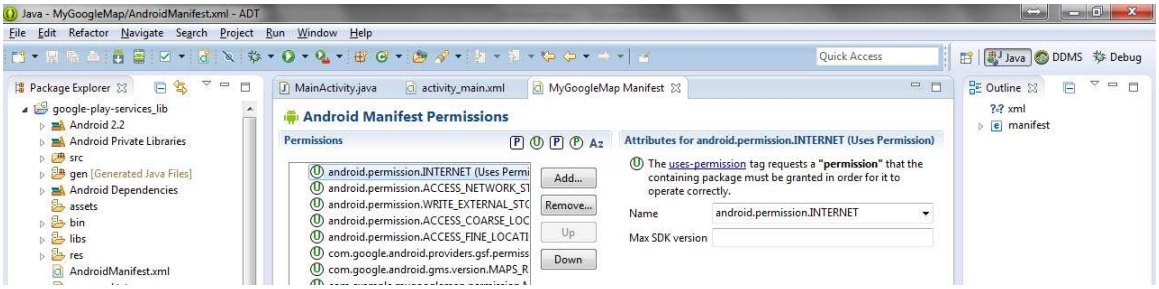
9. Create the Meta-Data with the following attributes in "**AndroidManifest.xml**".
   - Name: **com.google.android.maps.v2.API_KEY**
   - Value: **[API Key]**



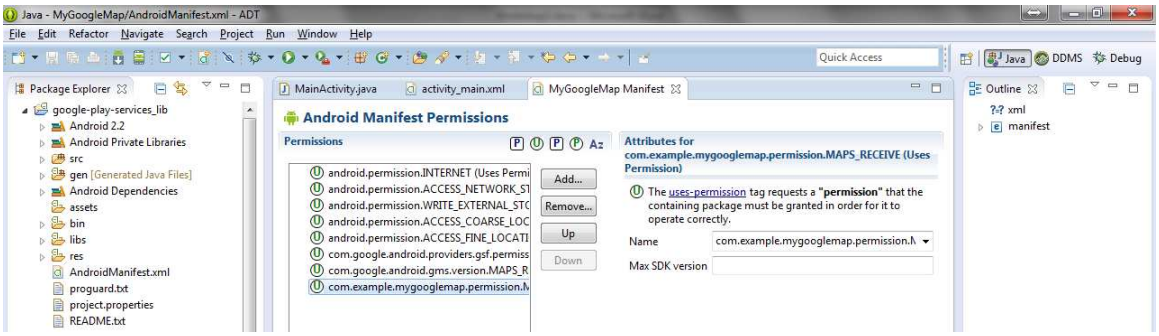
10. Create another Meta-Data with the following attributes in "**AndroidManifest.xml**".
   - Name: **com.google.android.gms.version**
   - Value: **@integer/google_play_services_version**



11. Add the following uses permission in "**AndroidManifest.xml**":
    - android.permission.INTERNET
    - android.permission.ACCESS_NETWORK_STATE
    - android.permission.WRITE_EXTERNAL_STORAGE
    - android.permission.ACCESS_COARSE_LOCATION
    - android.permission.ACCESS_FINE_LOCATION
    - com.google.android.providers.gsf.permission.READ_GSERVICES
    - com.google.android.gms.version.MAPS_RECEIVE



12. Add the permission for your application in "**AndroidManifest.xml**", you need to change ***com.example.mygooglemap*** to your package name.
    - com.example.mygooglemap.permission.MAPS_RECEIVE

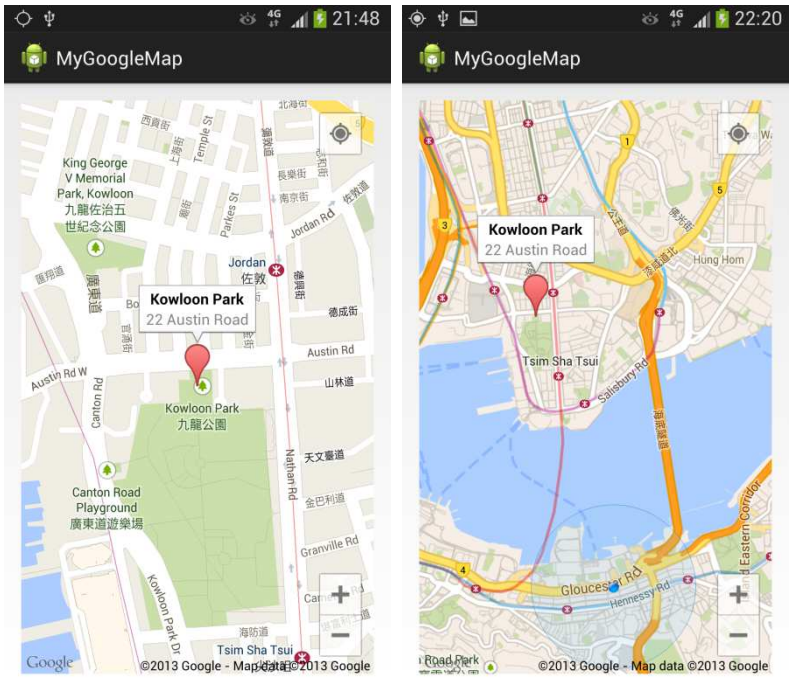


13. Modify the source file "**MainActivity.java**" as follow:

```
package com.example.mygooglemap;


import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
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.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
```

```
public class MainActivity extends Activity {
    static final LatLng location = new LatLng(22.302617,114.17009);
    private GoogleMap map;

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

        // Define the map
        map = ((MapFragment)
            getFragmentManager().findFragmentById(R.id.map)).getMap();

        // Create Pointer
        Marker pointer = map.addMarker(new MarkerOptions().position(location)
            .title("Kowloon Park").snippet("22 Austin Road"));

        // Change Pointer color
        pointer.setIcon(BitmapDescriptorFactory.defaultMarker(
            BitmapDescriptorFactory.HUE_RED));

        // Move the camera instantly to location with a zoom of 16.
        map.moveCamera(CameraUpdateFactory.newLatLngZoom(location, 16));

        // Show My Location button (False to disable)
        map.getUiSettings().setMyLocationButtonEnabled(true);

        // Show current location (False to disable)
        map.setMyLocationEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

14. Save and execute the app in your mobile. You can press the GPS to find out your location. If you cannot see the map, check whether your key is correct. (This app is not workable in emulator)



2.4 Changing Map Type

1. Modify the source file "**MainActivity.java**" as follow:

```
package com.myexample.mygooglemap;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
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.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends Activity {
    static final LatLng location = new LatLng(22.302617,114.17009);
    private GoogleMap map;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Define the map
        map = ((MapFragment)
            getFragmentManager().findFragmentById(R.id.map)).getMap();

        // Create Pointer
        Marker pointer = map.addMarker(new MarkerOptions().position(location)
            .title("Kowloon Park").snippet("22 Austin Road"));

        // Change Pointer color
        pointer.setIcon(BitmapDescriptorFactory.defaultMarker(
            BitmapDescriptorFactory.HUE_RED));

        // Move the camera instantly to location with a zoom of 16.
        map.moveCamera(CameraUpdateFactory.newLatLngZoom(location, 16));

        // Show My Location button (False to disable)
        map.getUiSettings().setMyLocationButtonEnabled(true);

        // Show current location (False to disable)
        map.setMyLocationEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Add the menu items
        menu.add(0, 1, 1, "Normal");
        menu.add(0, 2, 2, "Hybrid");
        menu.add(0, 3, 3, "Satellite");
        menu.add(0, 4, 4, "Terrain");
        menu.add(0, 5, 5, "None");

        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
```

```
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()) {
            case 1:
                map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                break;
            case 2:
                map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                break;
            case 3:
                map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                break;
            case 4:
                map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                break;
            case 5:
                map.setMapType(GoogleMap.MAP_TYPE_NONE);
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

2. Save and execute the app again. Use the menu to change the map type (This app is not workable in emulator).