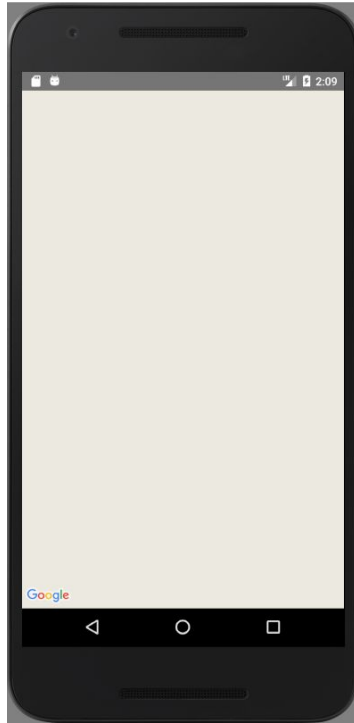


Google Maps Troubleshooting



Before you go through the troubleshooting guide below, make sure that you've consulted the class [FAQ](#), Google's [Map Activity Tutorial](#), as well as these helpful resources from our professors:

- General Google Maps Information:
<https://faculty.cs.byu.edu/~barker/cs240/notes/22a-maps/>
- Basic Map Fragment Example:
https://students.cs.byu.edu/~cs240ta/fall2018/jones_files/?path=MapFragment2/

If you've consulted the resources above and are still having trouble with your map, run through the following troubleshooting steps to resolve the issue.

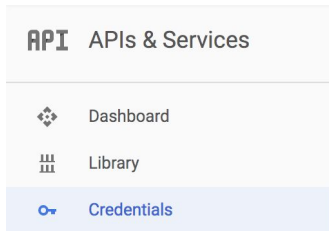
Maps Troubleshooting Solutions

The following are listed roughly in order corresponding to how easy the solution is plus how commonly it fixes the issue.

- 1) [Make sure your google maps API key is valid](#)
- 2) [Disable restrictions on maps API key](#)
- 3) [Create a new API key](#)
- 4) [Ensure google maps android api v2 is enabled on the google developer console](#)
- 5) [Add locations permissions to manifest](#)
- 6) [Downgrade Google Play services to 11.0.4](#)
- 7) [If SDK version is below 27, Upgrade SDK to version 27 or higher](#)
- 8) [Override Fragment lifecycle methods in fragment and call them on your SupportMapFragment/MapView](#)
- 9) [Use a different emulator \(Nexus 5/6X with Google Play\)](#)
- 10) [Modify Android Manifest for Android 28](#)
- 11) [Run on a real device](#)
- 12) [Sling some Google-fu](#)

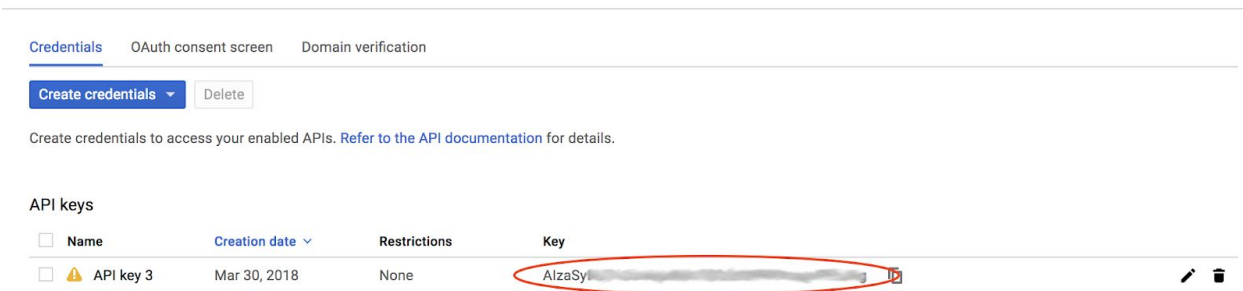
Make sure your google maps API key is valid

- Visit <https://console.developers.google.com>
- Select the “credentials” tab on the left side



- Ensure the API key listed matches what you're using in your project

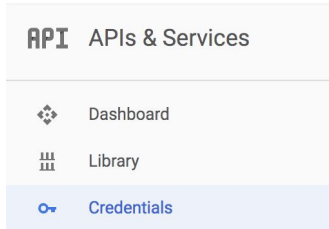
Credentials

A screenshot of the Google Cloud Platform "Credentials" page. The page has a header with "Credentials", "OAuth consent screen", and "Domain verification". Below the header is a "Create credentials" button and a "Delete" button. A message says "Create credentials to access your enabled APIs. Refer to the API documentation for details." Below this is a table titled "API keys". The table has columns for "Name", "Creation date", "Restrictions", and "Key". There is one row in the table with a warning icon, "API key 3", "Mar 30, 2018", "None", and a key value "AlzaSy..." which is circled in red. There are also edit and delete icons for this key.

<input type="checkbox"/>	Name	Creation date	Restrictions	Key
<input type="checkbox"/>	⚠ API key 3	Mar 30, 2018	None	AlzaSy...

Disable restrictions on maps API key

- Visit <https://console.developers.google.com>
- Select the “credentials” tab on the left side



- Click on the “Edit” icon for your API key

API keys

<input type="checkbox"/>	Name	Creation date	Restrictions	Key	
<input type="checkbox"/>	⚠ API key 3	Mar 30, 2018	None	AlzaSy...	

- On the edit page, set the application restrictions to “none”

Key restrictions

This key is unrestricted. To prevent unauthorized use and quota theft, restrict your key. [Learn more](#)

⚠ Application restrictions: None ⚠ API restrictions: None

Application restrictions

Application restrictions specify which web sites, IP addresses, or apps can use this key. You can set one restriction type per key.

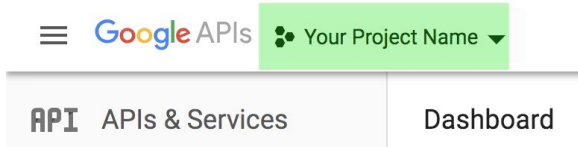
Application restrictions

- None
- HTTP referrers (web sites)
- IP addresses (web servers, cron jobs, etc.)
- Android apps
- iOS apps

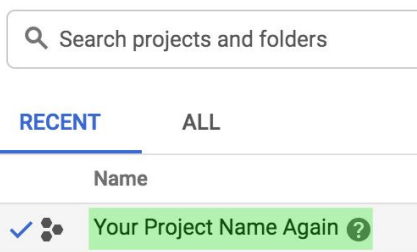
- Finally, hit “save” at the bottom of the edit page

Create a new API key

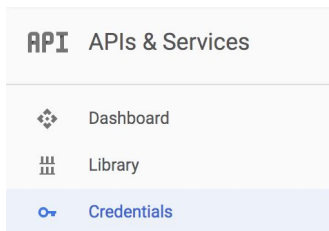
- Visit <https://console.developers.google.com>
- Click the dropdown at the top left of the screen.



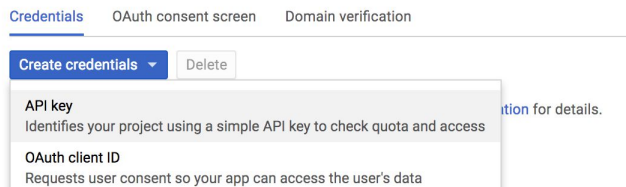
- In the resulting window, re-select your current project
- Select a project**



- Next, select the “credentials” tab on the left side



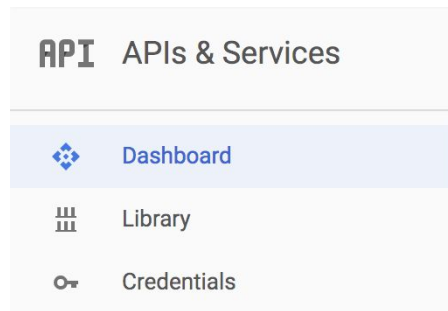
- Select the blue button that says “Create Credentials”, and then select “API key” from the dropdown menu



- Copy the generated API key and use it in your Android Studio Project

Ensure Google Maps Android API v2 is enabled

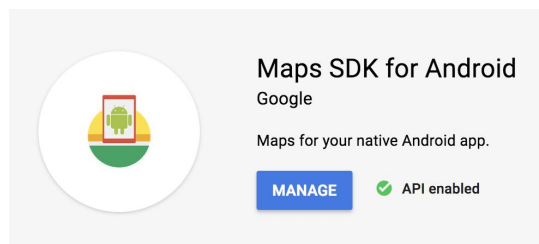
- Visit <https://console.developers.google.com>
- Select the “dashboard” tab on the left side



- Click on “Enable APIs and Services” at the top of the page

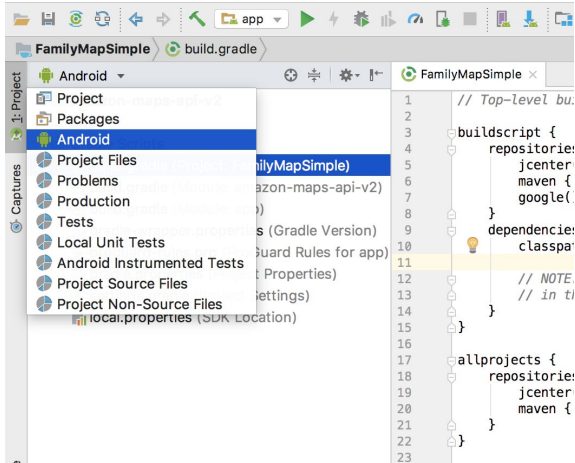
The screenshot shows the 'Enabled APIs and services' dashboard. At the top, there is a 'Dashboard' tab and a highlighted '+ ENABLE APIS AND SERVICES' button. Below this, the text reads 'Enabled APIs and services' and 'Some APIs and services are enabled automatically'. There is a filter for 'Activity for the last hour' with options: 1 hour, 6 hours, 12 hours, 1 day, 2 days, 4 days, 7 days, 14 days, 30 days. Three charts are displayed: 'Traffic' (Requests/sec), 'Errors' (Percent of requests), and 'Median latency' (Milliseconds). All three charts show 'There is no traffic/errors/latency data for this time period.' Below the charts is a table with columns: API, Requests, Errors, Error ratio, Latency, median, Latency, 98%, and a 'Disable' button. The table contains one entry: 'BigQuery API' with dashes in all other columns.

- Enter “Maps SDK for Android” into the search bar, and select the Android Maps SDK
- If your Maps API is enabled, you will see a header like this ---->
- Otherwise, your Maps is disabled. Click the “Enable” button to activate the API.



Add locations permissions to manifest

- First, make sure your project view is in the “Android” mode. Change this by selecting the corresponding option in the drop down menu just above the “project” pane.



- Next, open “app” -> “manifests” -> “androidmanifest.xml”



- Add the following permissions to your android manifest, just above the start of the “<application” tag:

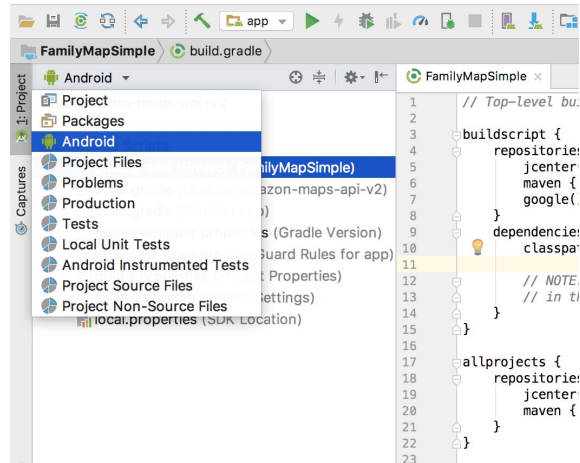
```

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

```

Downgrade Google Play services to 11.0.4

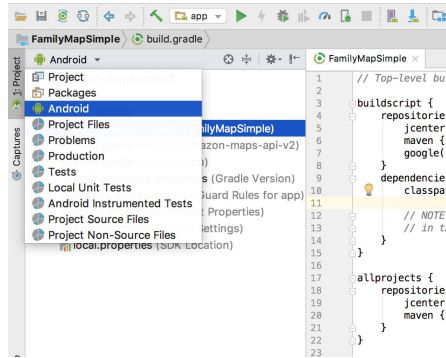
- First, make sure your project view is in the “Android” mode. Change this by selecting the corresponding option in the drop down menu just above the “project” pane.



- Next, open “Gradle Scripts” -> “build.gradle (Module: app)”
- Find the line that reads:
implementation 'com.google.android.gms:play-services-maps:Ver.Num.Here'
- Change the version number to 11.0.4, so the line reads
implementation 'com.google.android.gms:play-services-maps:11.0.4'

If below SDK 27, Upgrade Project + Emulator SDK to version 27 or higher

- To check your APK version, check in the “build.gradle” file of the App module.
- Make sure your project view is in the “Android” mode. Change this by selecting the corresponding option in the drop down menu just above the “project” pane.



- Next, open “Gradle Scripts” -> “build.gradle (Module: app)”
 - Find the options that say “compileSdkVersion” and “targetSdkVersion”, they should have the same number listed for both.
 - If this number is lower than 27, raise it so your SDK version is 27.
 - Clean your project by selecting “Build” -> “Clean Project” from the application menu.
-
- You should also change your emulator to match your SDK version if applicable. To do this, follow the steps listed for: [Using a Different Emulator](#)

Override Fragment lifecycle methods and call them on your SupportMapFragment/MapView

**In the MapFragment you created, first make sure your SupportMapFragment or MapView widget is a class variable, that way we can access it throughout the file.

- In the MapFragment you created (e.g. MapFragment.java), override the following methods:
 - onStart
 - onStop
 - onPause
 - onDestroy
 - onResume
 - onLowMemory

(An easy way to override methods in android studio is simply to start typing the name of the method, and let your IDE's autocomplete generate the method stub.)

- You should now have several method bodies that look like this:

```
@Override
public void lifeCycleMethod() {
    super.lifeCycleMethod();
}
```

- Inside each of your new methods, call the corresponding method on your SupportMapFragment or MapView widget. You may want to add a check for null to protect against exceptions.

- Your method bodies should now look like this:

```
@Override
public void lifeCycleMethod() {
    super.lifeCycleMethod();
    if (mSupportMapFrag_or_MapView != null) {
        mSupportMapFrag_or_MapView.lifeCycleMethod();
    }
}
```

- Clean your project by selecting "Build" -> "Clean Project" from the application menu.

Use a different emulator (Nexus 5/6X with Google Play)

- Open the AVD Manager by selecting “Tools” -> “AVD Manager” from the application menu
- This will list your currently created emulators. If your emulator API level is already above 27 and you are using a Nexus 5/6X with google play, you are ok.

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
[Icon]	Nexus 5 API 27 Google Play	[Play Store Icon]	1080 x 1920: xxhdpi	27	Android 8.1 (Google Play)	x86	4.5 GB	[Play] [Edit] [Dropdown Arrow]

- If your API level is below 27, select “Create Virtual Device” at the bottom of this window.
- Select “Nexus 5X” as your target device, and click “Next”
- Select a system image from the options. 27 or 28 will do. (You may need to download the image first)

Select a system image

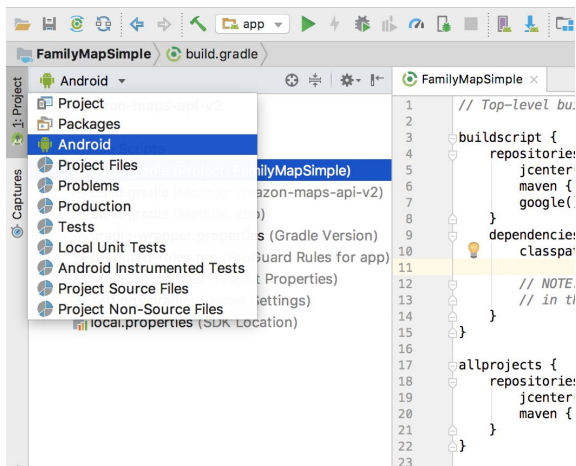
Release Name	API Level	ABI	Target
Pie Download	28	x86	Android 9.0 (Google Play)
Oreo	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

- Select “Next”, then “Finish” to create your emulator

Modify Android Manifest for Android 28

**The following maps fix is only applicable for SDK versions 28 and above. Your SDK version can be found inside your App module's "build.gradle" file.

- First, make sure your project view is in the "Android" mode. Change this by selecting the corresponding option in the drop down menu just above the "project" pane.



- Next, open "app" -> "manifests" -> "androidmanifest.xml"



- Add the following flag to your manifest, inside the "<application" tag (i.e. where your activities are declared):

```
<uses-library android:name = "org.apache.http.legacy" android:required = "false" />
```

Run on a real device

Sometimes Map problems are caused by emulator issues that vary from person to person. In these cases running your app on a real device is a great option. For more information on how to run your app on a real device, consult “*Chapter 2: Running on a Device*” in the BigNerdRanch Android book, found here:

<https://www.oreilly.com/library/view/android-programming-the/9780134706061/>

Google is your best friend

If all else fails and your map is still giving you trouble, don't worry - you're not alone. Many others have ran into issues with the google map across the android development community, so there is a lot of discussion on the topic online. Take your own project's situation and symptoms into mind and use your google skills to track down a solution!