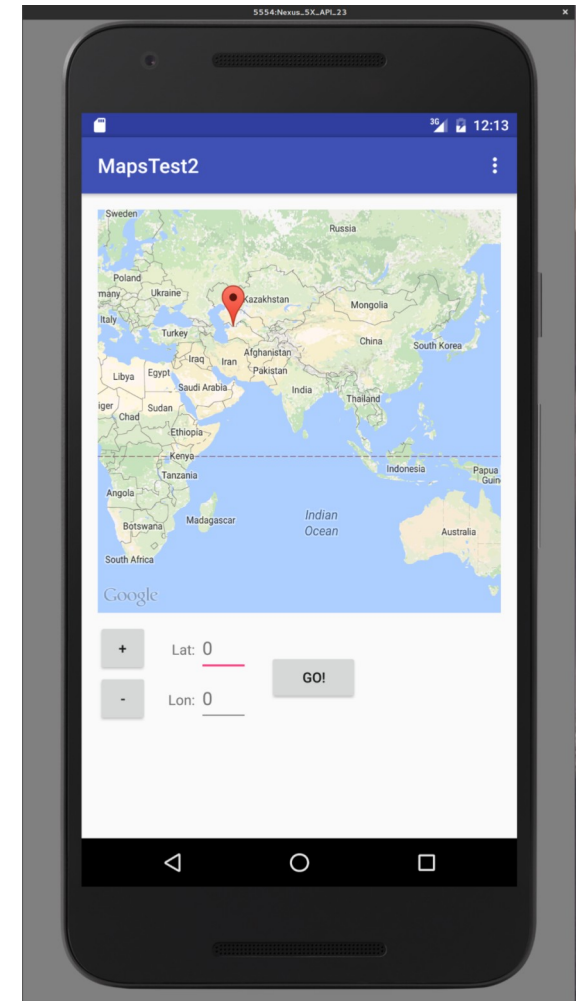
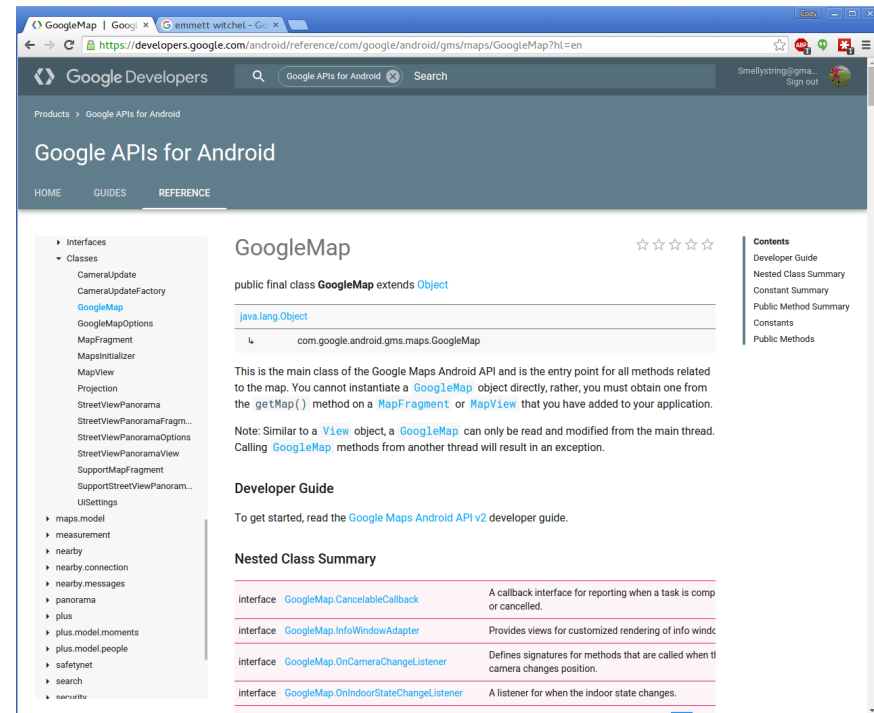


# The GoogleMap API



# The Official Documentation

<https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap?hl=en>



The screenshot shows the official Google Developers documentation for the `GoogleMap` class. The page is titled "GoogleMap" and is part of the "Google APIs for Android" reference. The main content area includes the following sections:

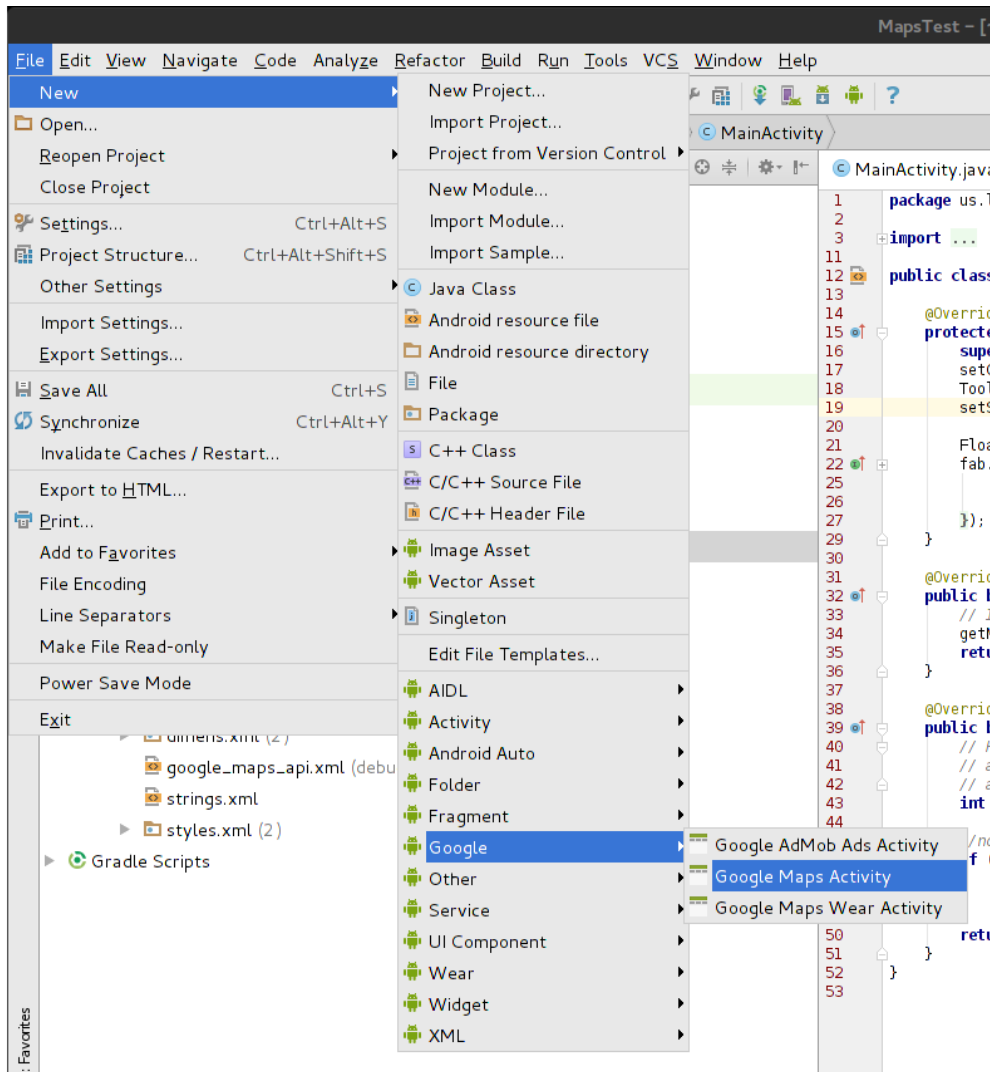
- Class Declaration:** `public final class GoogleMap extends Object`
- Package:** `com.google.android.gms.maps.GoogleMap`
- Description:** This is the main class of the Google Maps Android API and is the entry point for all methods related to the map. You cannot instantiate a `GoogleMap` object directly, rather, you must obtain one from the `getMap()` method on a `MapFragment` or `MapView` that you have added to your application.
- Note:** Similar to a `View` object, a `GoogleMap` can only be read and modified from the main thread. Calling `GoogleMap` methods from another thread will result in an exception.
- Developer Guide:** To get started, read the [Google Maps Android API v2 developer guide](#).
- Nested Class Summary:** A table listing several interfaces provided by the class.

Interface	Description
<code>GoogleMap.CancelableCallback</code>	A callback interface for reporting when a task is completed or cancelled.
<code>GoogleMap.InfoWindowAdapter</code>	Provides views for customized rendering of info windows.
<code>GoogleMap.OnCameraChangeListener</code>	Defines signatures for methods that are called when the camera changes position.
<code>GoogleMap.OnIndoorStateChangeListener</code>	A listener for when the indoor state changes.

# Step 1: Add Maps Activity

- This generates a lot of useful code and XML
- We aren't actually going to use the Activity
- You can skip this if you are ok with generating this XML by yourself

# Step 1: Add Maps Activity



- File >
- New >
- Google Maps Activity

# Get an API key

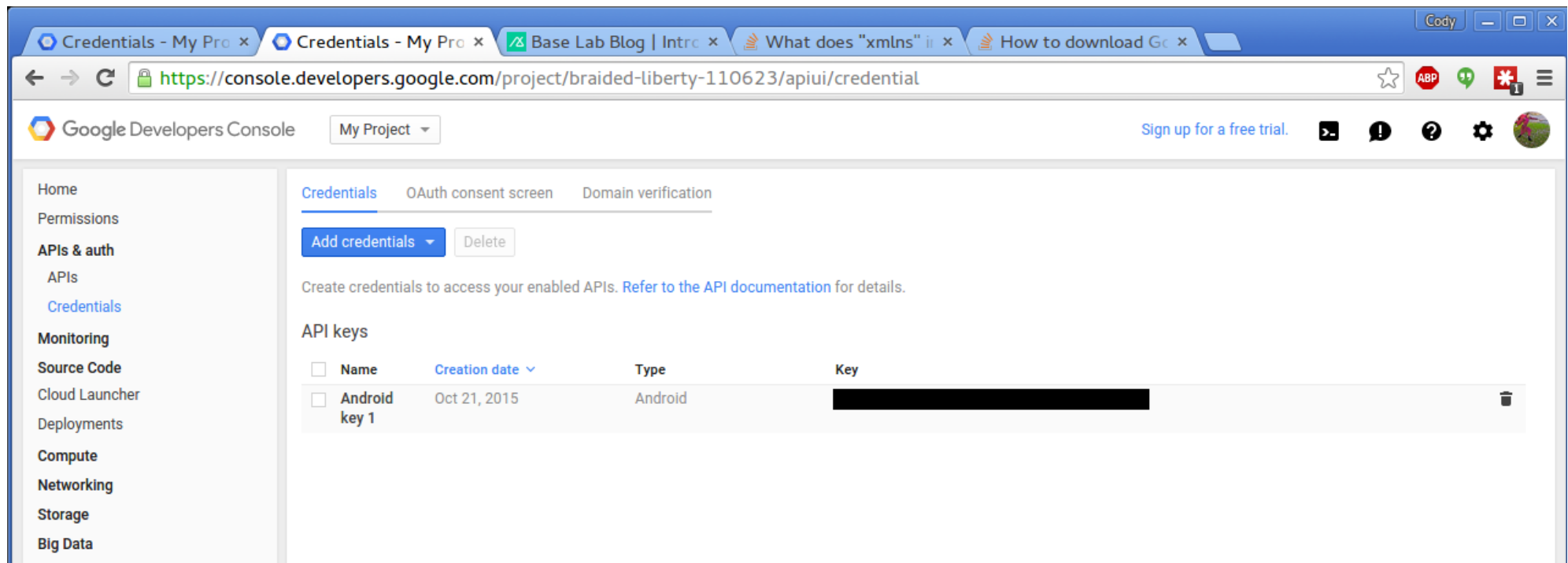
- Not required to use the GoogleMap API
  - But helpful
- You need a new one for every app you write
- It is easy and free

# Get an API key


- Step 1: open the file  
values/google\_maps\_api.xml
- Follow the link
- Request an API key
- Copy and paste the key back into the XML file

# Get an API key

- You can see your previous keys at <https://console.developers.google.com>



The screenshot shows the Google Developers Console interface. The browser address bar displays the URL <https://console.developers.google.com/project/braided-liberty-110623/apiui/credential>. The page title is "Credentials - My Project". The main content area shows the "Credentials" tab selected, with sub-tabs for "OAuth consent screen" and "Domain verification". Below the tabs, there are buttons for "Add credentials" and "Delete". A message states: "Create credentials to access your enabled APIs. Refer to the API documentation for details." Under the heading "API keys", there is a table with the following data:

<input type="checkbox"/>	Name	Creation date	Type	Key	
<input type="checkbox"/>	Android key 1	Oct 21, 2015	Android	[REDACTED]	

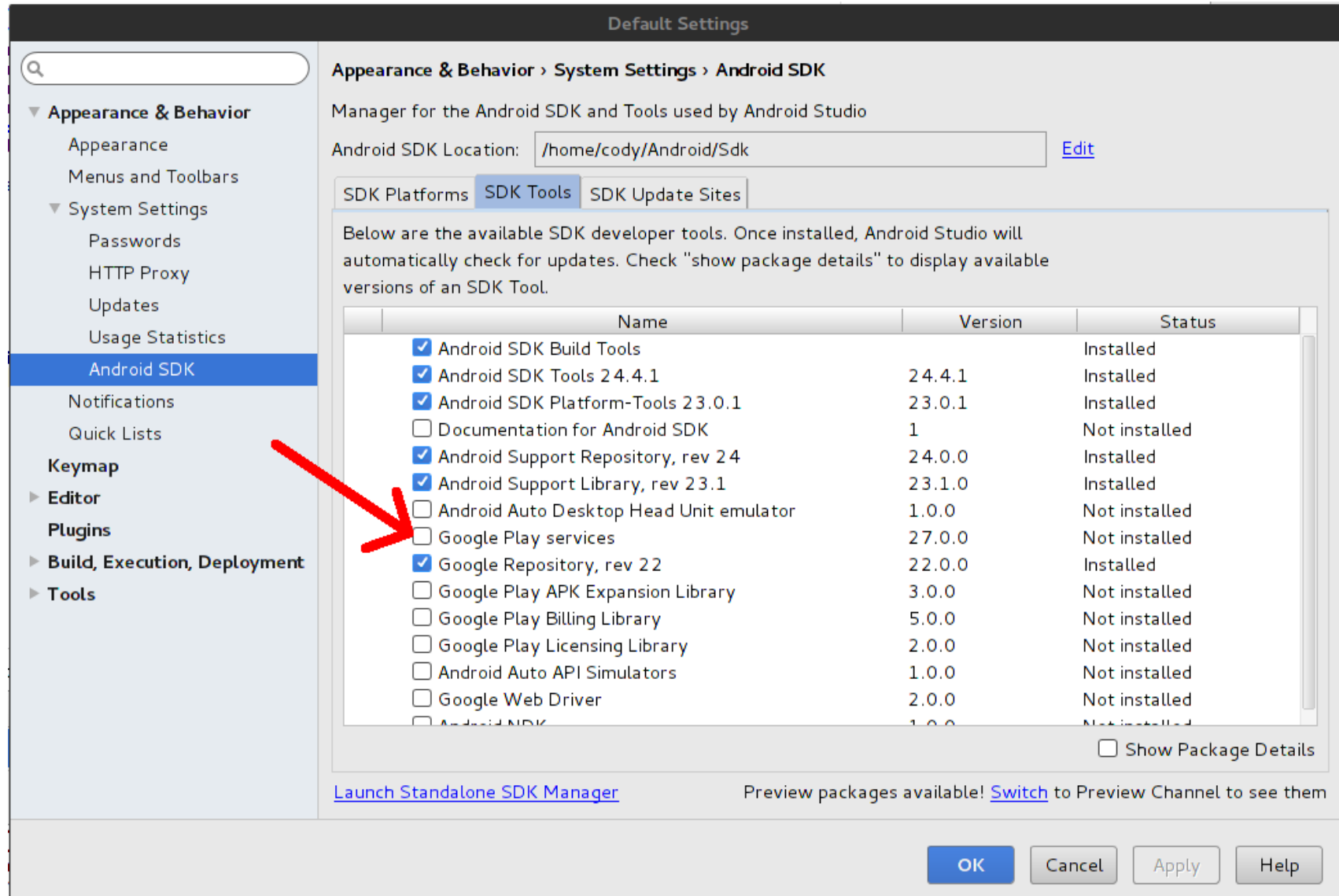
# Set up the Google Play Services SDK

Step 1: download the package

SDK Manager > SDK Tools > Google Play Services



# Set up the Google Play Services SDK



Default Settings

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location:  [Edit](#)

SDK Platforms | **SDK Tools** | SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
<input checked="" type="checkbox"/>	Android SDK Build Tools		Installed
<input checked="" type="checkbox"/>	Android SDK Tools 24.4.1	24.4.1	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-Tools 23.0.1	23.0.1	Installed
<input type="checkbox"/>	Documentation for Android SDK	1	Not installed
<input checked="" type="checkbox"/>	Android Support Repository, rev 24	24.0.0	Installed
<input checked="" type="checkbox"/>	Android Support Library, rev 23.1	23.1.0	Installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.0.0	Not installed
<input type="checkbox"/>	Google Play services	27.0.0	Not installed
<input checked="" type="checkbox"/>	Google Repository, rev 22	22.0.0	Installed
<input type="checkbox"/>	Google Play APK Expansion Library	3.0.0	Not installed
<input type="checkbox"/>	Google Play Billing Library	5.0.0	Not installed
<input type="checkbox"/>	Google Play Licensing Library	2.0.0	Not installed
<input type="checkbox"/>	Android Auto API Simulators	1.0.0	Not installed
<input type="checkbox"/>	Google Web Driver	2.0.0	Not installed
<input type="checkbox"/>	Android NDK	1.0.0	Not installed

Show Package Details

[Launch Standalone SDK Manager](#) Preview packages available! [Switch](#) to Preview Channel to see them

OK Cancel Apply Help

# Set up the Google Play Services SDK

Step 2: Build an emulator with the SDK

ADV > Create Virtual Device

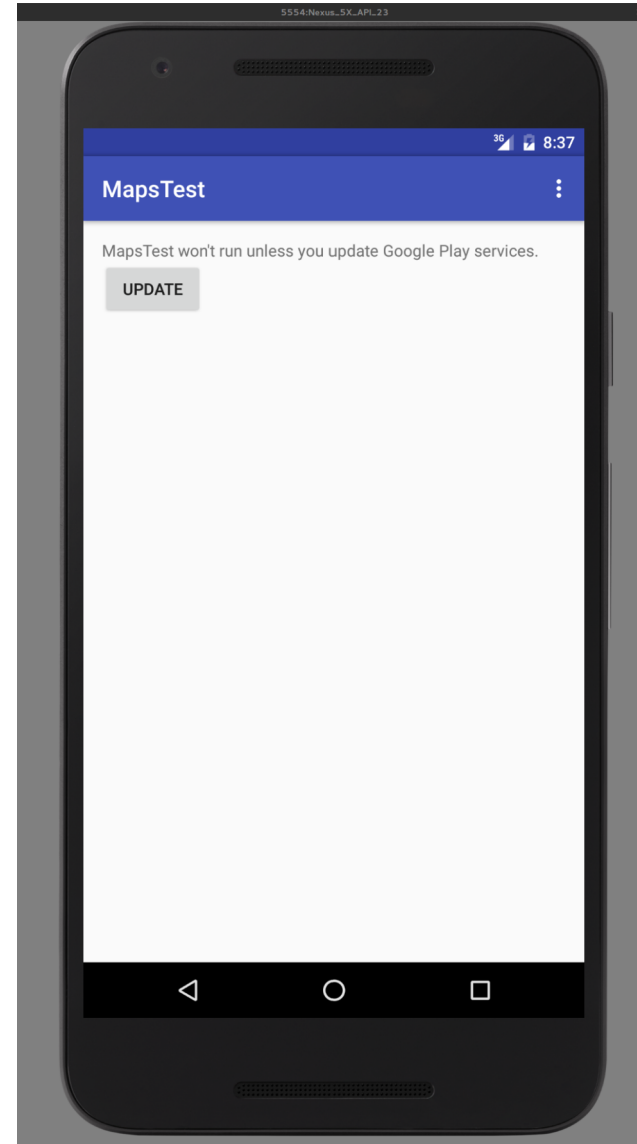
You may need to use an image from Google

# Set up the Google Play Services SDK

If you see this error:

“MapsTest won't run unless you update Google Play services”

Tell gradle to expect a lower version



# Set up the Google Play Services SDK

In build.gradle (Module: app)

Lower the version until it works. “7.+” works for me.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.0'
    compile 'com.android.support:design:23.1.0'
    compile 'com.google.android.gms:play-services:7.+'
    //compile 'com.google.android.gms:play-services:8.1.0'
}
```

# Fix some permissions

In the Android manifest add the following:

```
] <!--  
    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" />  
  
<!--  
    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" />
```