

- **Create a new project**
 - **File -> New -> Android Application Project**
 - You can leave all the options with their default value
 - Choose BlankActivity in the Create Activity screen
- **Ensure you have the latest version (v2) of the Google Play Services installed.**
 1. Select Window > Android SDK Manager or run android at the command line.
 2. Scroll to the bottom of the package list and select Extras > Google Play services.
The Google Play services SDK is downloaded to your computer and installed in your Android SDK environment at
`<android-sdk-folder>/extras/google/google_play_services/`

- **Add the Google Play Services project into your Eclipse workspace.**
 - a. Click File -> Import..., select Android -> Existing Android Code into Workspace
 - b. Browse to and select

```
<android-sdk-folder>/extras/google/google_play_services/libproject/  
google-play-services_lib
```

- **To add the dependency to Google Play Services into your project.**
 - **Project -> Properties -> Android -> Library, Add -> google-play-services_lib.**
If Google Play Services is not listed in libraries than go to google play services
project -> Properties -> Android -> check is Library option.

- **Update MainActivity.java to extend FragmentActivity instead of Activity.**

- **Check that Google Play Services is available on the device at runtime to protect against error cases:**

```
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplication  
Context());
```

- **Update res/layout/activity_main.xml and replace the entire contents with :**

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

- **To use Google Maps API you need an API Key get it As follows and replace 'Your Google Map API_KEY with' in the manifest file with this:**

- Go to the [Google APIs Console](#)
- In Services enable 'Google Maps Android API v2' and agree to the terms
- In the left navigation bar, click ->API Access.
- Click Create New Android Key....
- In the resulting dialog, enter the SHA-1 fingerprint, then a semicolon, then your application's package name. For example:
- BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.android.mapexample
- If you have debug and production keys then add both in.
- To obtain the SHA1 of your debug certificate use keytool:
 - **keytool -list -alias androiddebugkey -keystore ~/.android/debug.keystore -storepass android -keypass android -v**
- For Windows computers the .android directory is in your user directory
- Your new API key will be a 40 character string like AlzbSyBaEqeUpYvlsbUe0OqTPn1-ABVMHVbdNxo.

- **Add the following tag into your AndroidManifest.xml just before the closing </application> tag**

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="your_api_key"/>
```

- **Your application now also needs the following permissions in the AndroidManifest.xml**

```
<permission
    android:name="your_package_name.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>
<uses-permission
    android:name="your_package_name.permission.MAPS_RECEIVE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
    android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
```

- **Optionally you can include one or both of these permissions to enable auto-location**

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

- **Maps v2 uses OpenGL so the following uses-feature is also required**

```
<uses-feature android:glEsVersion="0x00020000"
android:required="true"/>
```

- **Test your application, checking the logcat output for any errors.**
- **If it is working try out some of the other GoogleMap options:**

```
mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
mMap.addMarker(new MarkerOptions().position(new LatLng(0,
0)).title("Marker"));
mMap.setMyLocationEnabled(true);
```

- more details at <https://developers.google.com/maps/documentation/android/views>

Attribution Requirements. If you use the Google Maps Android API in your application, you must include the Google Play Services attribution text as part of a "Legal Notices" section in your application. Including legal notices as an independent menu item, or as part of an "About" menu item, is recommended. The attribution text is available by making a call to `GooglePlayServicesUtil.getOpenSourceSoftwareLicenseInfo`.

https://docs.google.com/document/pub?id=19nQzvKP-CVLd7_VrpwnHfl-AE9fjbJySowONZZtNHzw