The process of setting up the starter code is slightly more complicated in this course than for previous courses because you're actually becoming a Google Maps API developer! But follow these steps and you'll be ready to develop your own Maps application in no time. If you get stuck, post in the forum for help.

In Part 1 of this guide you will download the starter code and load it into Eclipse. This is technically all you need to do to complete all the assignments for this course, but if you only follow Part 1, the MapApp (Google Maps style application) might not run.

If you get a "Something Went Wrong" error when you run the Map App, then come back and follow Part 2. In Part 2 we walk you through the process of getting your own Maps API developer key so you can develop and run the Map App. This usually fixes the issues, but is only needed if you are getting the Something Went Wrong error.

Part 1: Download the starter code and load it into Eclipse

This course has been running since fall of 2015. As such the starter code is pretty stable and we do not anticipate finding major problems with it. In the event that we find and fix a major issue, we will post instructions about how to fix the problem in the discussion forums.

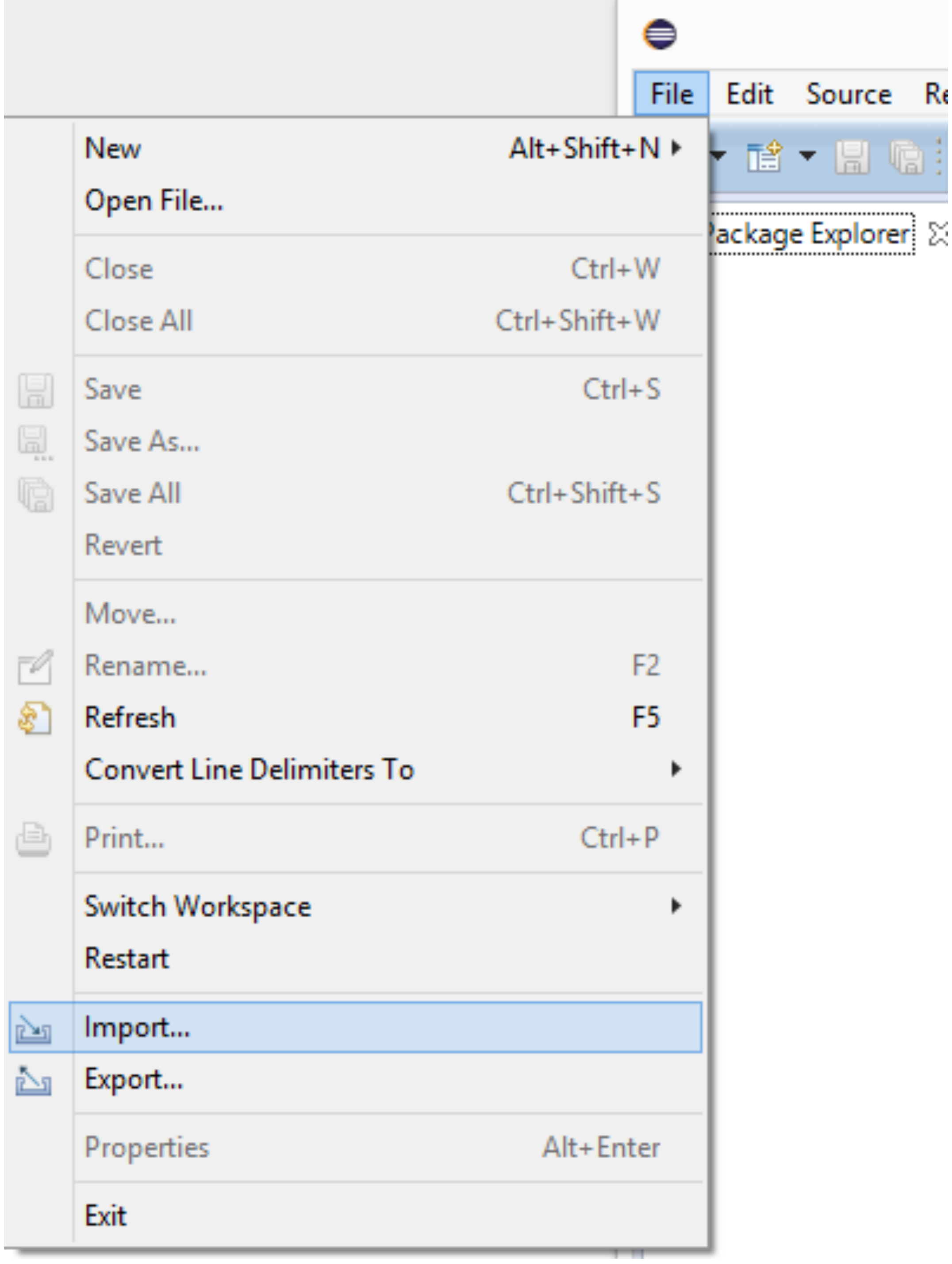
1. Download the starter code for this course here:

Course3StarterCode.zip

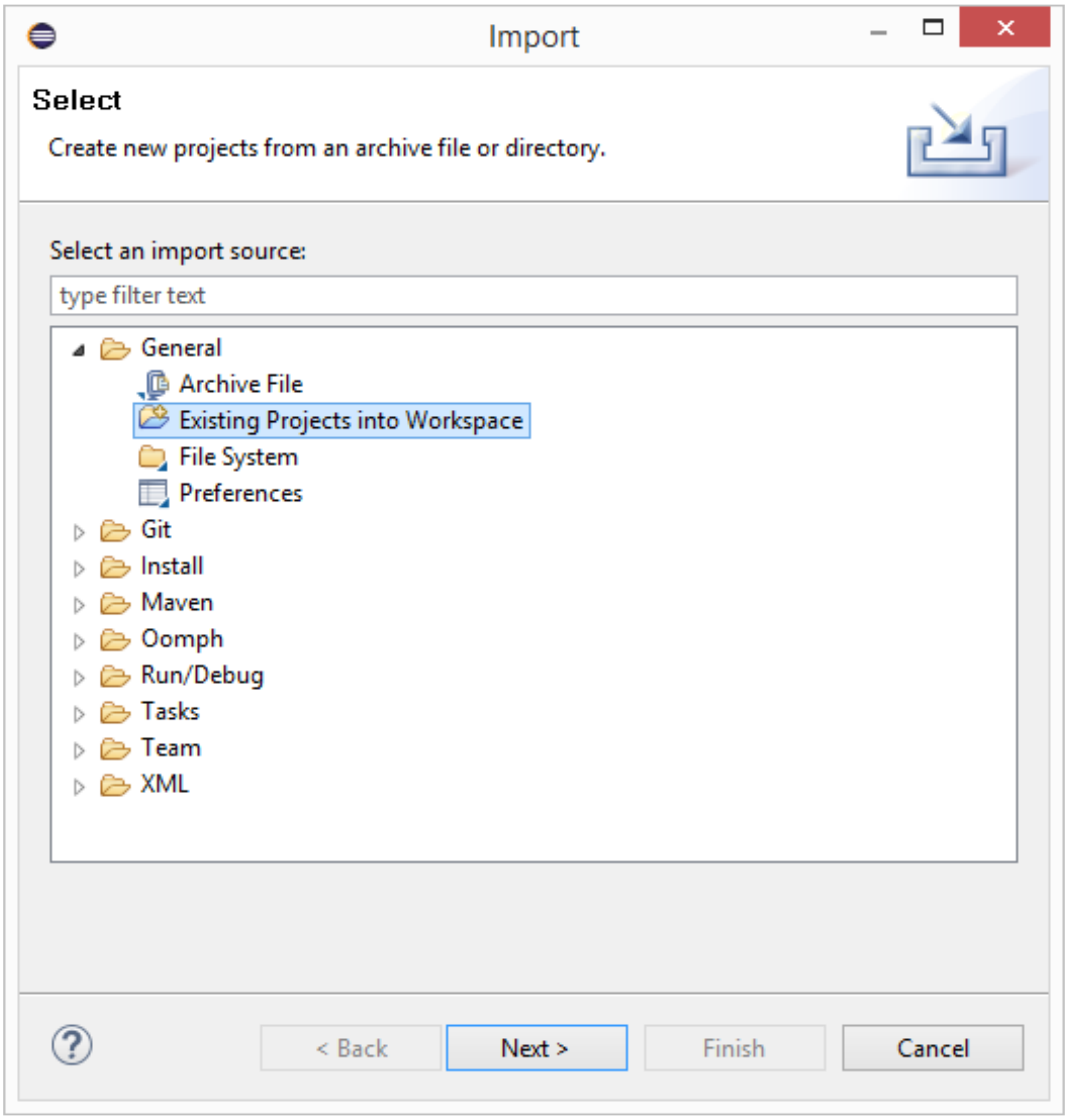
2. Extract the zip file somewhere outside of your Eclipse workspace folder.

3. Open Eclipse if it is not already open.

4. From the File menu in Eclipse select Import...



5. In the Import window expand General and select Existing Projects into Workspace. Then click Next >



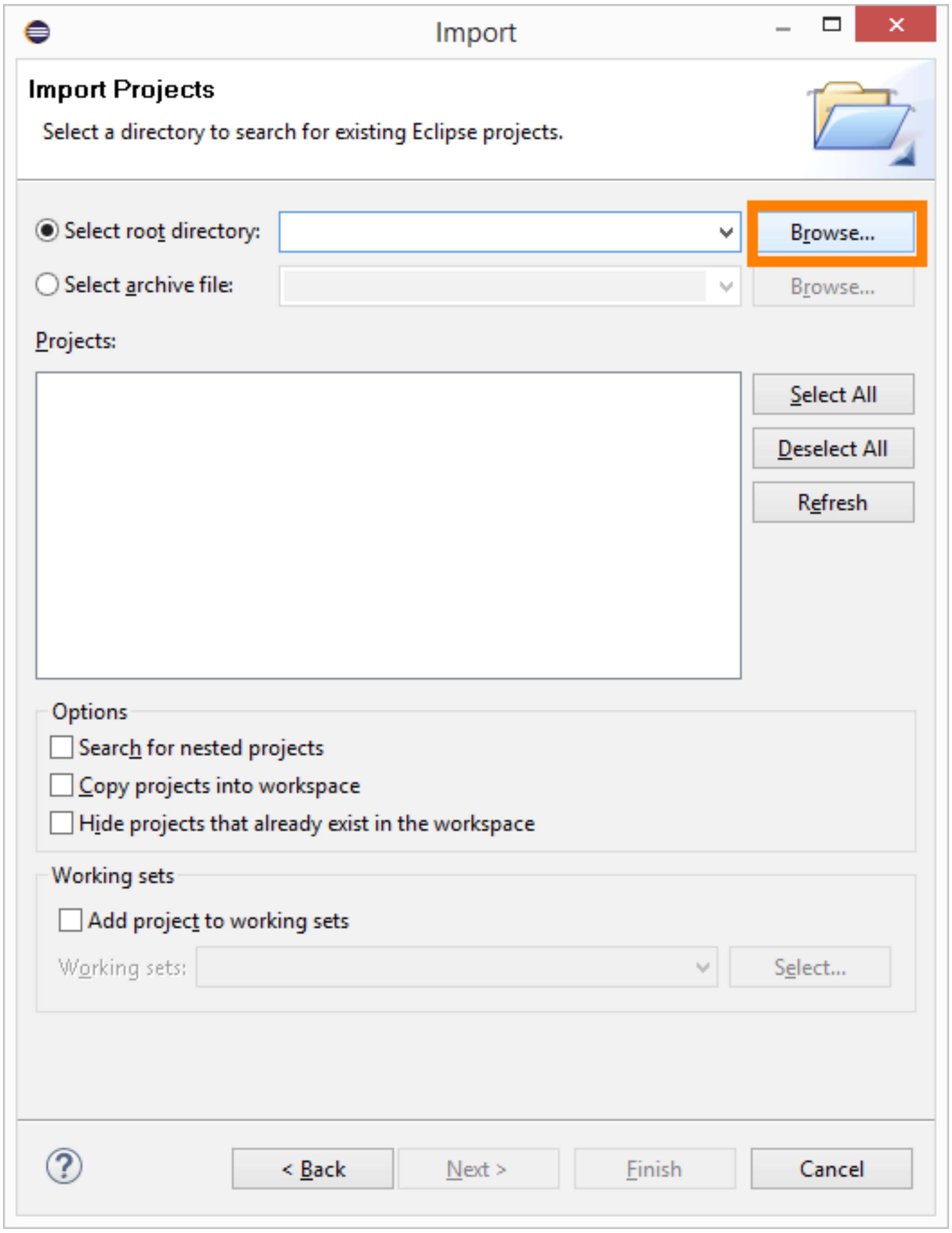
Import

Select

Create new projects from an archive file or directory.

Select an import source:

6. In the window that appears, click Browse next to "Select root directory" (keep this option selected).



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

Options

□ Search for nested projects

□ Copy projects into workspace

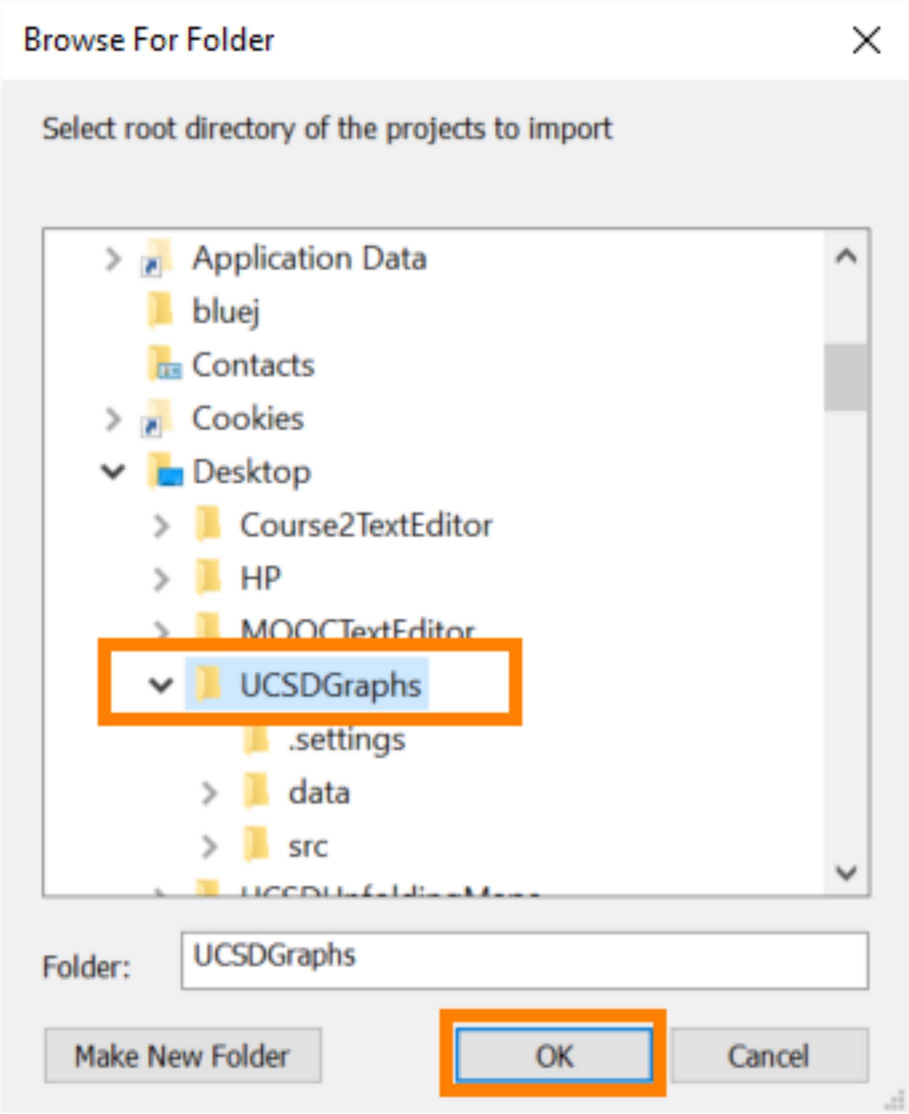
□ Hide projects that already exist in the workspace

Working sets

□ Add project to working sets

Working sets:

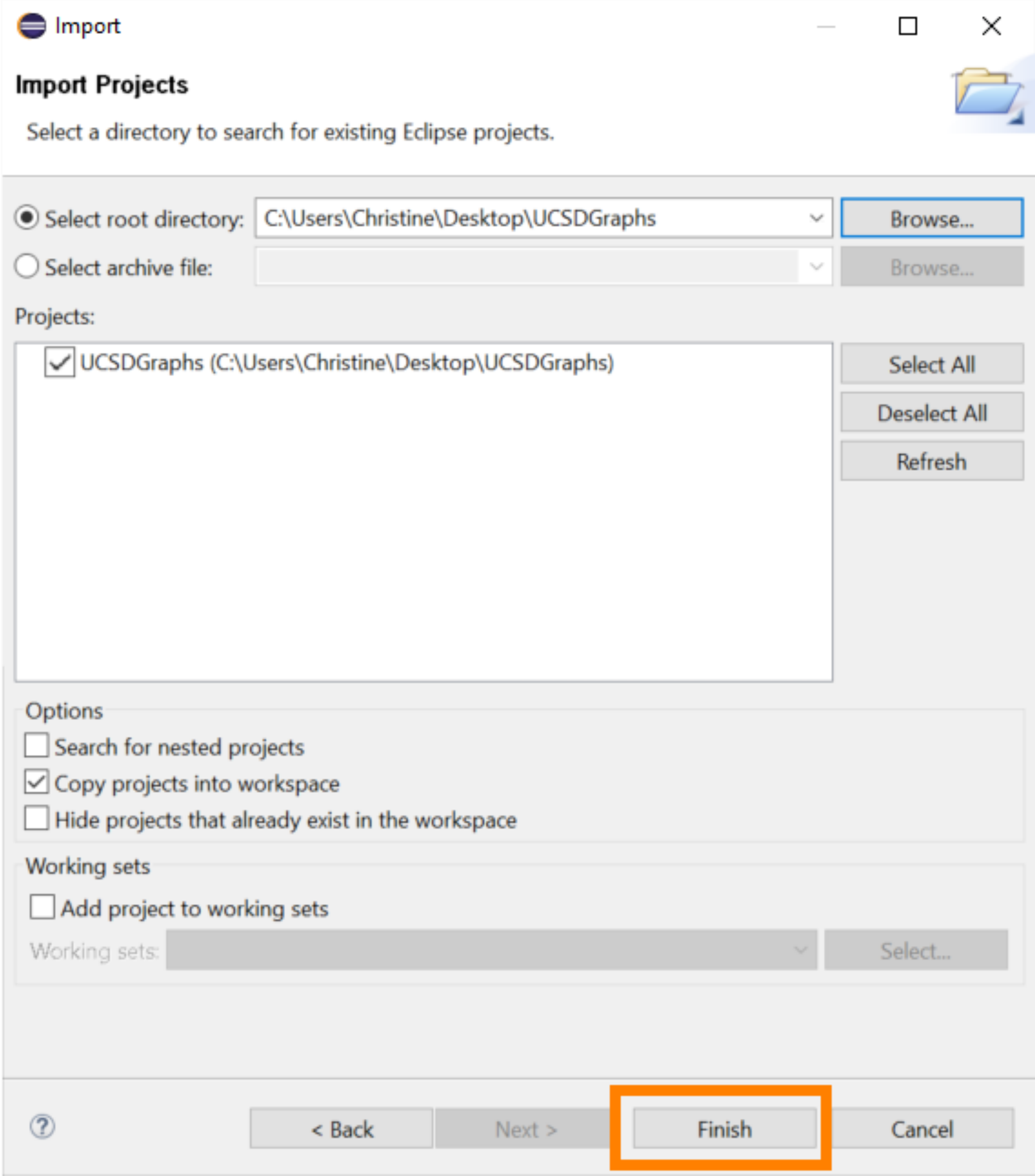
7. In the File Chooser, browse to where you uncompressed the starter code. It should be in a folder named UCSDGraphs. Select this folder (NOT one of the subfolders) and click OK.



Select root directory of the projects to import

Folder: UCSDGraphs

8. You should see the UCSDGraphs project selected under Projects: Make sure the option to "Copy projects into workspace" is checked, then click Finish.



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Users\Christine\Desktop\UCSDGraphs

○ Select archive file:

Projects:

☑ UCSDGraphs (C:\Users\Christine\Desktop\UCSDGraphs)

Options

☐ Search for nested projects

☑ Copy projects into workspace

☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

9. Now, you should see the project in your Eclipse workspace. In the Package Explorer you will see the UCSDGraphs project. If you expand this project you will see several folders including src, data and some library folders. src is where all of your source (Java) code that we provide and that you will write is located. If you expand src you will see a number of "packages," which we will explain as you go through the programming assignments in the different modules. A package is just a collection of files that are organized into their own space so they won't interfere with other files of the same name in other packages. The instructions for the programming assignment in this module give you more information about how to compile and run your code.

It's expected to see some warnings (the tiny yellow exclamation points next to the items in the Package Explorer) and generally you can ignore them for now, though sometimes they do give you hints about how to improve your code. You can see what they are in the window at the bottom. However you should not see any errors (which would be displayed in red).

10. (Optional: If you plan to test with JUnit--not explicitly covered in this course) Add JUnit to your project. To do this, go to Project->Properties. Select "Java Build Path". Select the "Libraries" tab and "Add Library". Select JUnit, then JUnit 4.

11. Set up the workspace JRE. Right click on the UCSDGraphs project folder in the Package Explorer and select Build Path->Configure Build Path. Go on the Libraries tab and click on "Add Library". Select the "JRE System Library" and click Next. Select "Workspace default JRE" and click Finish. Then click OK.

12. Finally, notice that when you imported this project into Eclipse you got a COPY of all of the starter code files you downloaded into your workspace. You will find the actual files Eclipse is using (including the source (src) and data files) in a directory in your workspace folder called UCSDGraphs. Go find these files so you know where they are and what Eclipse is doing behind the scenes.

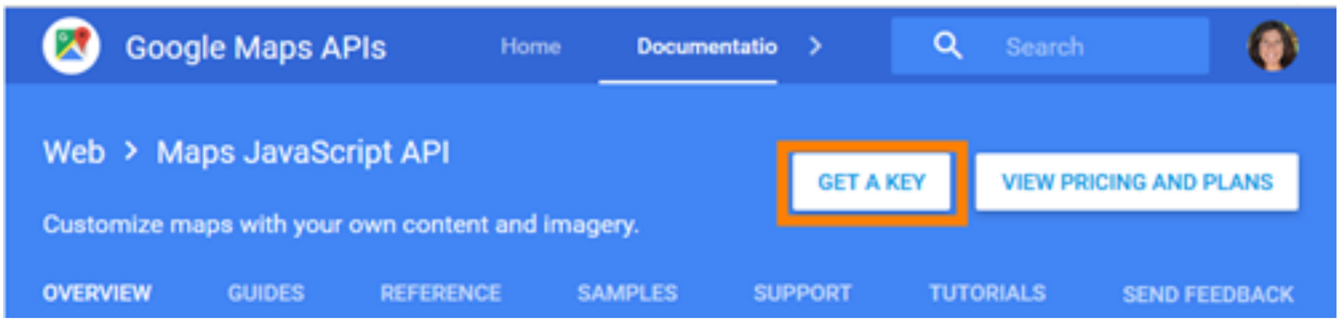
(If needed) Part 2: Obtain and use your Google Maps API Javascript Developer Key

Why do I need this key?

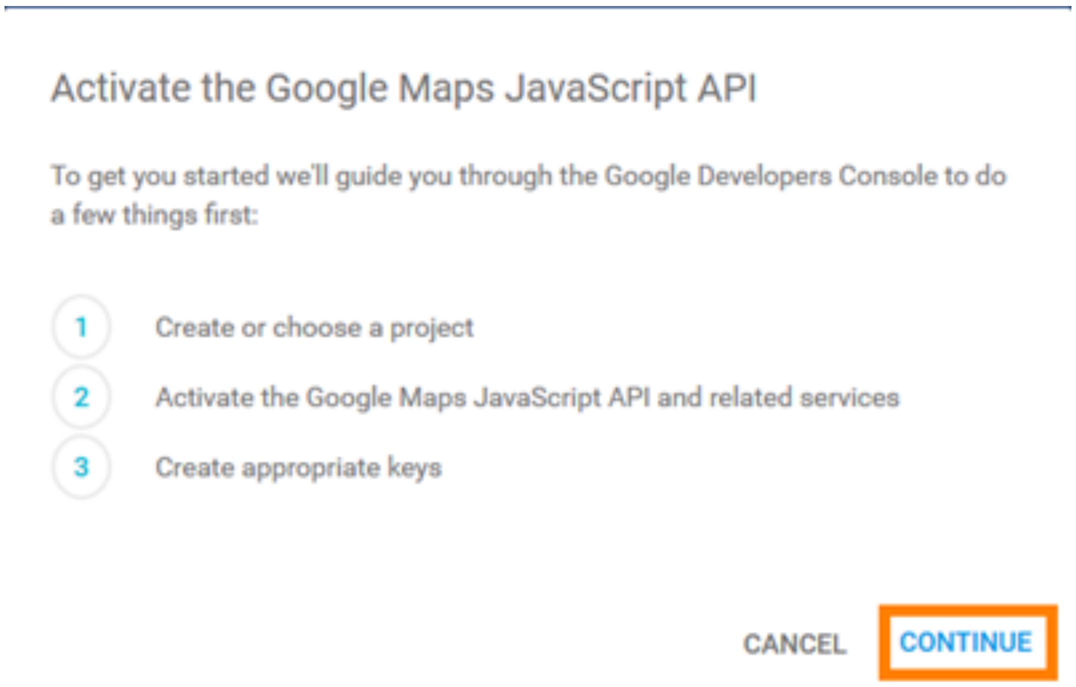
In this project you are developing an application that uses the Google Maps API. Pretty cool, huh? In order to do this, you need to get yourself a developer key. Google allows you to develop apps that are used by small numbers of people for free, but they don't want you making the next viral app without paying them for their part in your product. So Google uses your key to track how many calls are being made to their APIs. They will contact you to ask for money if you make too many calls, but don't worry. You're allow thousands of calls per day, so there's no way you'll hit this limit in this course!

1. Go to https://developers.google.com/maps/documentation/javascript/

2. Click on Get a Key at the top of the page



3. In the window the appears, click "Continue"



Activate the Google Maps JavaScript API

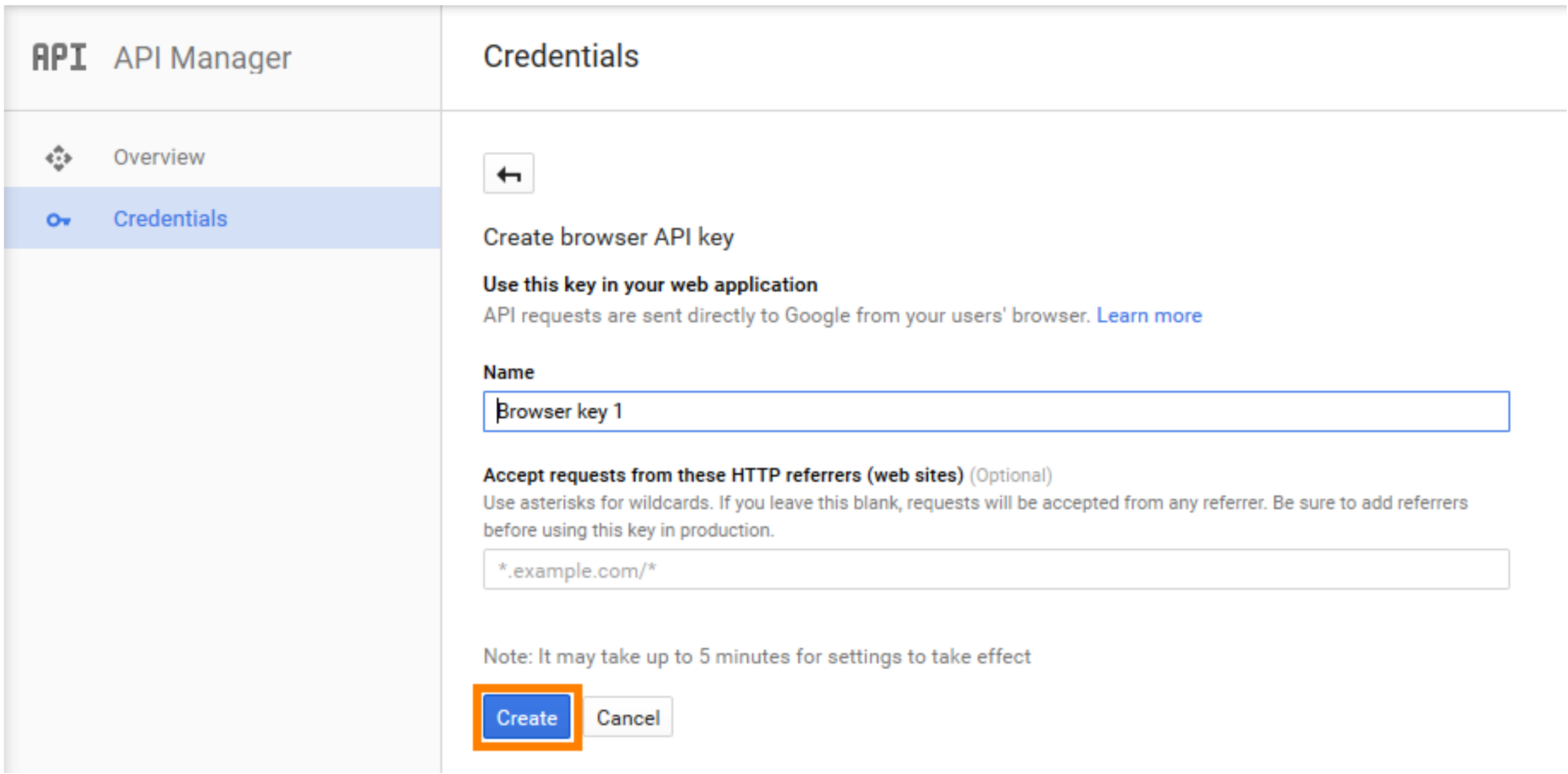
To get you started we'll guide you through the Google Developers Console to do a few things first:

1. Create or choose a project
2. Activate the Google Maps JavaScript API and related services
3. Create appropriate keys

(You might get a "Server Error" message. We sometimes do. If that happens just click OK, and continue and there should be no problem)

4. On the next screen make sure "Create a New Project" is selected (it should be by default) and click "Continue"

5. Next you will create your Browser Key. You can leave everything with its default values, or change the name of the key. You do not need to place any restrictions on your key, but make sure you DO NOT GIVE IT TO ANYONE ELSE. Click "Create" to create your key for the project you just created.



Credentials

Create browser API key

Use this key in your web application
API requests are sent directly to Google from your users' browser. Learn more

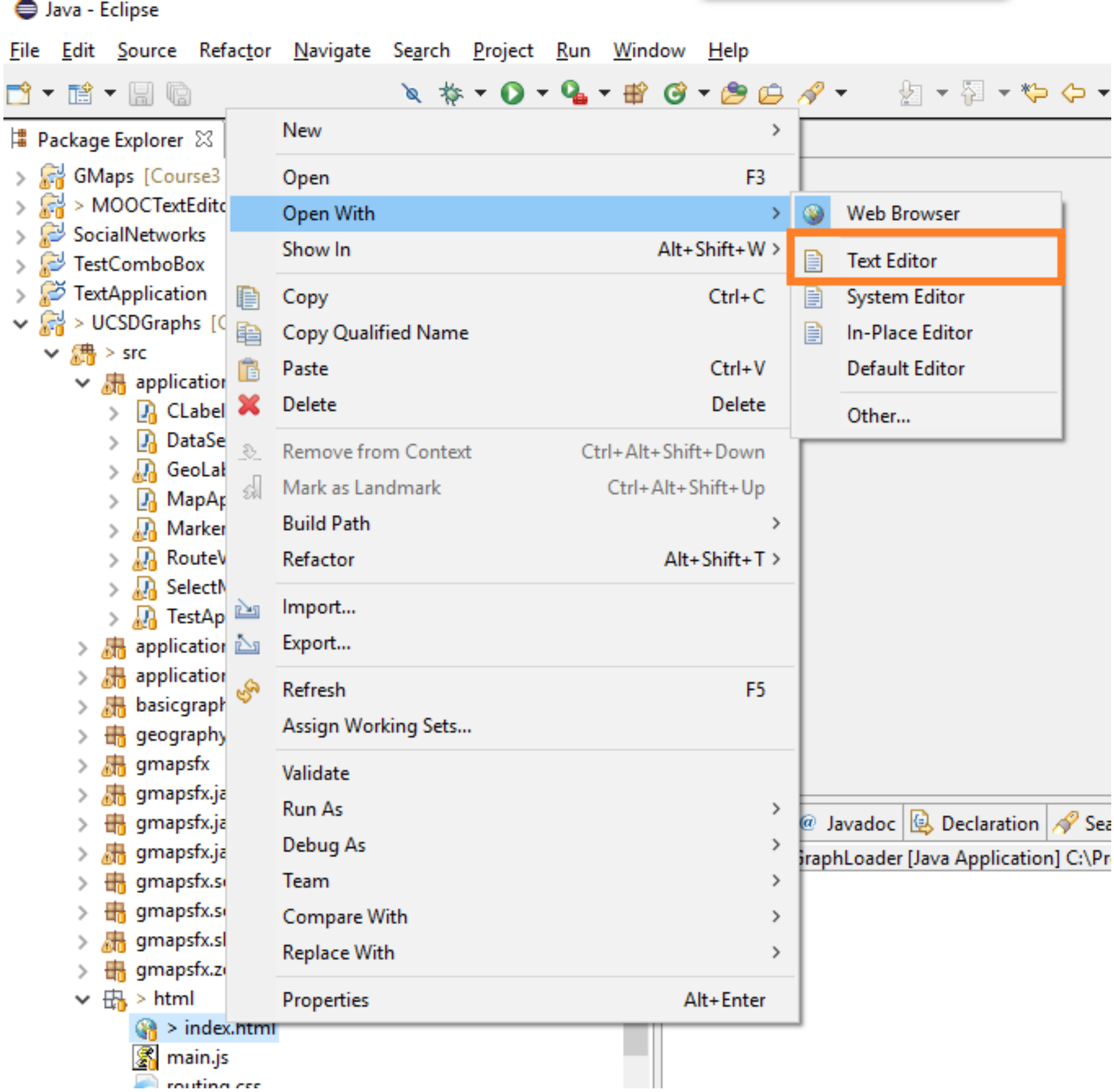
Name
Browser key 1

Accept requests from these HTTP referrers (web sites) (Optional)
Use asterisks for wildcards. If you leave this blank, requests will be accepted from any referrer. Be sure to add referrers before using this key in production.
.example.com/

Note: It may take up to 5 minutes for settings to take effect

6. Select and copy the key that appears in the popup window and then click OK. We'll be pasting this key into your code shortly. If you don't copy it successfully or need to get it later, it shows up in the Credentials screen when you click OK.

7. In Eclipse, find the file src/html/index.html in the Package Explorer (left-hand window). Right-click on the file and select Open With->Text Editor. *The file will not open for editing if you double-click on it.*



8. First, comment out the line

```
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"
  ></script>
```

by placing it between <!--- and --> like so:

```
<!--- <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"
  ></script> -->

```

Then, remove the comment markers from the line which you can find below the big comment:

```
<!--- <script src="https://maps.googleapis.com/maps/api/js?key
  =[APIKeyHere]&callback=initMap"></script> -->

```

So that it now looks like:

```
<script src="https://maps.googleapis.com/maps/api/js?key=[APIKeyHere]&callback
  =initMap"></script>

```

Then replace the text "[APIKeyHere]" with the API Key you copied in step 6. Make sure you copy the whole key and don't include leading or trailing spaces. Save and close this file.

9. Just to be sure your change took effect, you should probably by pressing F5 while you are in the UCSDGraphs project or by right-clicking on the project (or at least the src/html directory) and selecting "Refresh". Then, find and open the file application.MapApp.java and run this file. If your key worked correctly, you should see a map load and be able to interact with the application.

If things don't work, try closing eclipse and reopening it. If that still doesn't work, give it 5 minutes and try again. The keys need a little time to be processed sometimes.

If things still don't work, try posting in the discussion forums for help. And always remember that while the Map Application is very cool, you can do all the work in this course without it so you can still proceed with the course.

Complete