**MainActivity.java**

**package** com.example.waqas03312033721.googlemapapp;  
  
**import** android.support.v4.app.FragmentActivity;  
**import** android.os.Bundle;  
  
**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;  
  
**public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback {  
  
 **private** GoogleMap **mMap**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_maps***);  
 *// 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**);  
 }  
  
  
 */\*\*  
 \* 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(**"My First Marker"**));  
 **mMap**.moveCamera(CameraUpdateFactory.*newLatLng*(sydney));  
 }  
}

**google\_map\_api.xml**

<**resources**>  
 *<!--* ***TODO: Before you run your application, you need a Google Maps API key.*** *To get one, follow this link, follow the directions and press "Create" at the end:  
  
 https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&keyType=CLIENT\_SIDE\_ANDROID&r=DD:95:04:55:F3:83:5D:34:2A:CD:47:11:9C:85:BC:8A:A9:FC:04:7A%3Bcom.example.waqas03312033721.googlemapapp  
  
 You can also add your credentials to an existing key, using this line:  
 DD:95:04:55:F3:83:5D:34:2A:CD:47:11:9C:85:BC:8A:A9:FC:04:7A;com.example.waqas03312033721.googlemapapp  
  
 Alternatively, follow the directions here:  
 https://developers.google.com/maps/documentation/android/start#get-key  
  
 Once you have your key (it starts with "AIza"), replace the "google\_maps\_key"  
 string in this file.  
 -->* <**string name="google\_maps\_key" templateMergeStrategy="preserve" translatable="false"**>  
 AIzaS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

</**string**>  
</**resources**>

**ManiFest**

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.waqas03312033721.googlemapapp"**>  
  
 <**uses-permission android:name="android.permission.INTERNET"**/>  
 <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"**/>  
 <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
 <**uses-permission android:name="com.google.android.providers.gsf.permission.Read\_GSERVICES"**/>  
 *<!--  
 The ACCESS\_COARSE/FINE\_LOCATION permissions are not required to use  
 Google Maps Android API v2, but you must specify either coarse or fine  
 location permissions for the 'MyLocation' functionality.  
 -->* <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
 <**uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
  
 *<!--  
 The API key for Google Maps-based APIs is defined as a string resource.  
 (See the file "res/values/google\_maps\_api.xml").  
 Note that the API key is linked to the encryption key used to sign the APK.  
 You need a different API key for each encryption key, including the release key that is used to  
 sign the APK for publishing.  
 You can define the keys for the debug and release targets in src/debug/ and src/release/.   
 -->* <**meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="@string/google\_maps\_key"** />  
  
 <**activity  
 android:name=".MapsActivity"  
 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**>