Programming Fundamentals 1

Produced by

Mr. Dave Drohan
(david.drohan@setu.ie)
Dr. Siobhán Drohan
Ms. Mairead Meagher

Department of Computing & Mathematics
South East Technological University
Waterford, Ireland

setu.ie

Introduction to Github

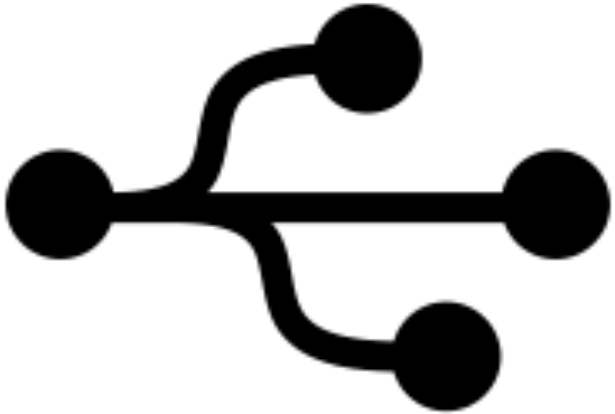
Source:

https://www.linkedin.com/learning/learning-github?u=57077417

Agenda

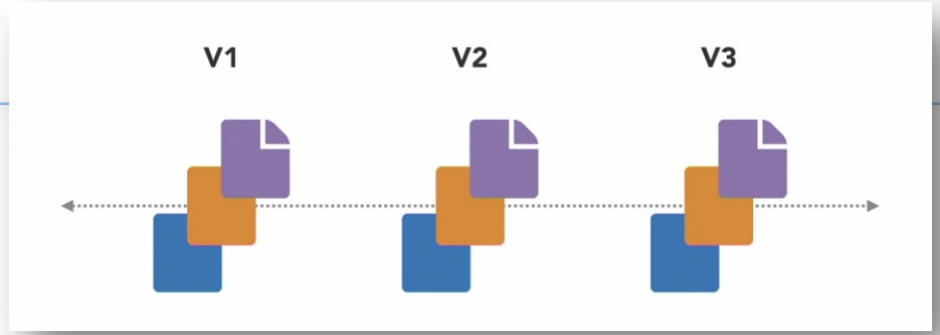
- Version Control
- Repository tour
- Vehicle App Example
- Demonstration…

Version Control – What is git?

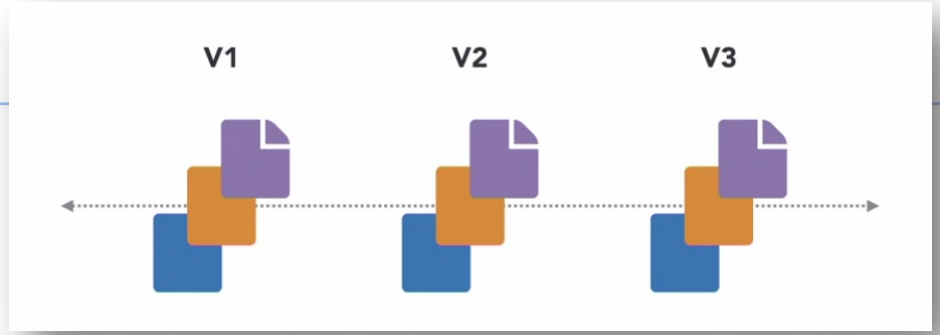


"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Developed by Linus Torvalds

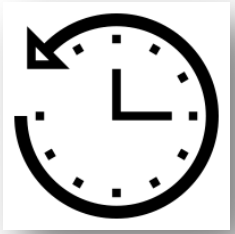


A Version Control System (VCS) is software designed to record changes made to a file or set of files over time.

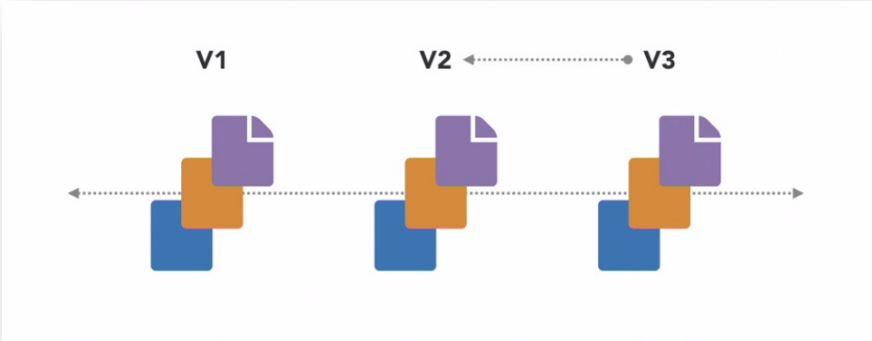


A Version Control System (VCS) is software designed to record changes made to a file or set of files over time.

This gives you the ability to revert or recall changes to a file, or set of files after you have made them.



If you want to go back to a version, you can revert one file to a previous version, or a selection of files to their previous versions, or even the entire project back to a specific version in time.



You can see what changes were made between files e.g.:

- When the change was introduced
- Who introduced the change
- What was changed
- Why was it changed

git

JAVA

You can version control any type of file on git, not just source code files:

Basics of git - 1

- Git is a version control system. Git manages all the repo's history, etc.

- When using Git, we have a Git repository (repo) which contains:
  - ❖ Source code and any other files
  - ❖ Files needed to keep the information of the various versions (these are contained in the hidden .git subdirectory)

- The user works on the **local** repo.

- Usually (but not necessarily), the user will also use a **remote server** so that
  - ❖ They can backup their work;
  - ❖ They can publish their work;
  - ❖ They can share their work with collaborators. This sharing is the most productive use of Git but as a starting point, not necessary.

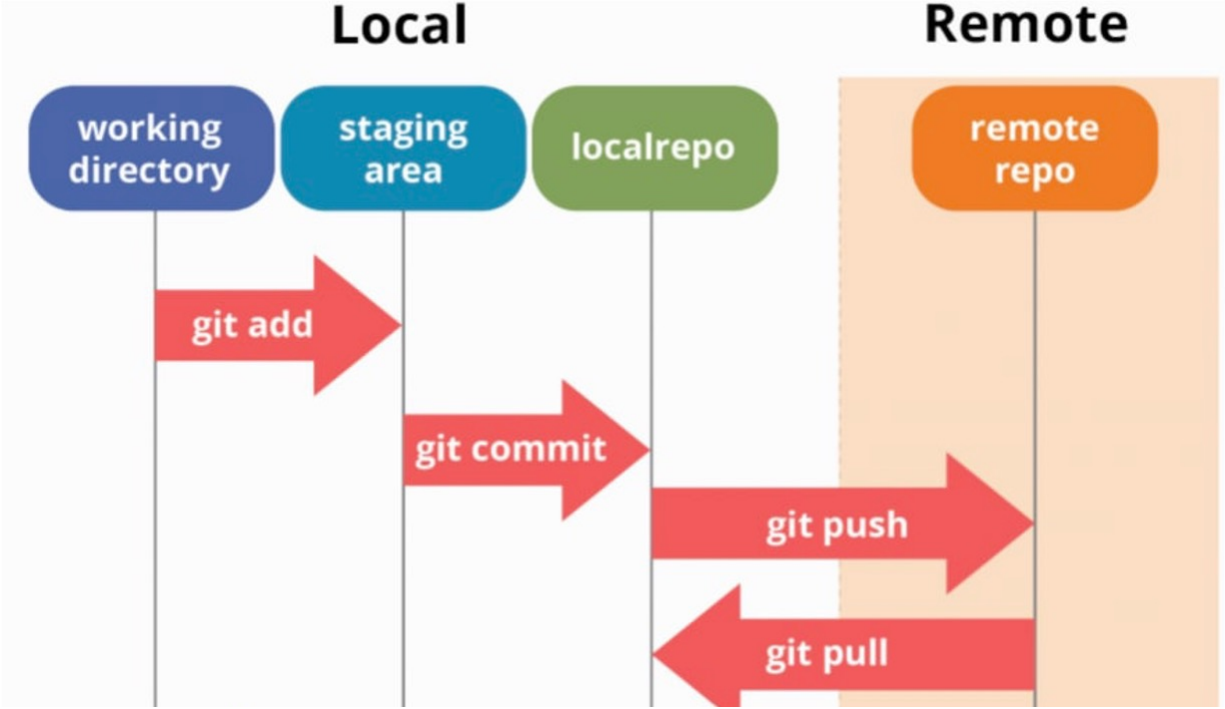
Basics of git - 2

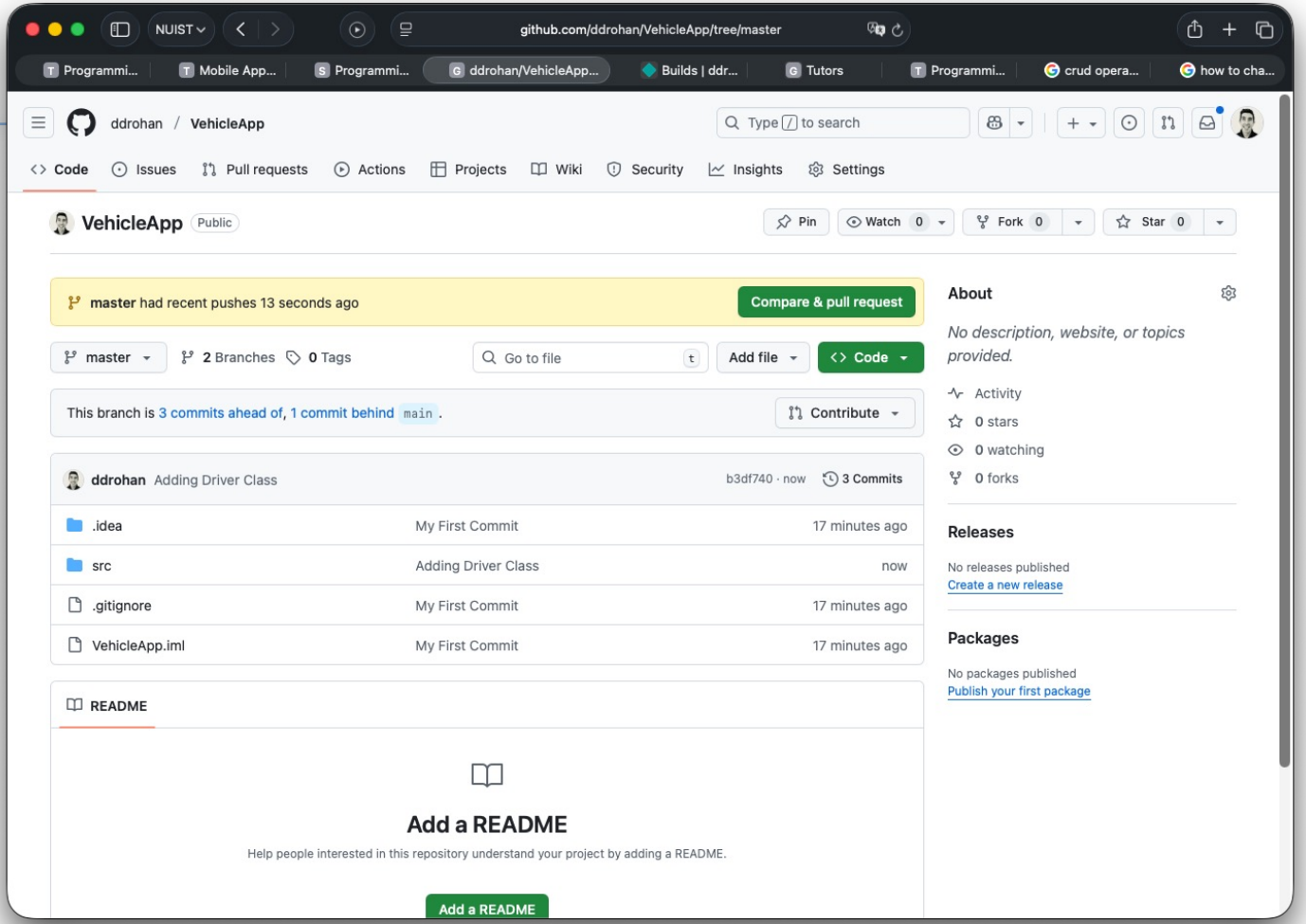
- To see a very good intro on git , see here :

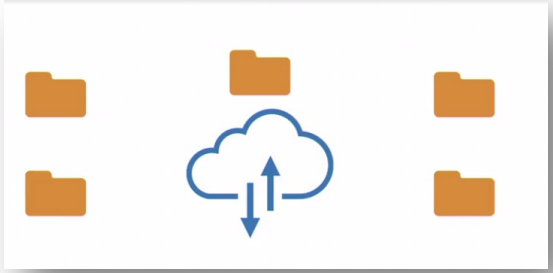
https://www.freecodecamp.org/news/learn-the-basics-of-git-in-under-10-minutes-da548267cc91/

Basics of git - 3

Basic commands



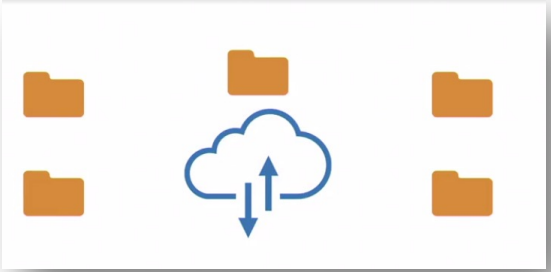

github.com/ddrohan/VehicleApp/tree/master
ddrohan / VehicleApp
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
VehicleApp
Public
Pin
Watch 0
Fork 0
Star 0
master had recent pushes 13 seconds ago
Compare & pull request
master
2 Branches
0 Tags
Go to file
t
Add file
Code
This branch is 3 commits ahead of, 1 commit behind main.
Contribute
ddrohan Adding Driver Class
b3df740 · now
3 Commits
.idea
My First Commit
17 minutes ago
src
Adding Driver Class
now
.gitignore
My First Commit
17 minutes ago
VehicleApp.iml
My First Commit
17 minutes ago
README
Add a README
Help people interested in this repository understand your project by adding a README.
Add a README
About
No description, website, or topics provided.
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package



Distributed Version Control allows for collaboration.

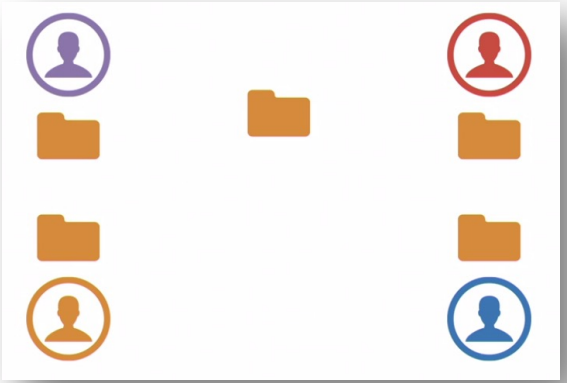
Each client fully mirrors the project by pulling down the entire repository.

git



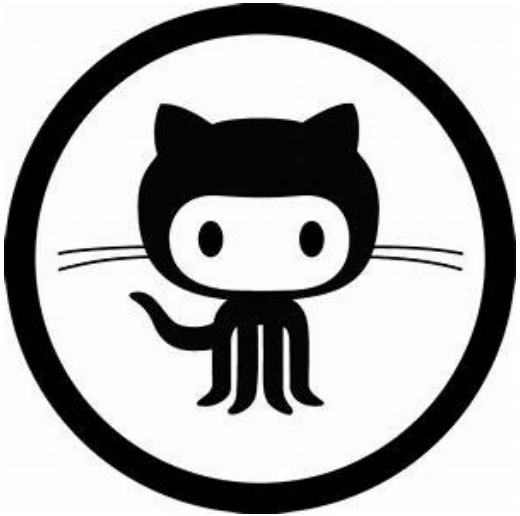
Distributed Version Control allows for collaboration.

Each client fully mirrors the project by pulling down the entire repository.



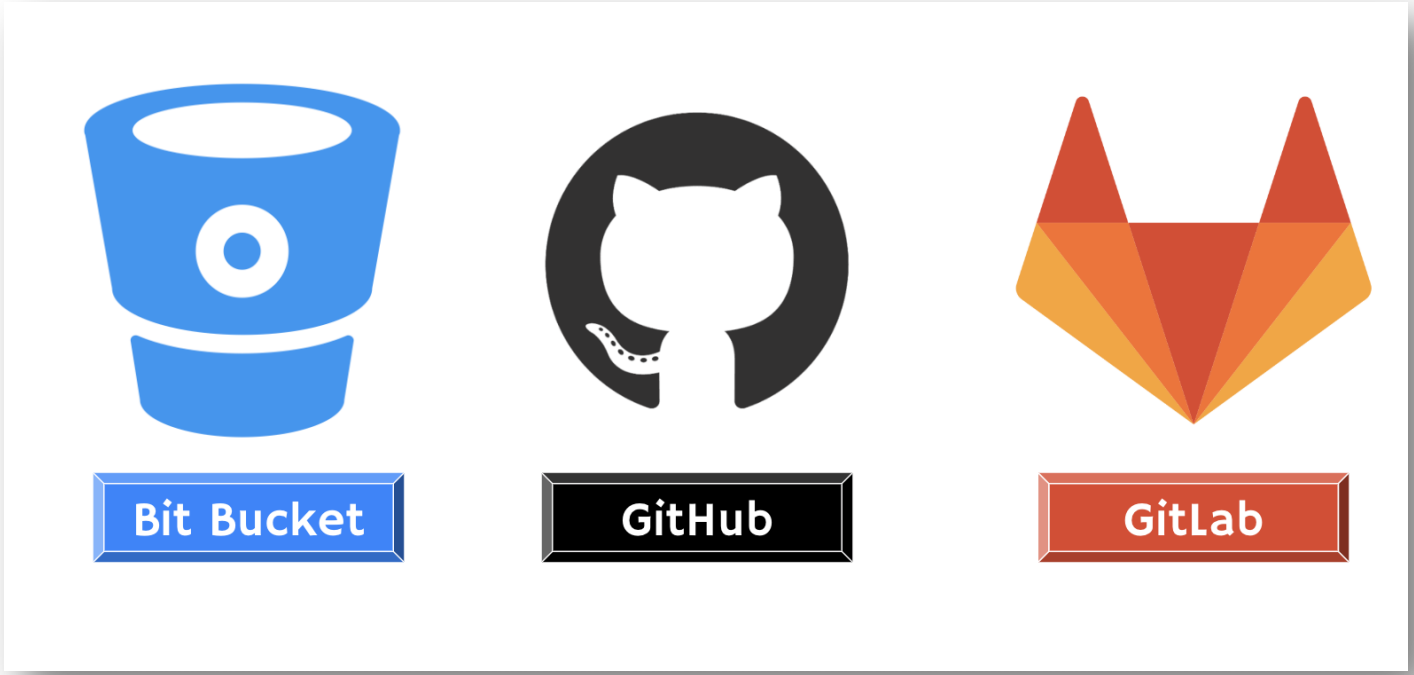
Everyone has a local copy of the entire project history → full back up.

A tour of a repository

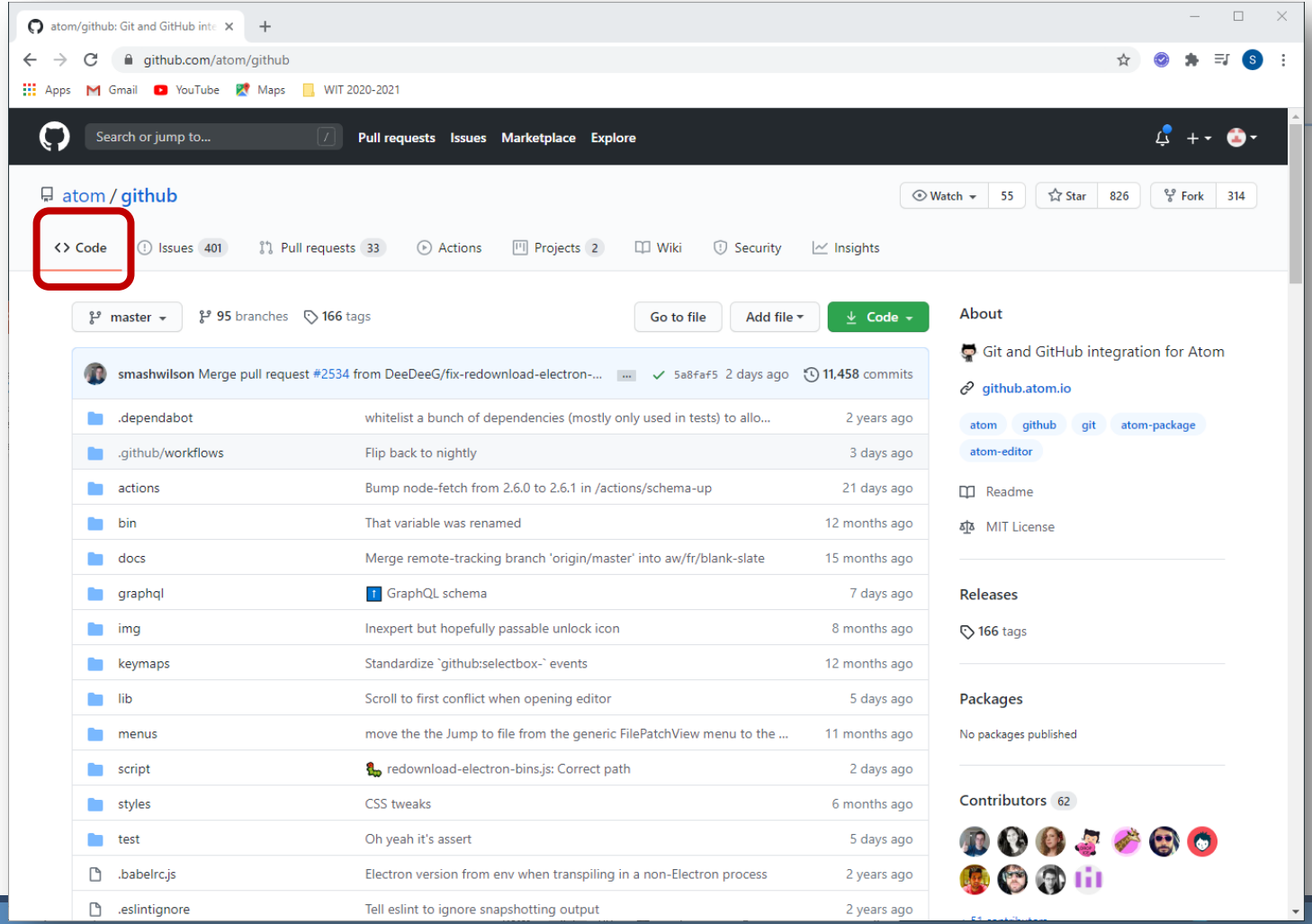


https://github.com/atom/github

Hosting services for Git repositories



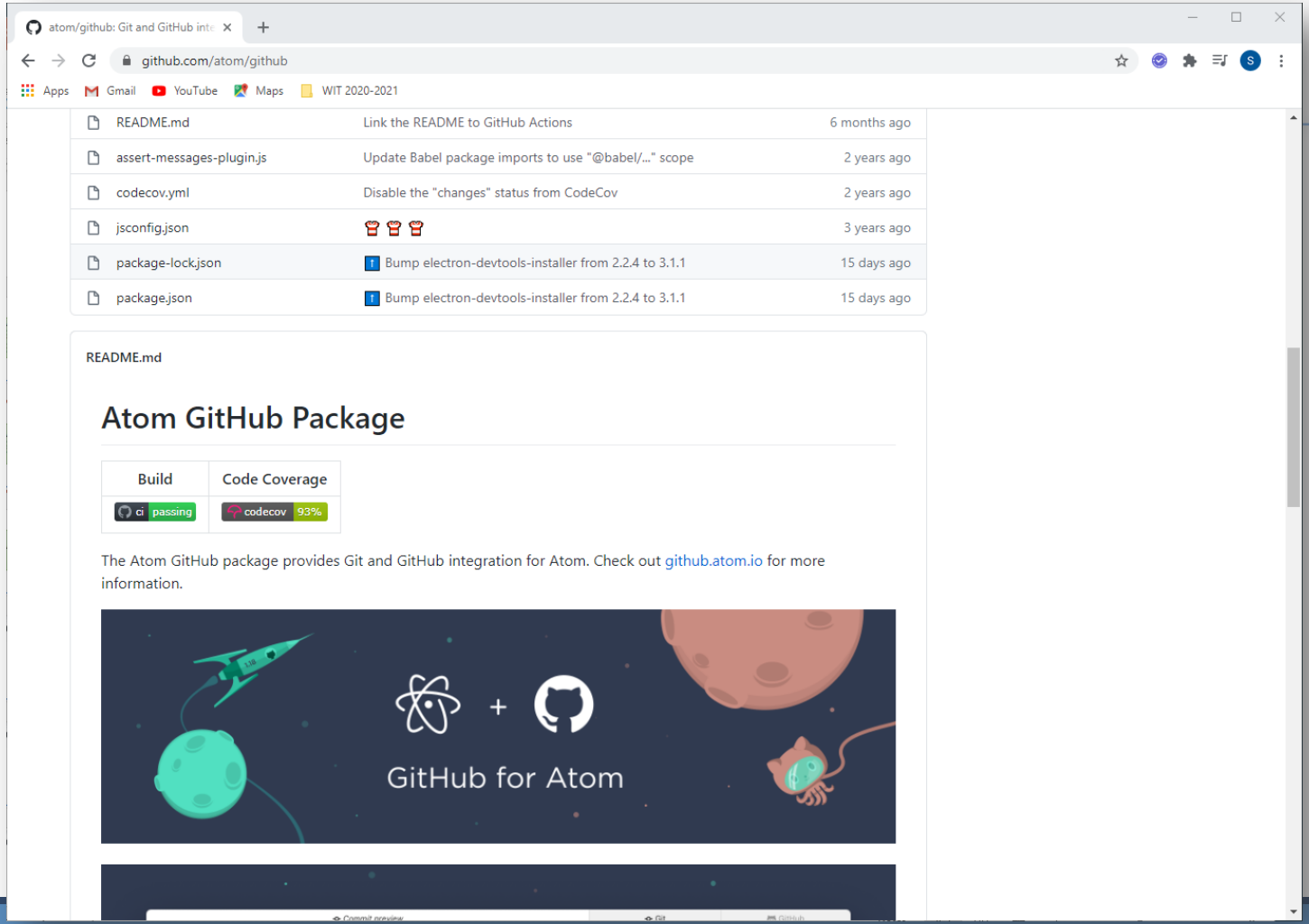
https://medium.com/swlh/what-do-i-need-to-know-about-git-5017bde0b572



Code View

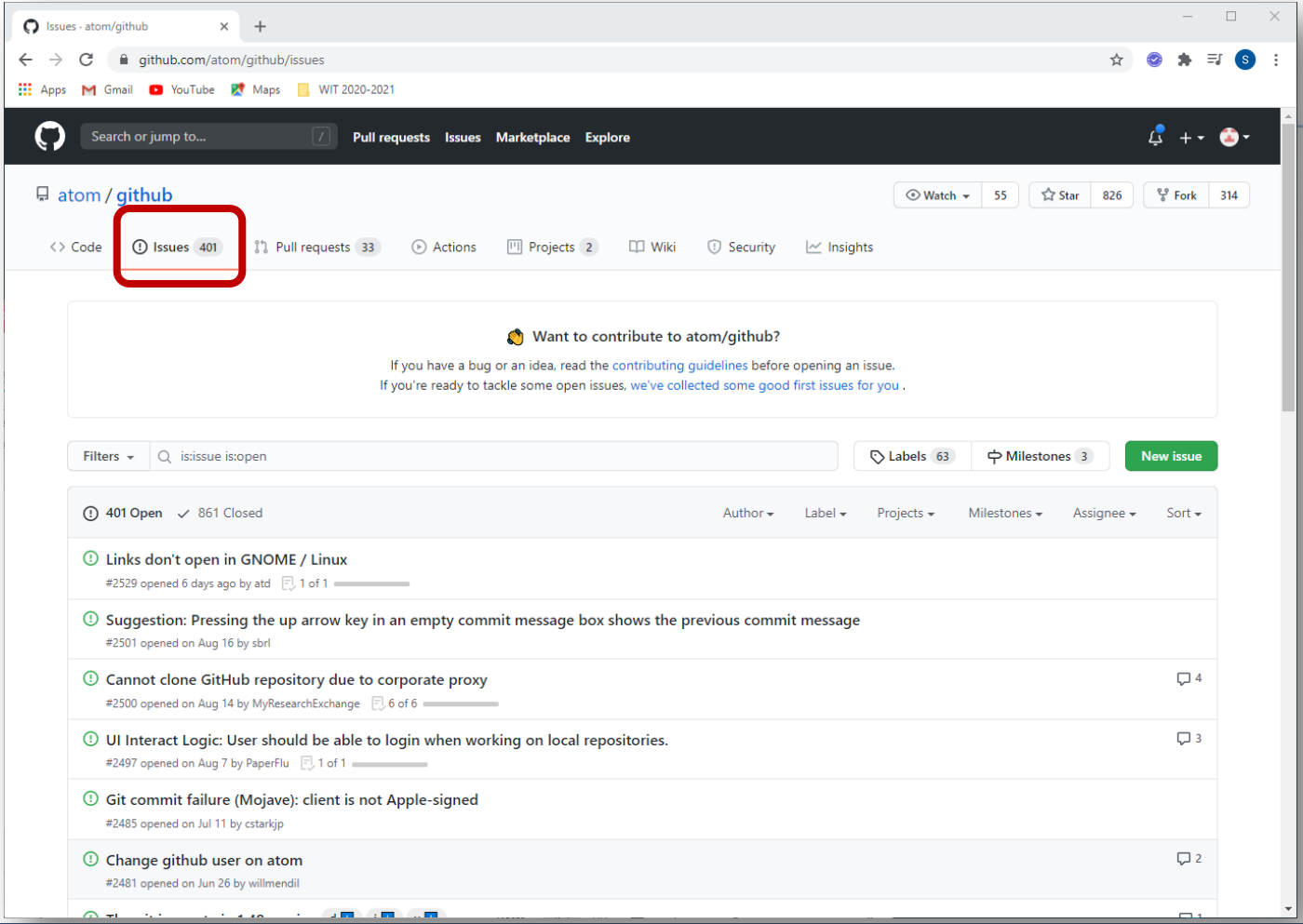
All files and folders in project.

Can edit file contents via GitHub UI.



Code View

README.md file is displayed after files and folders

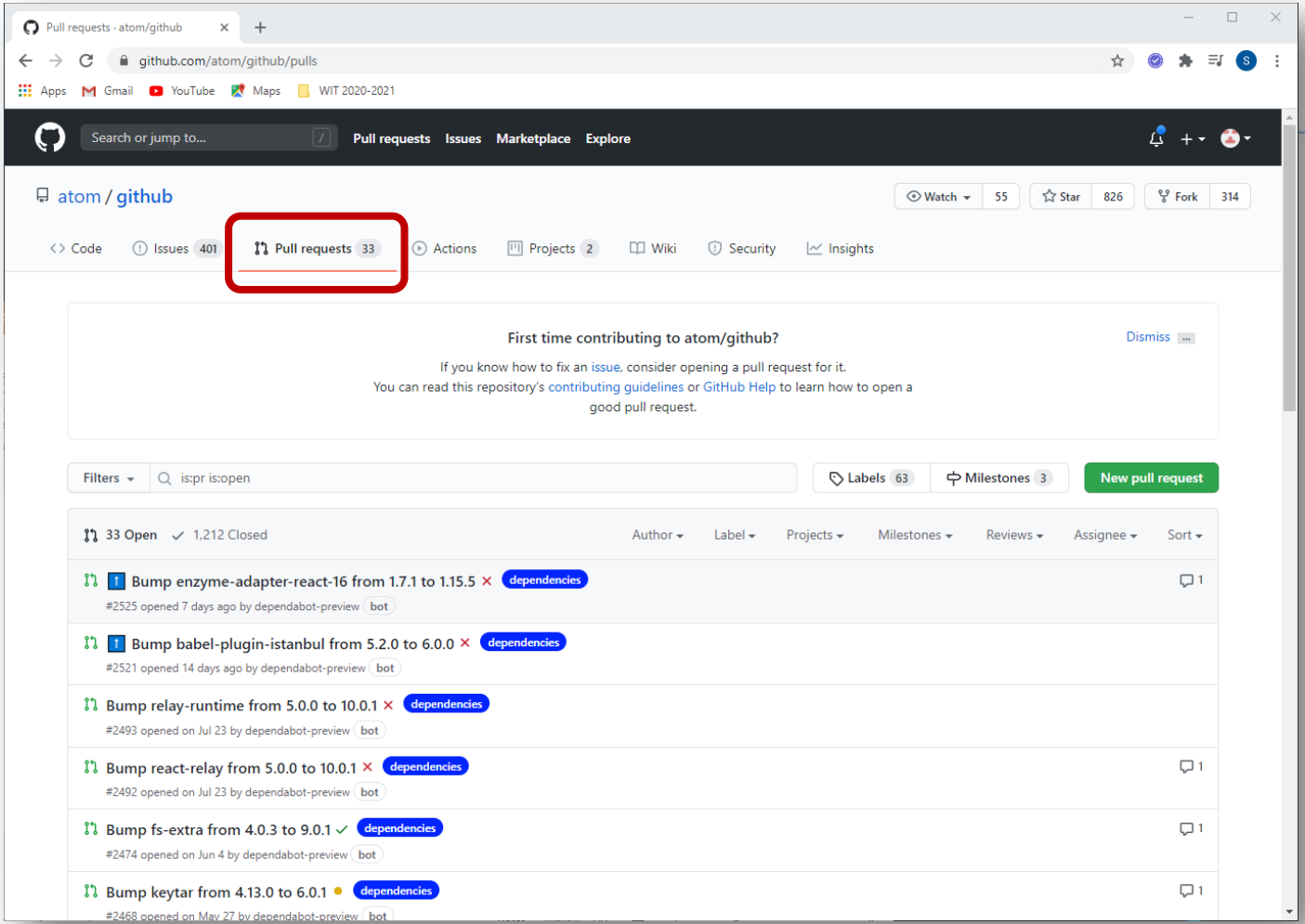


Issues View

Used to track bugs, feature requests, etc.

Can be assigned to specific members/ teams.

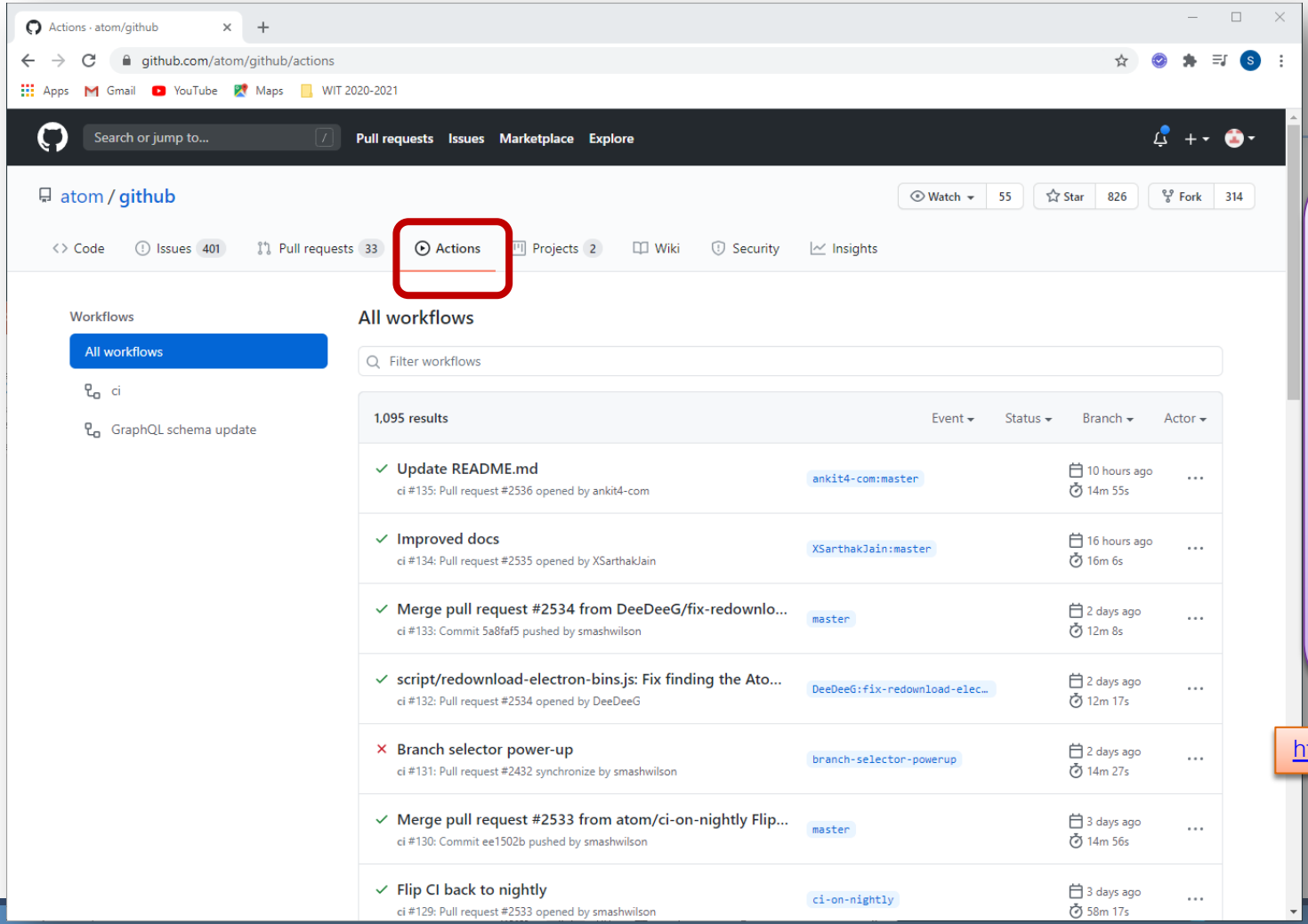
No code changes are attached.



Pull Requests

Represents a change, such as adding, deleting, updating files that the author of the repo wants to make.

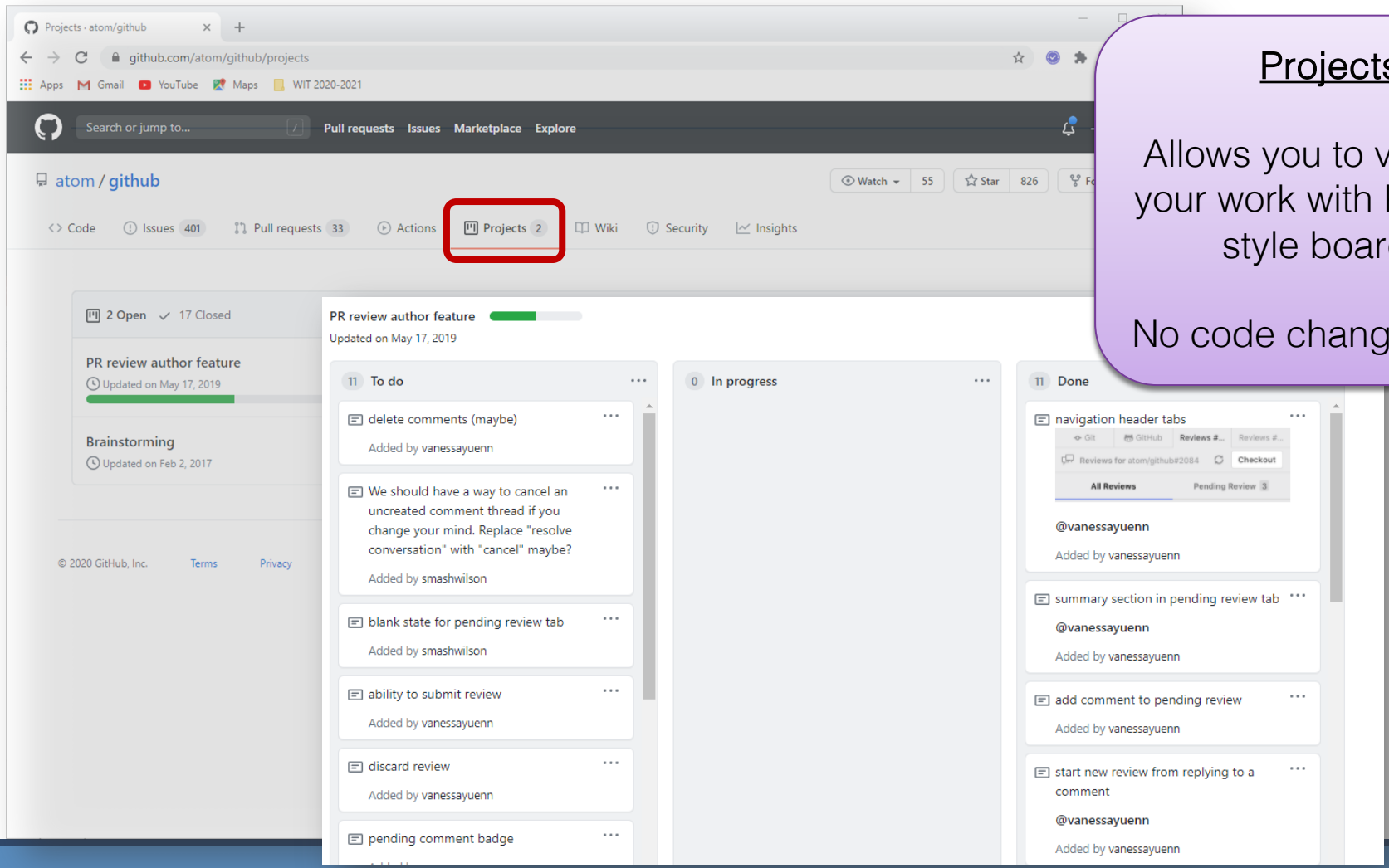
It includes code changes.



Actions

Allows you to automate your workflows with CI/CD. You can build, test and deploy your code from GitHub.

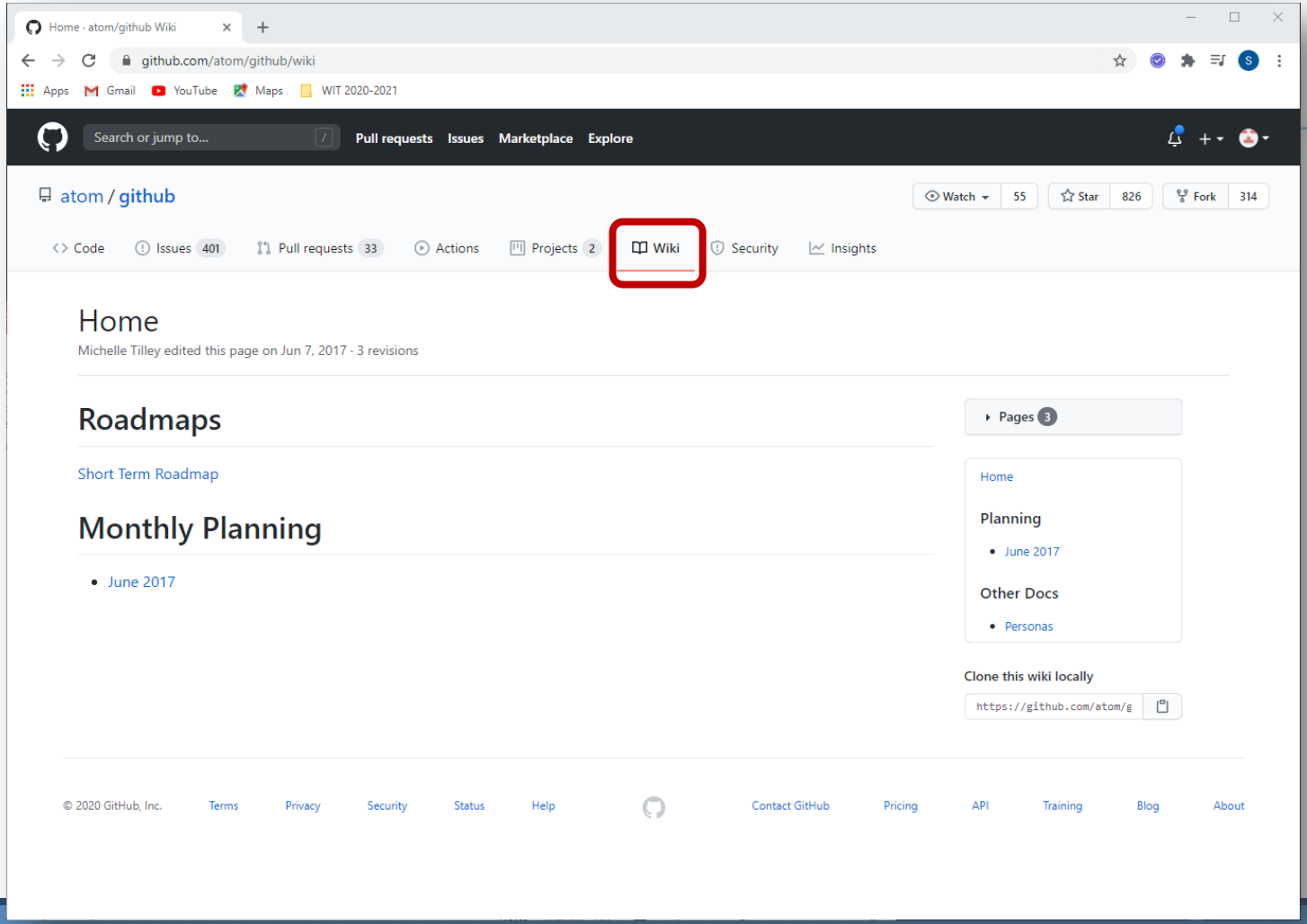
https://github.com/features/actions



Projects

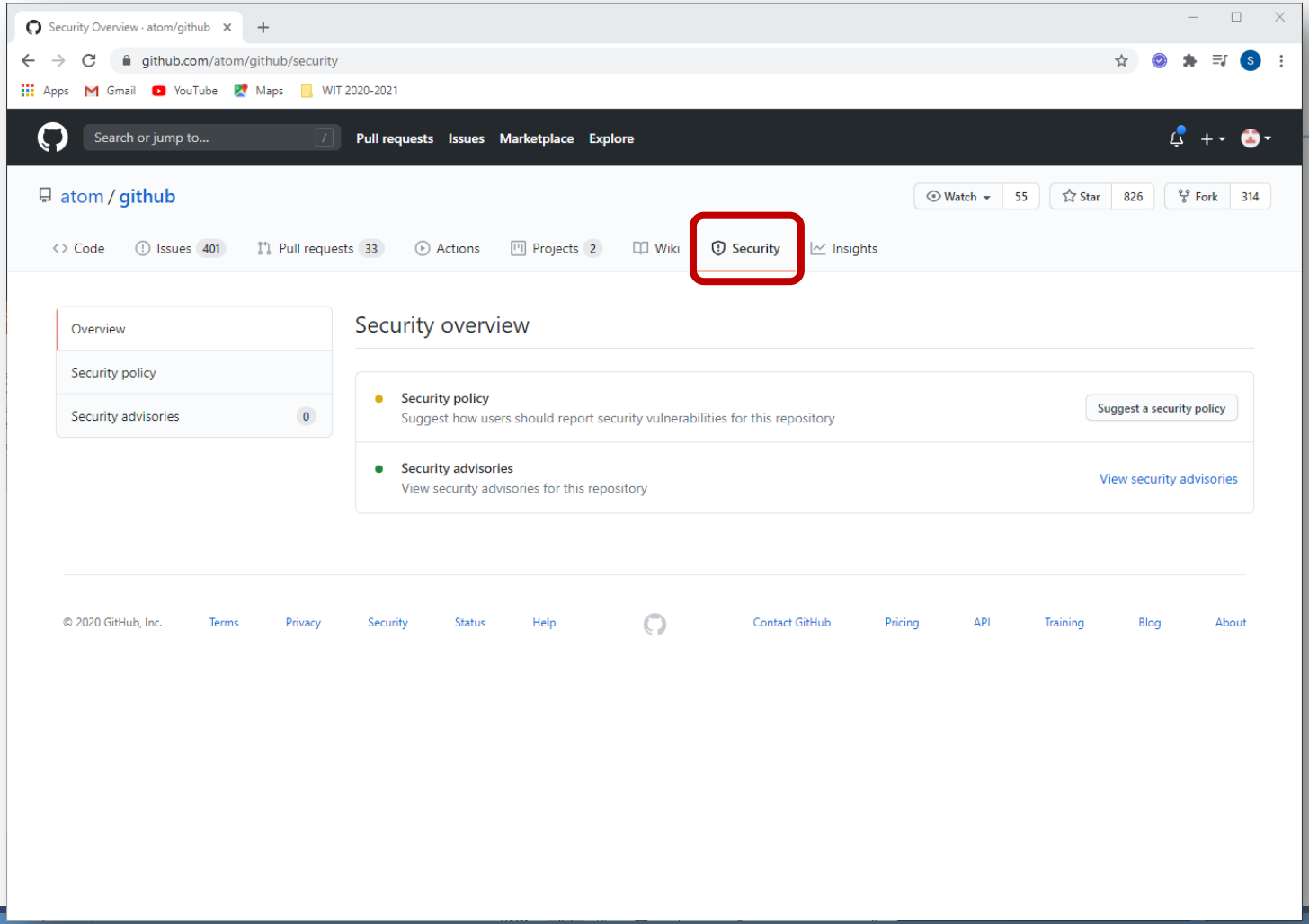
Allows you to visualise your work with Kanban-style boards.

No code changes here!



Wiki

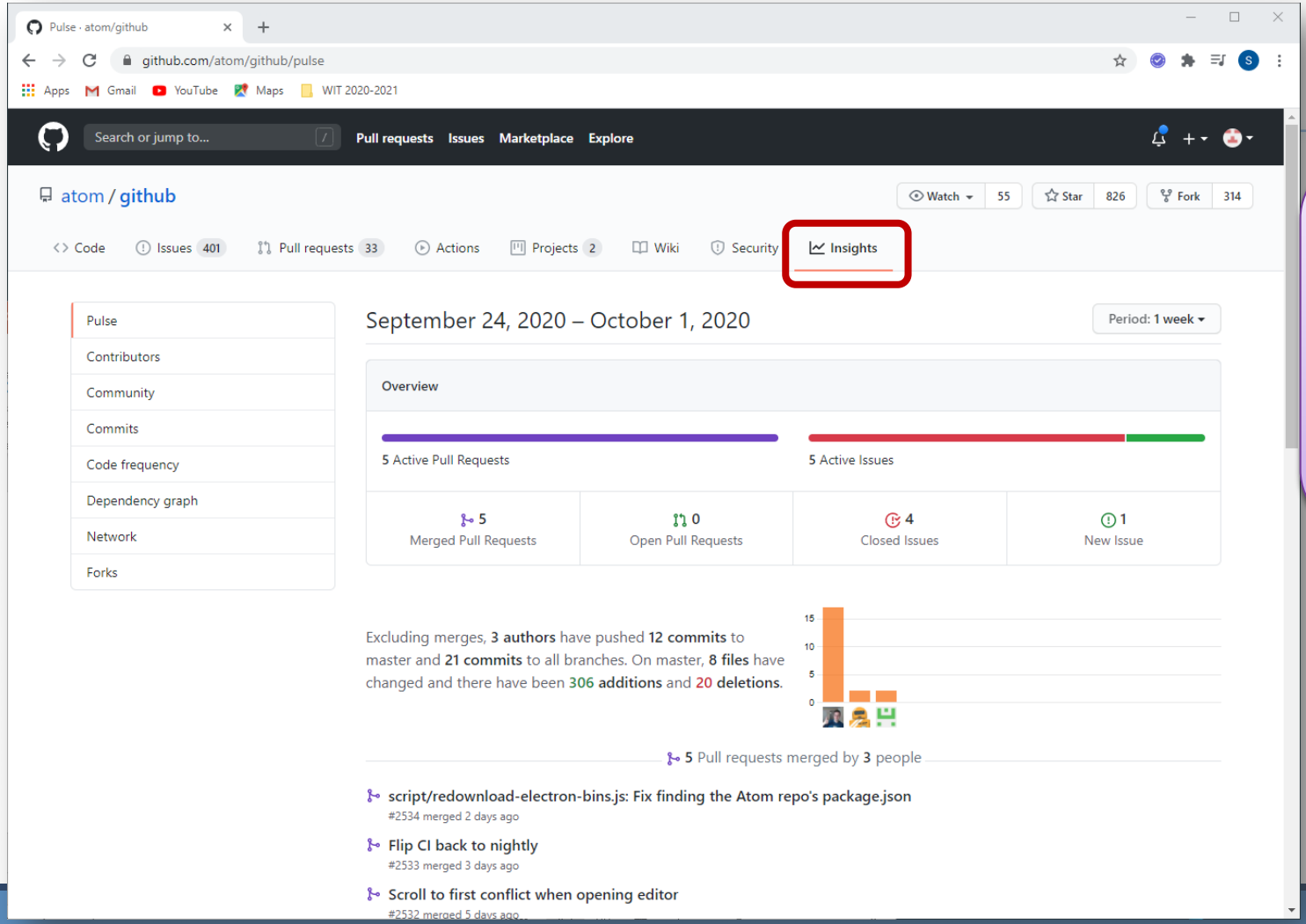
Documentation you can add to your project.



Security

You can give instructions on how to responsibly report a security vulnerability in your project. You can also view any security advisories for your repo.

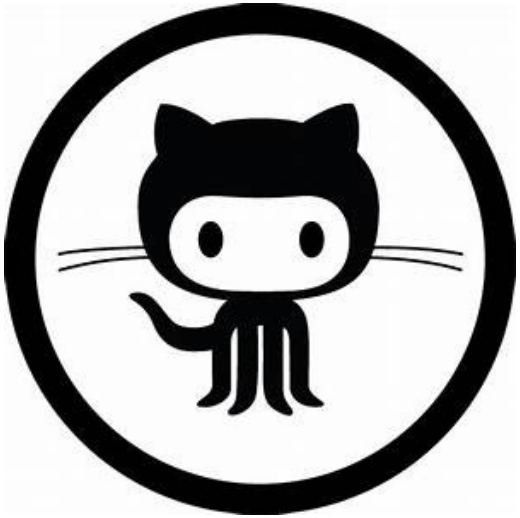
https://docs.github.com/en/free-pro-team@latest/github/managing-security-vulnerabilities/managing-security-vulnerabilities-in-your-



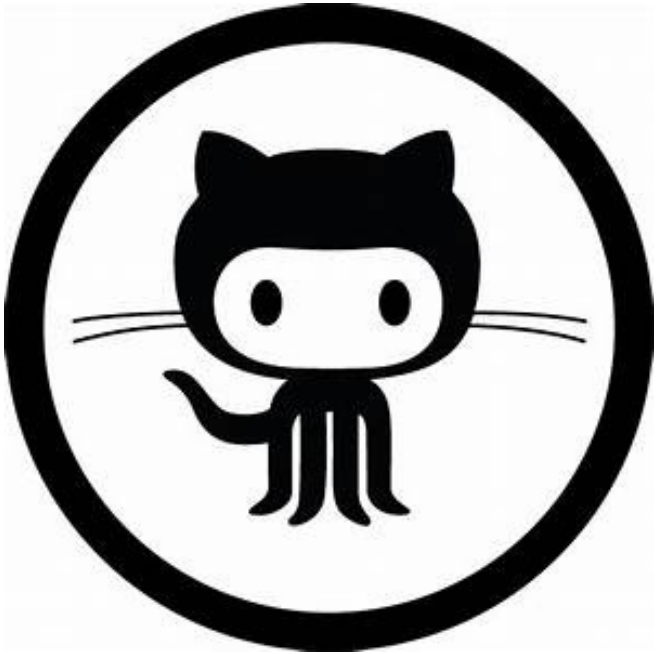
Insights

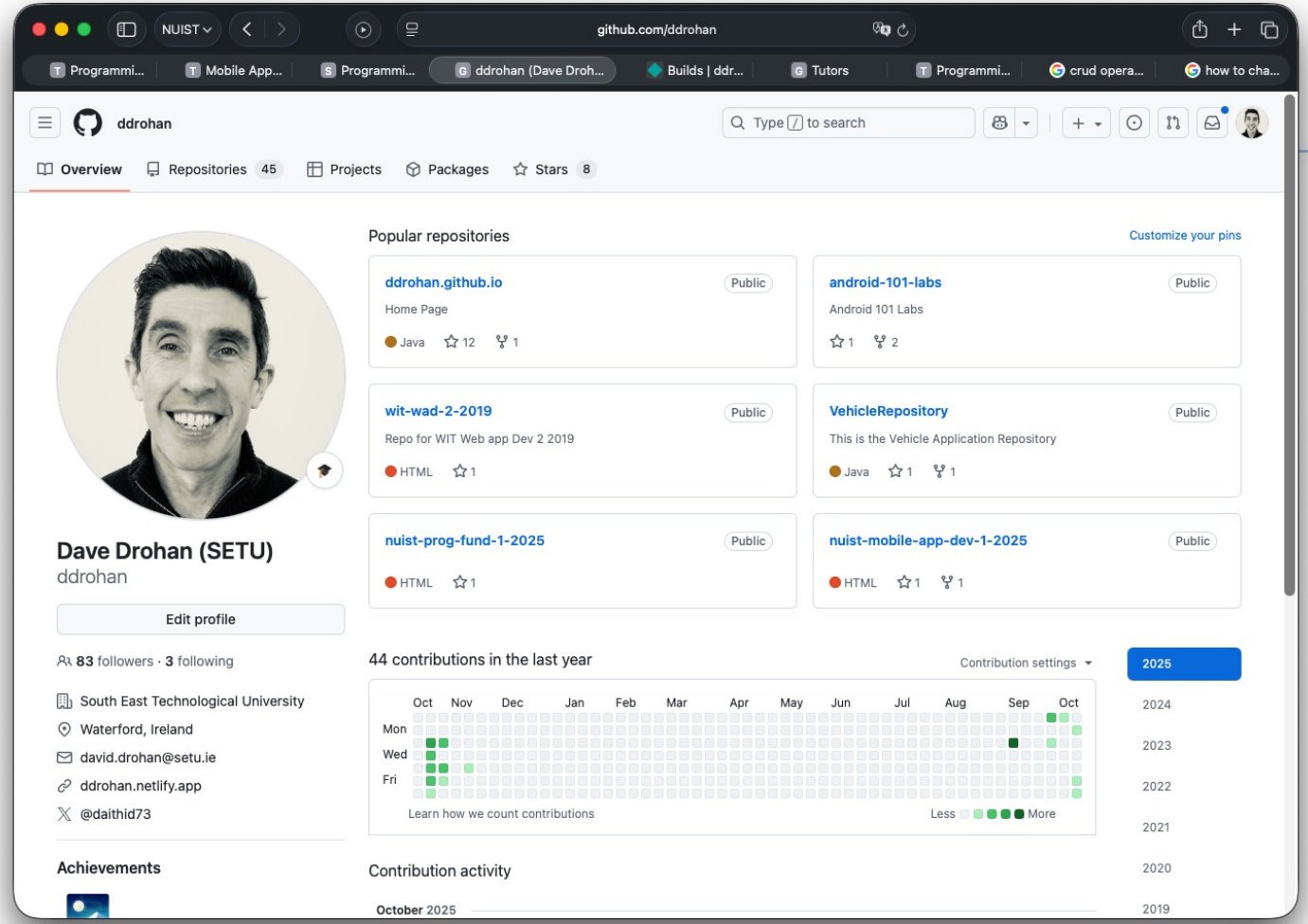
High level overview of the health / pulse of your repo.

Vehicle App Example

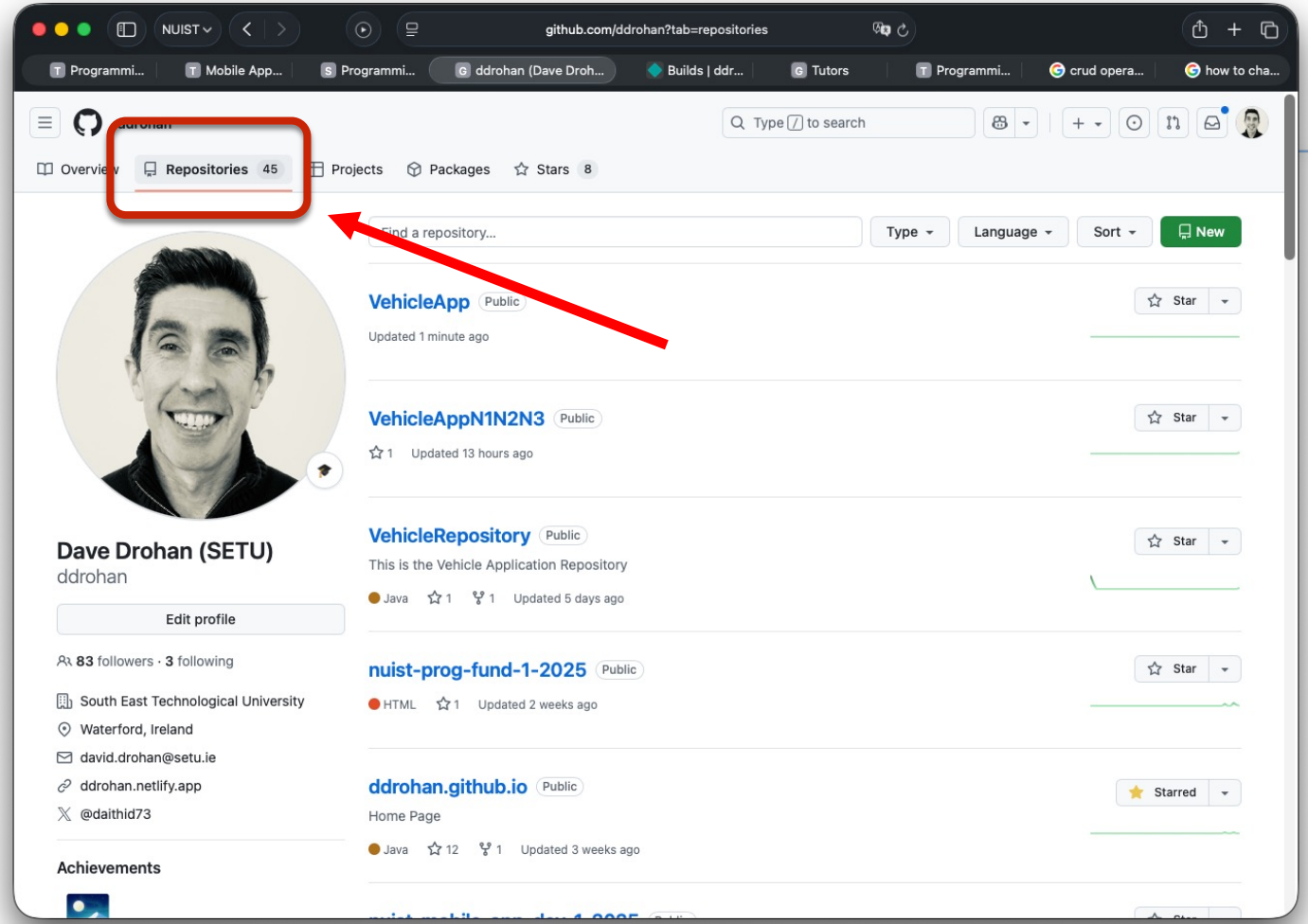


Step 1 – Create your Repository

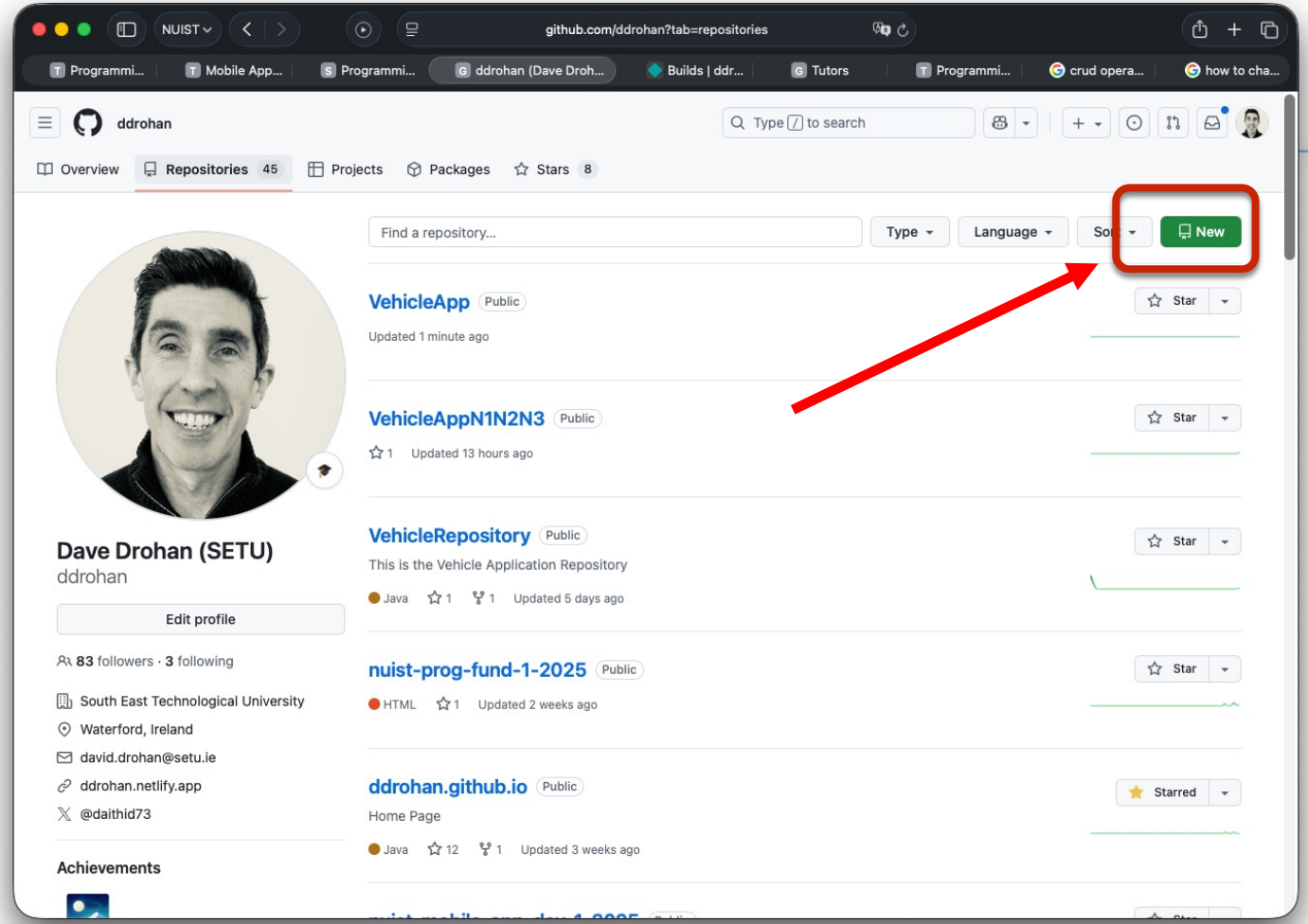




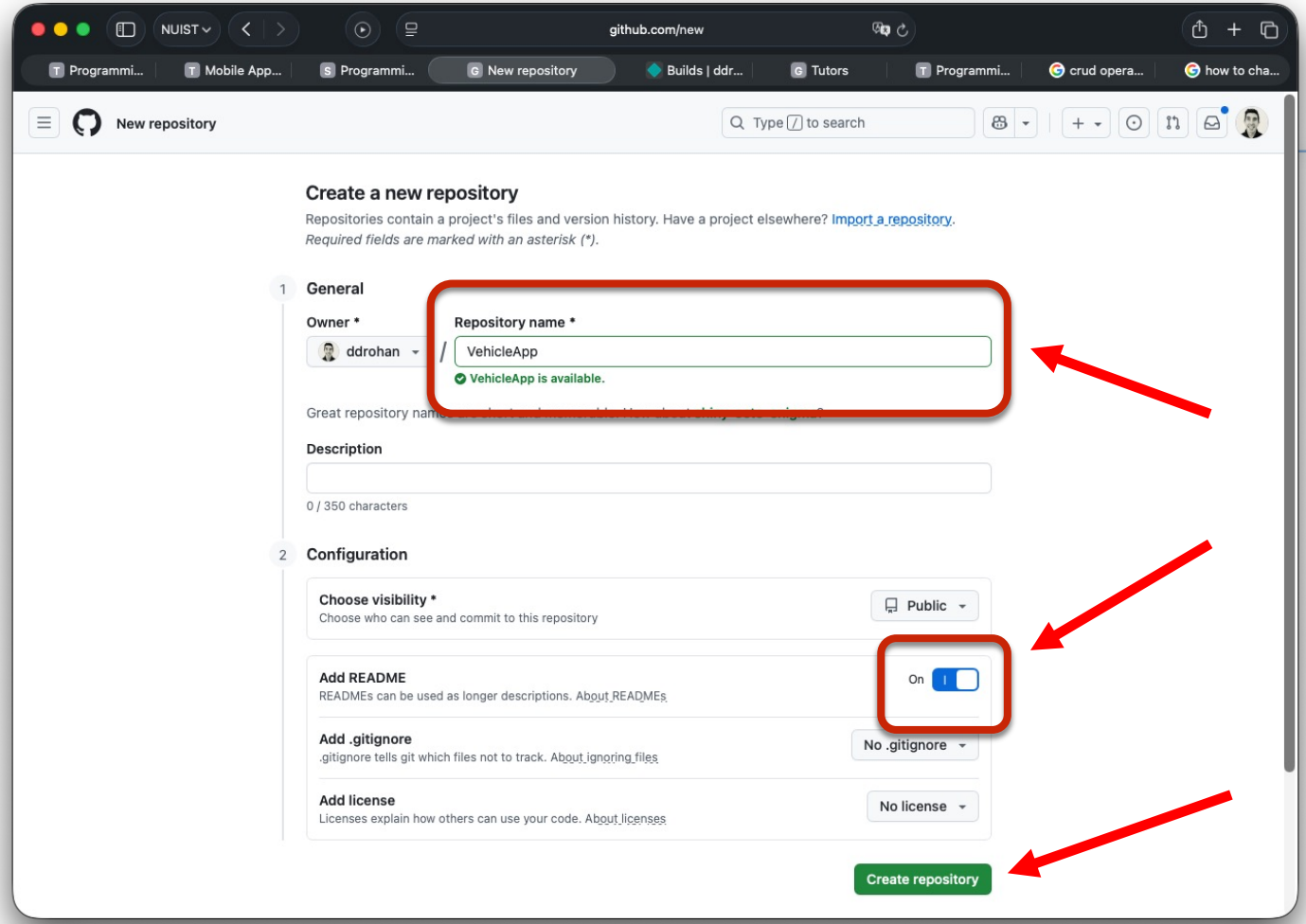
Login to github.com



Select Repositories tab

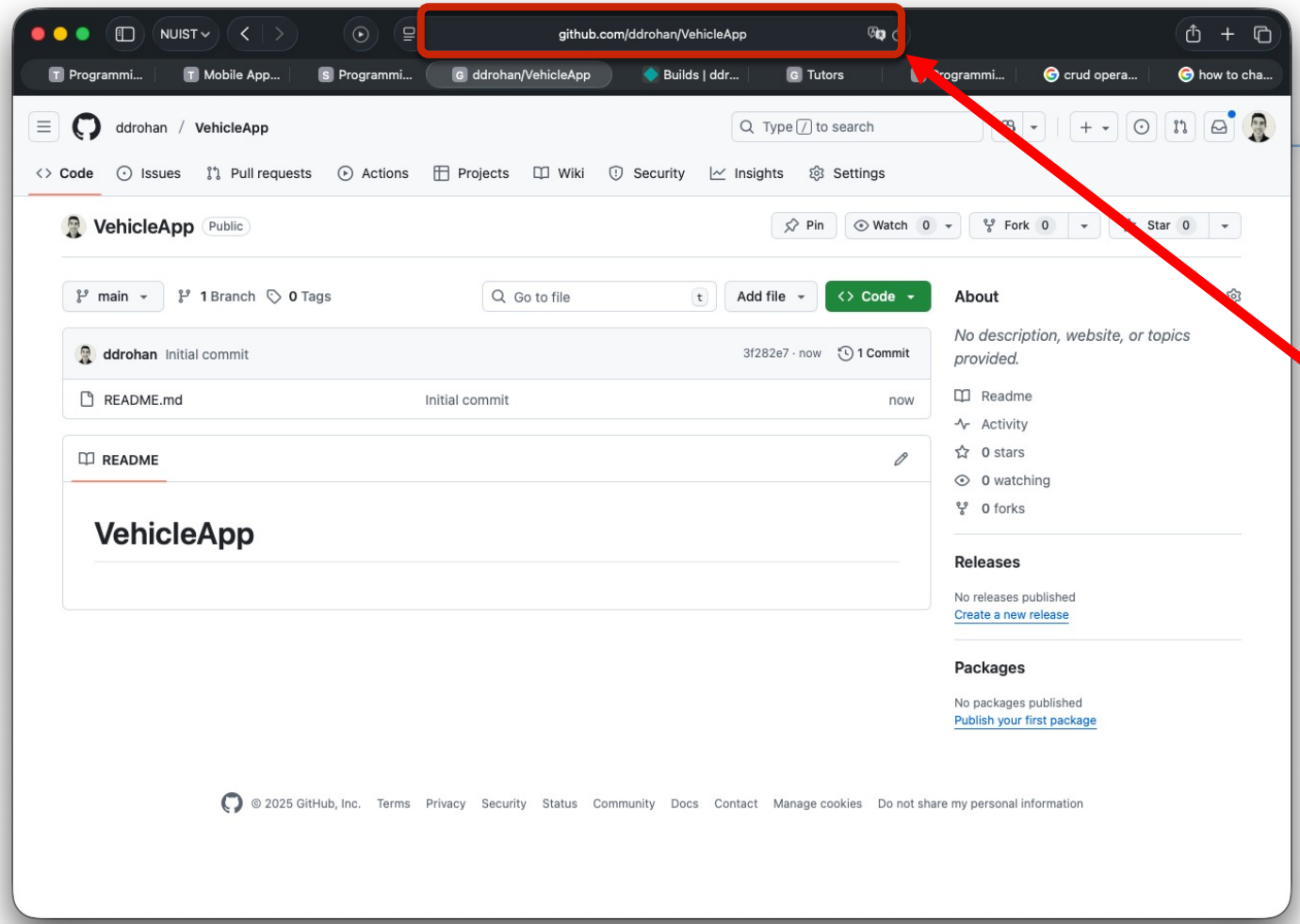


Create ‘New’ Repository



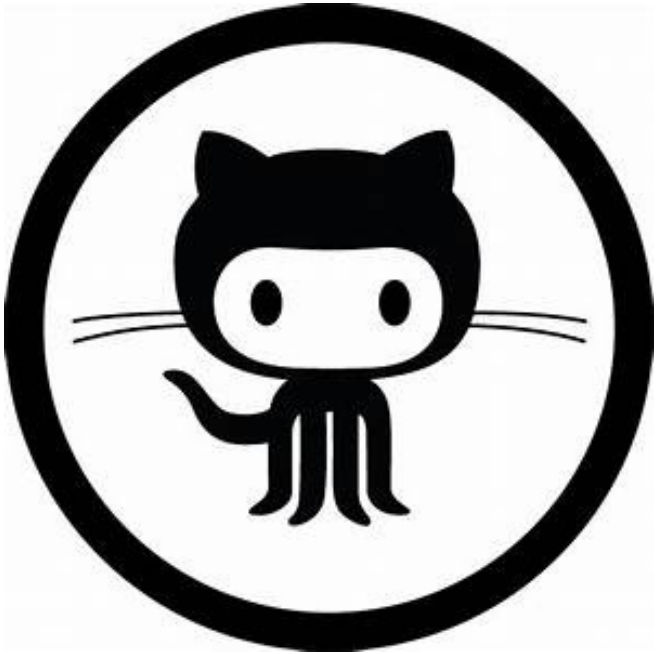
Name your repository and add a README

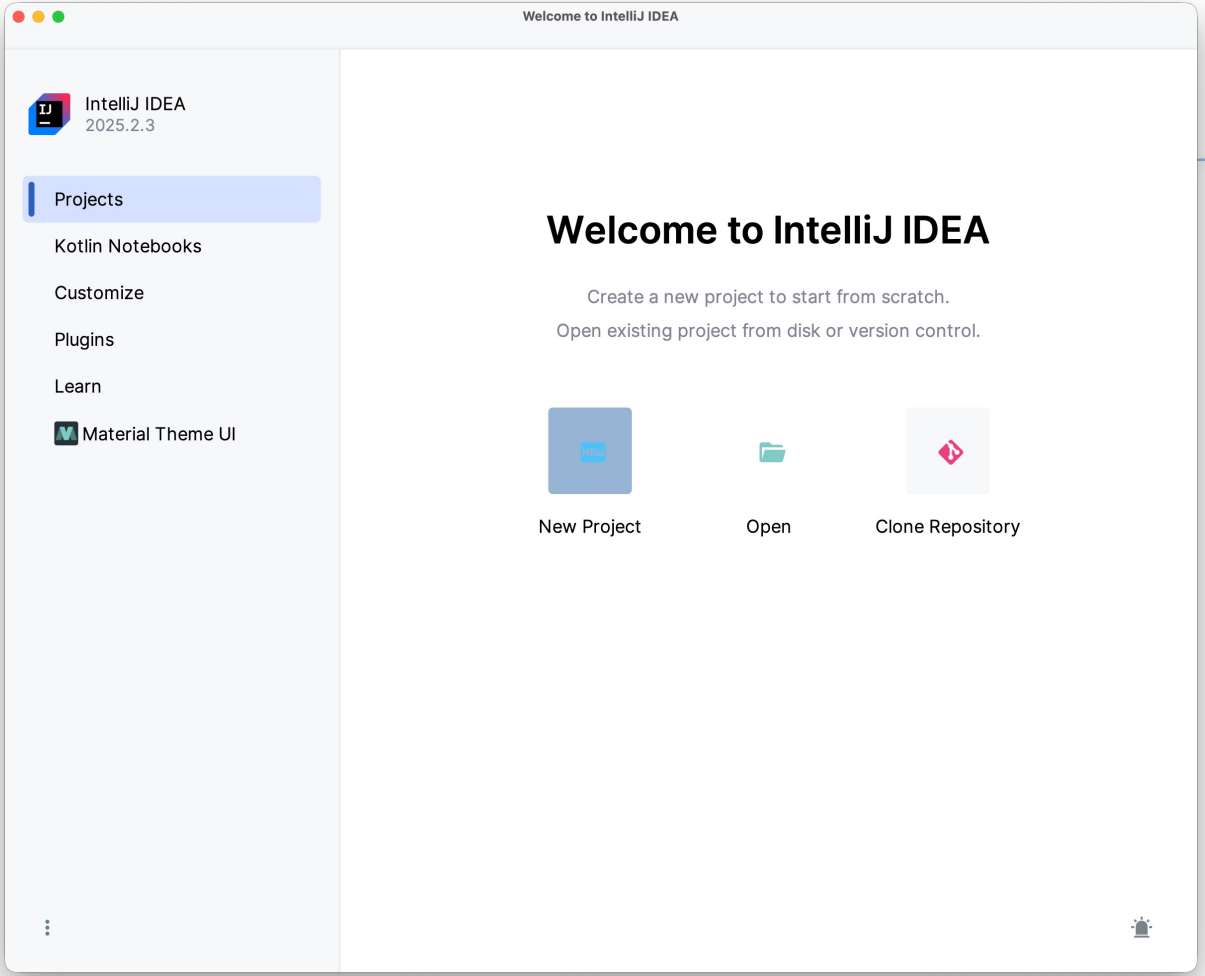
then Create



Copy this link as we will use it later

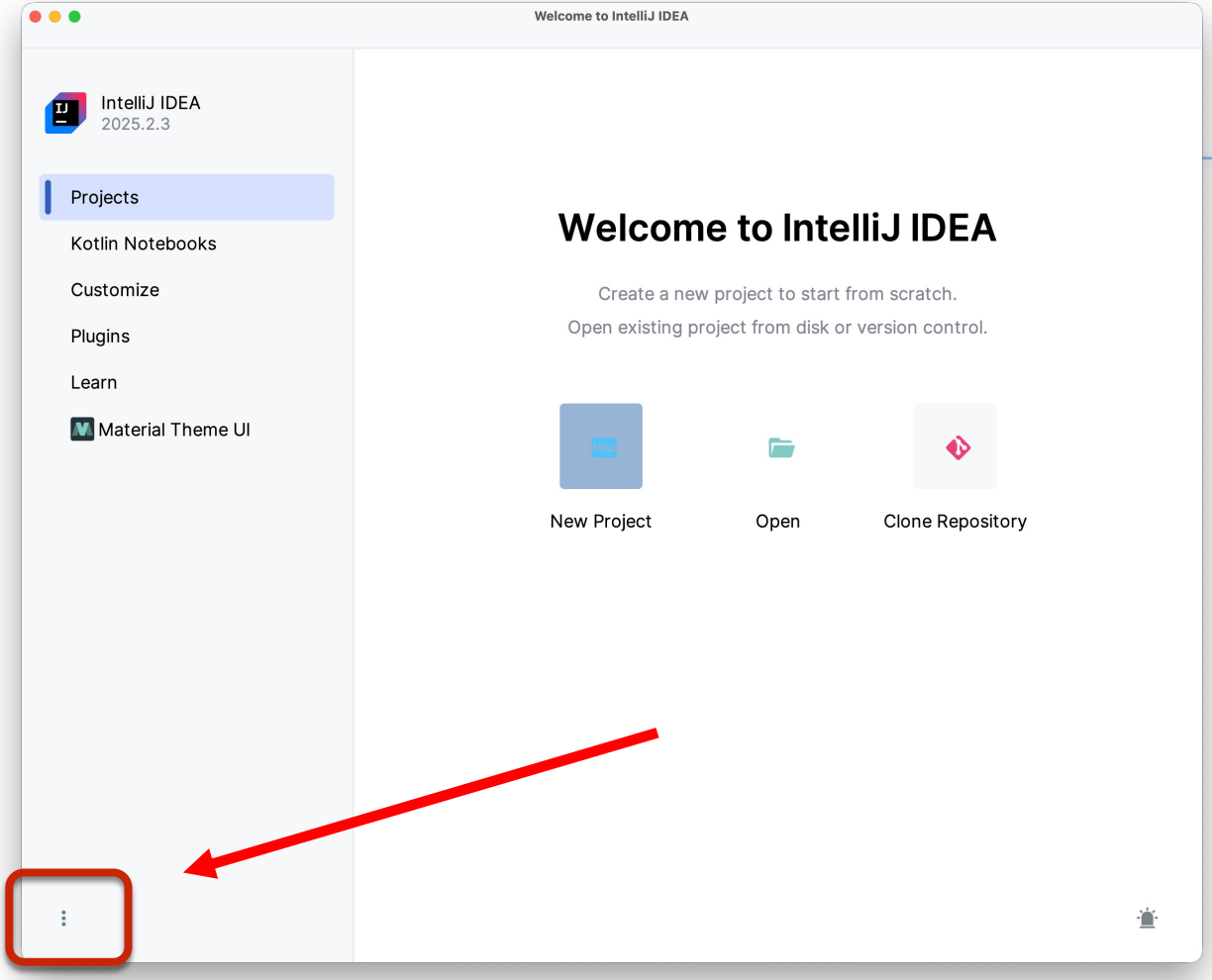
Step 2 – Configure Intellij with Github



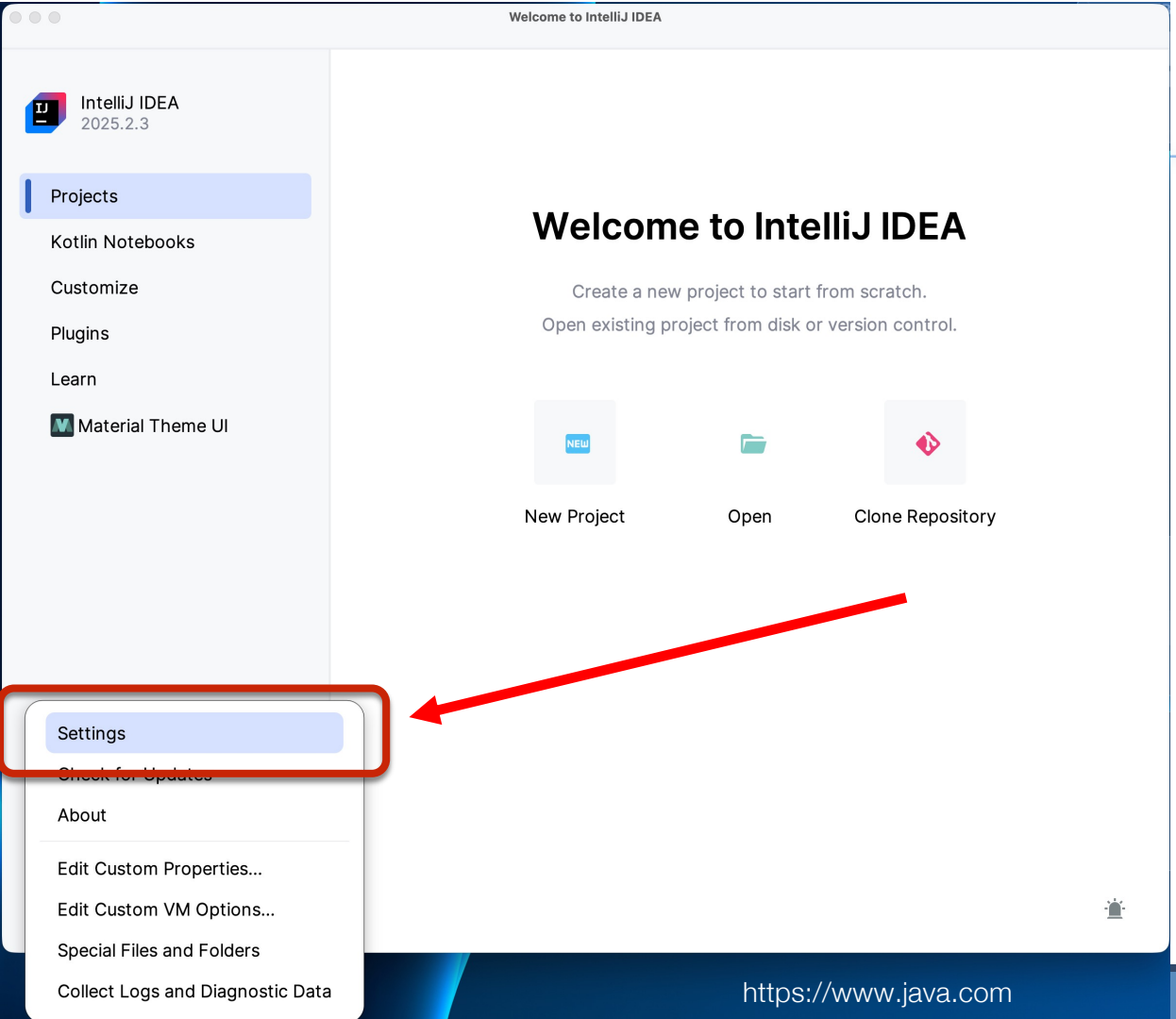


Launch Intellij

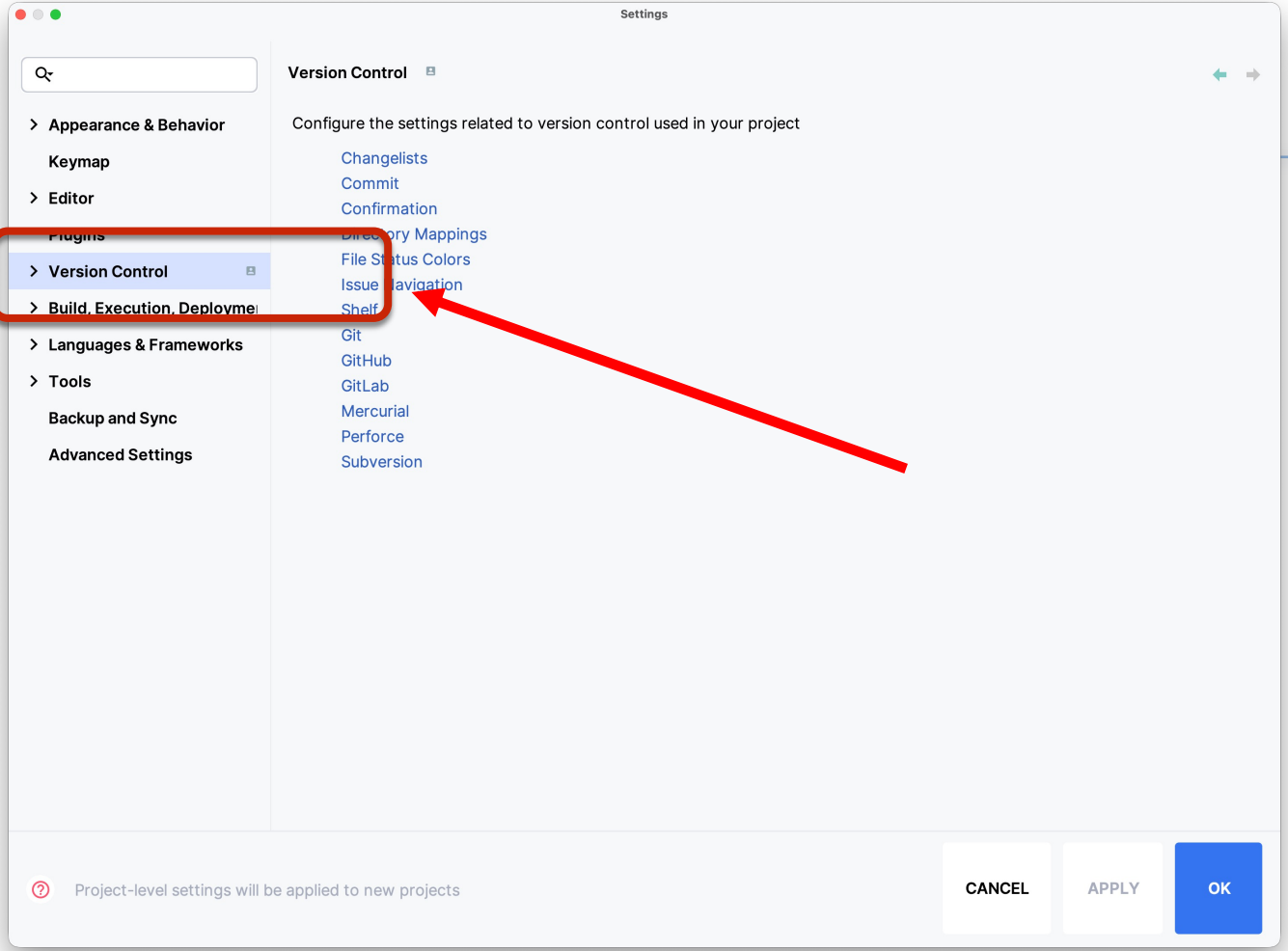
If a project is open select File->Close Project



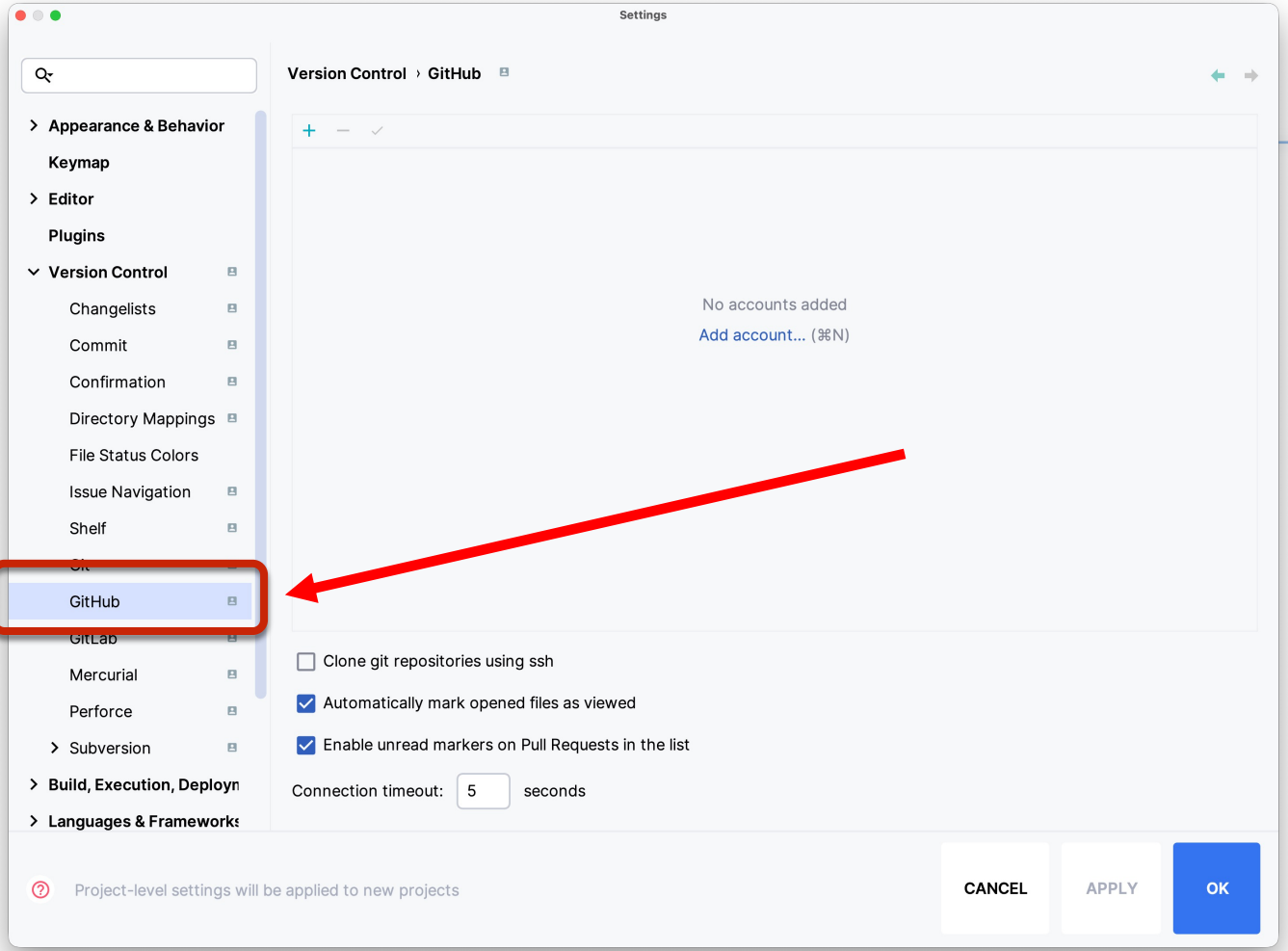
Select the Settings Option



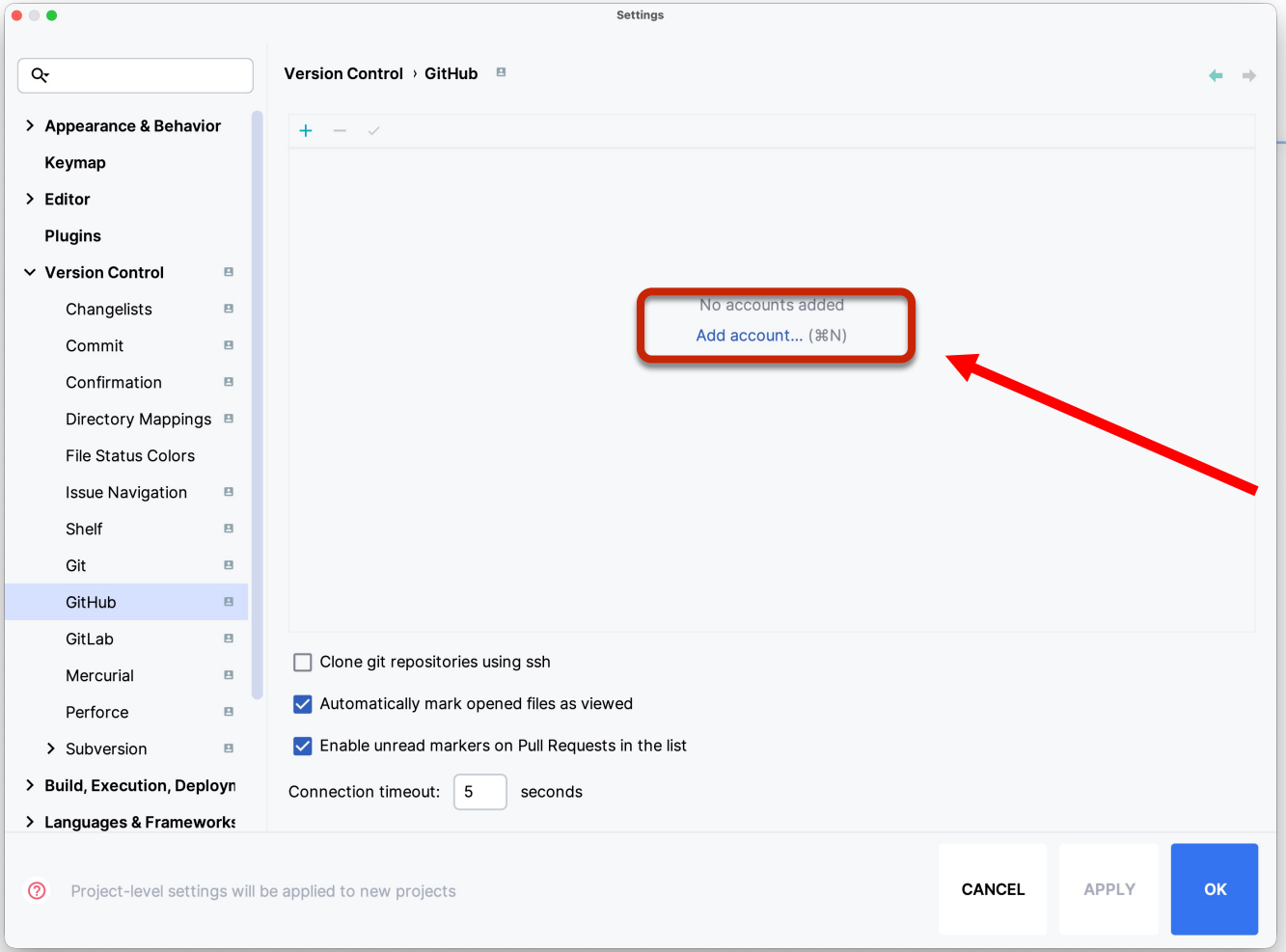
Choose Settings



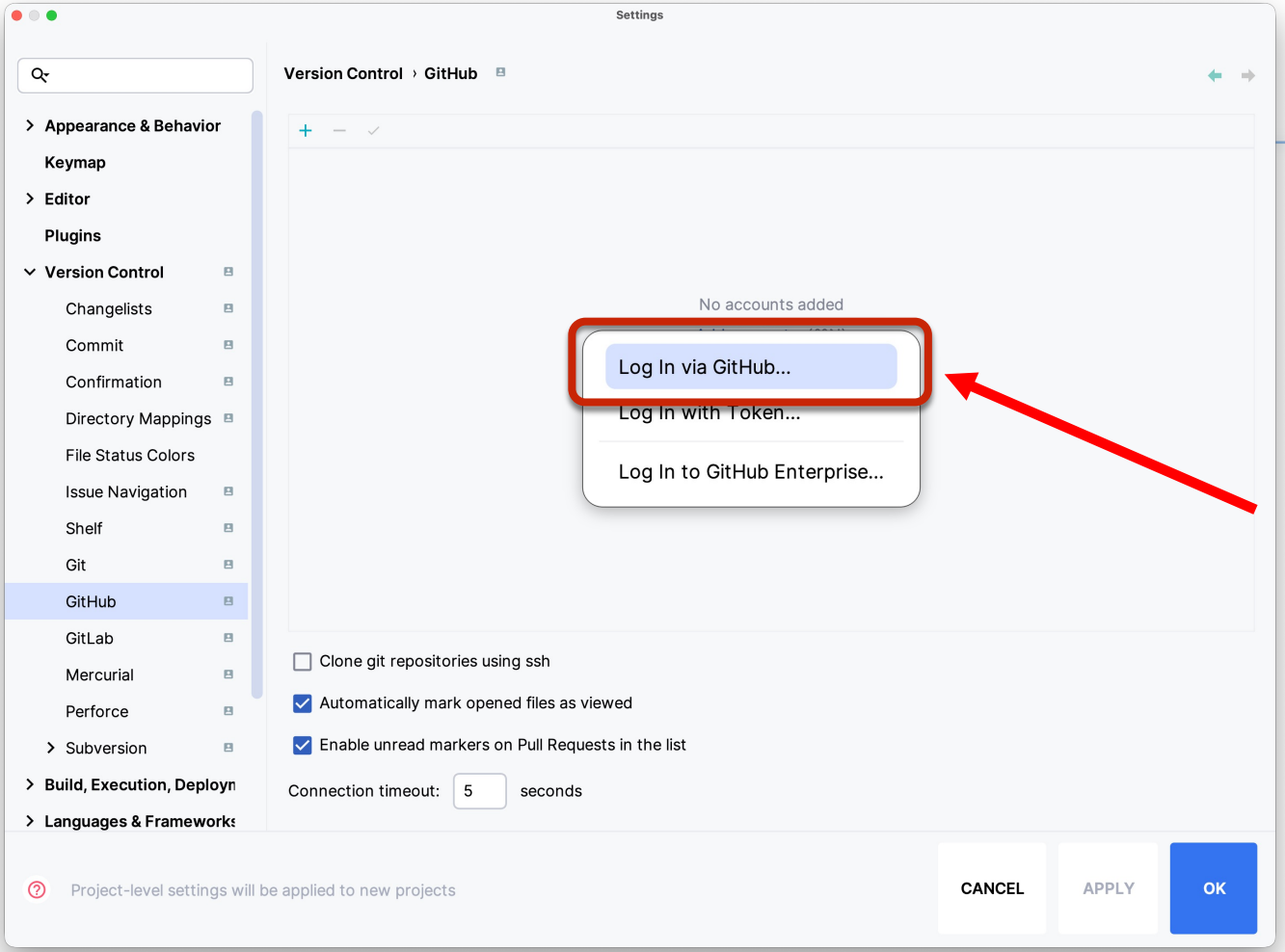
Choose Version Control



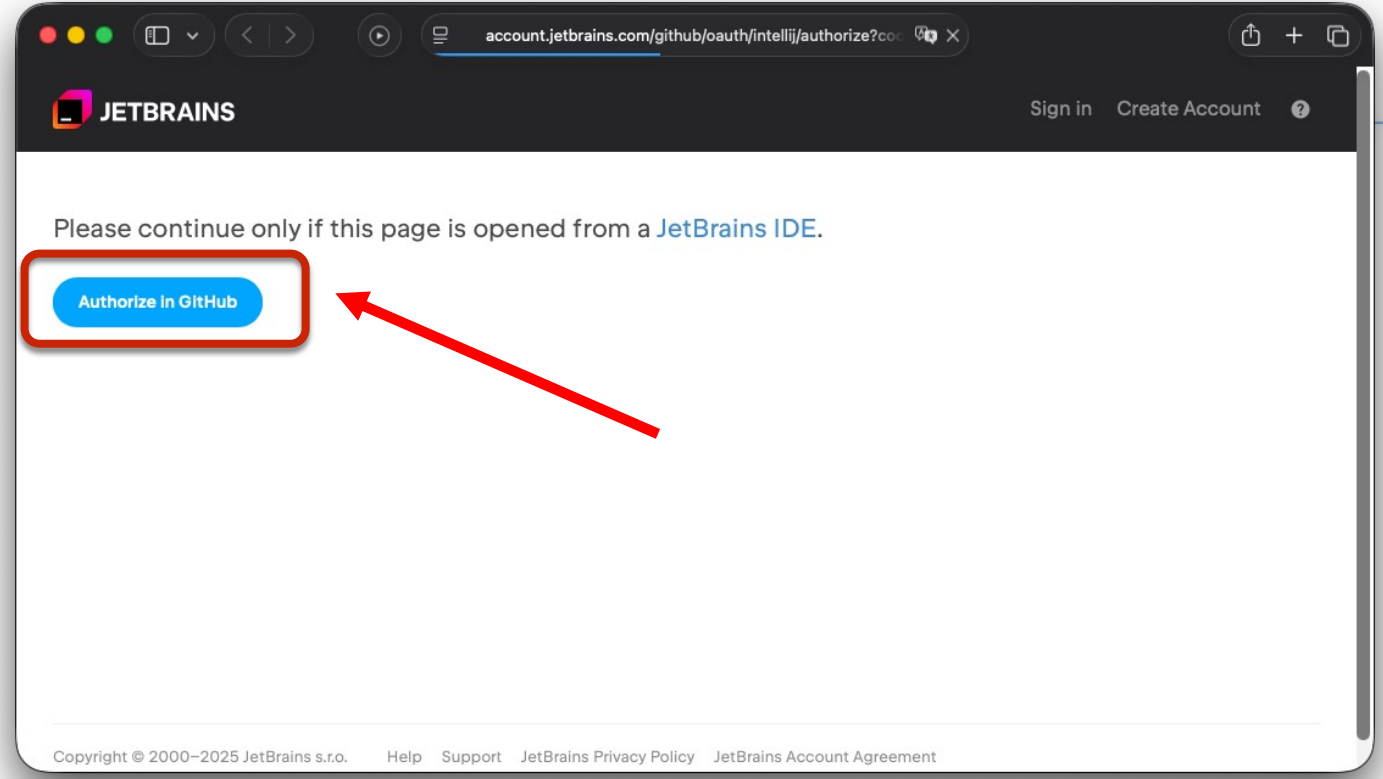
Choose Github



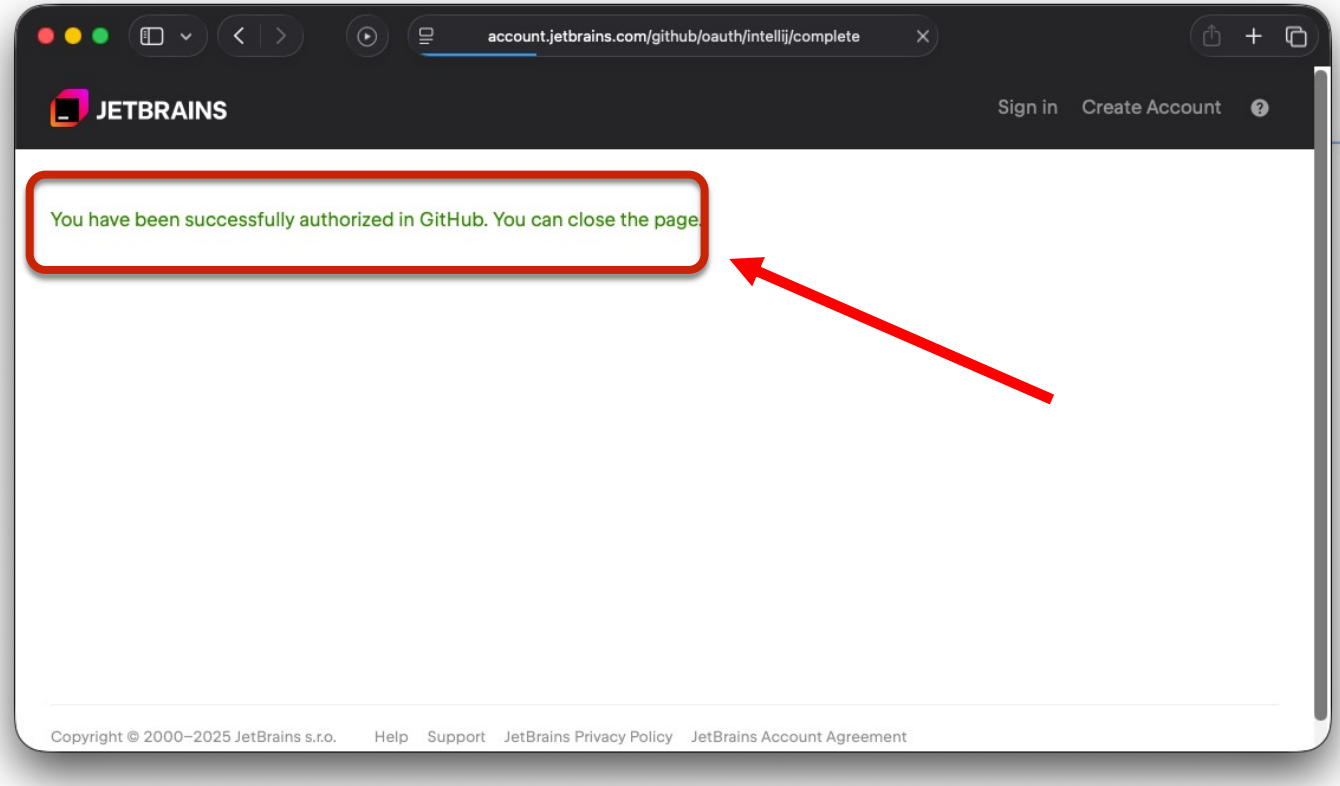
Choose Add account



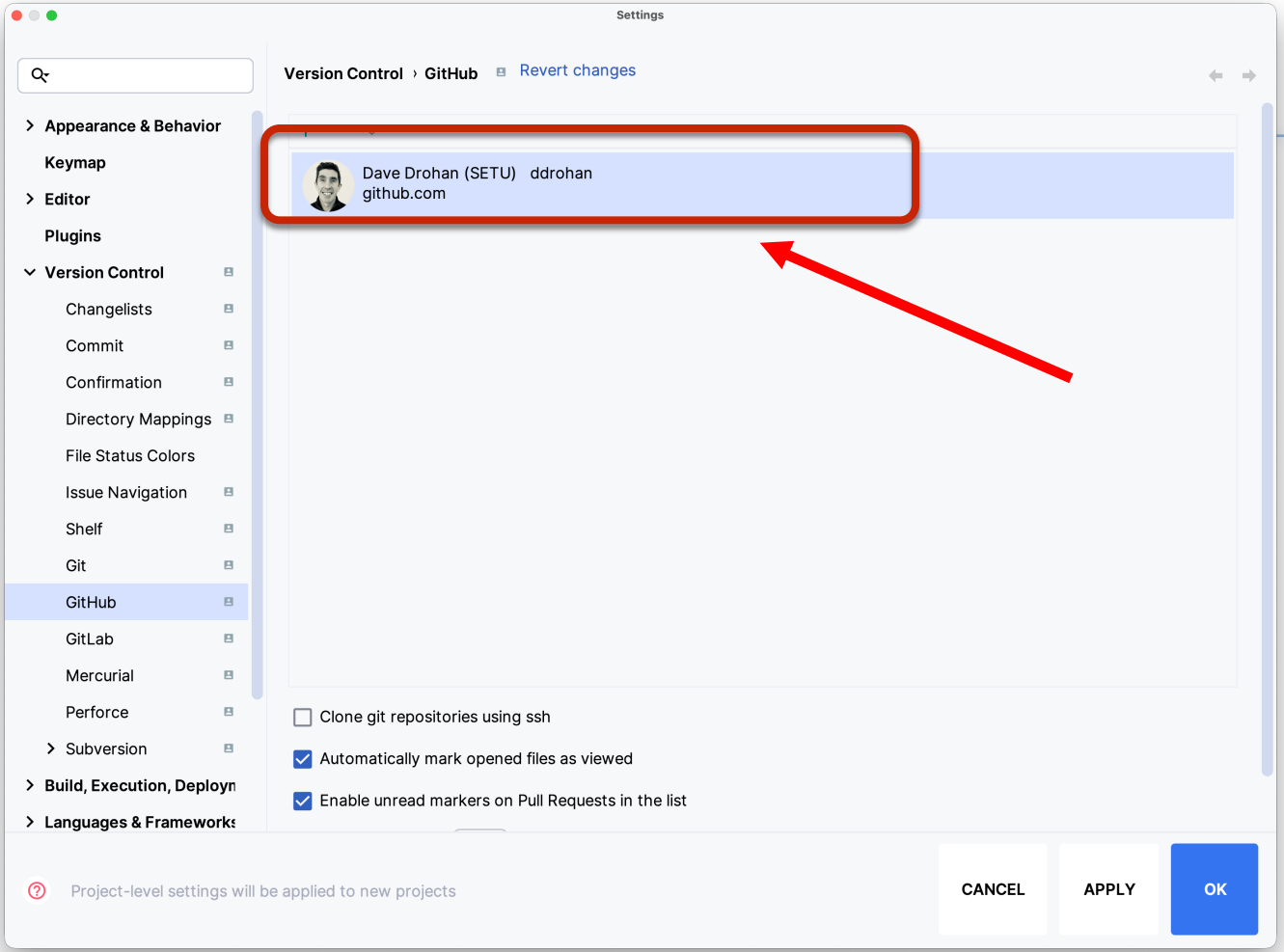
Choose Log In via Github



You will be sent to the jetbrains site so Authorise in Github

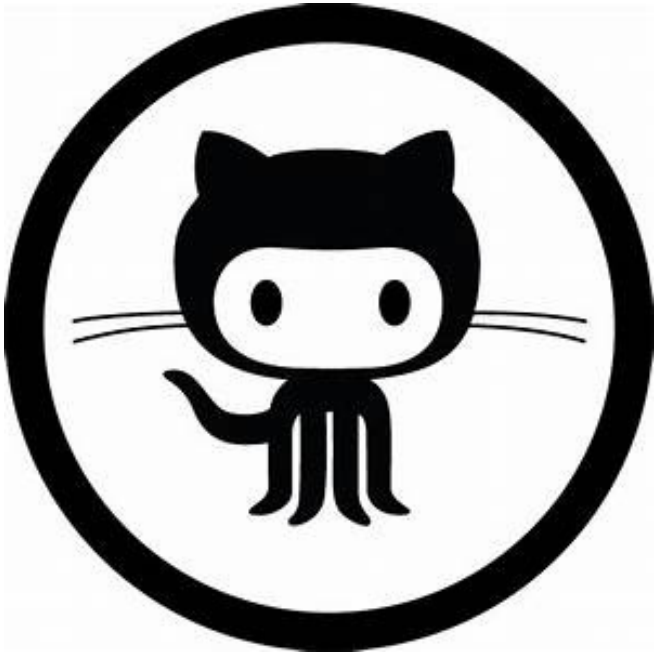


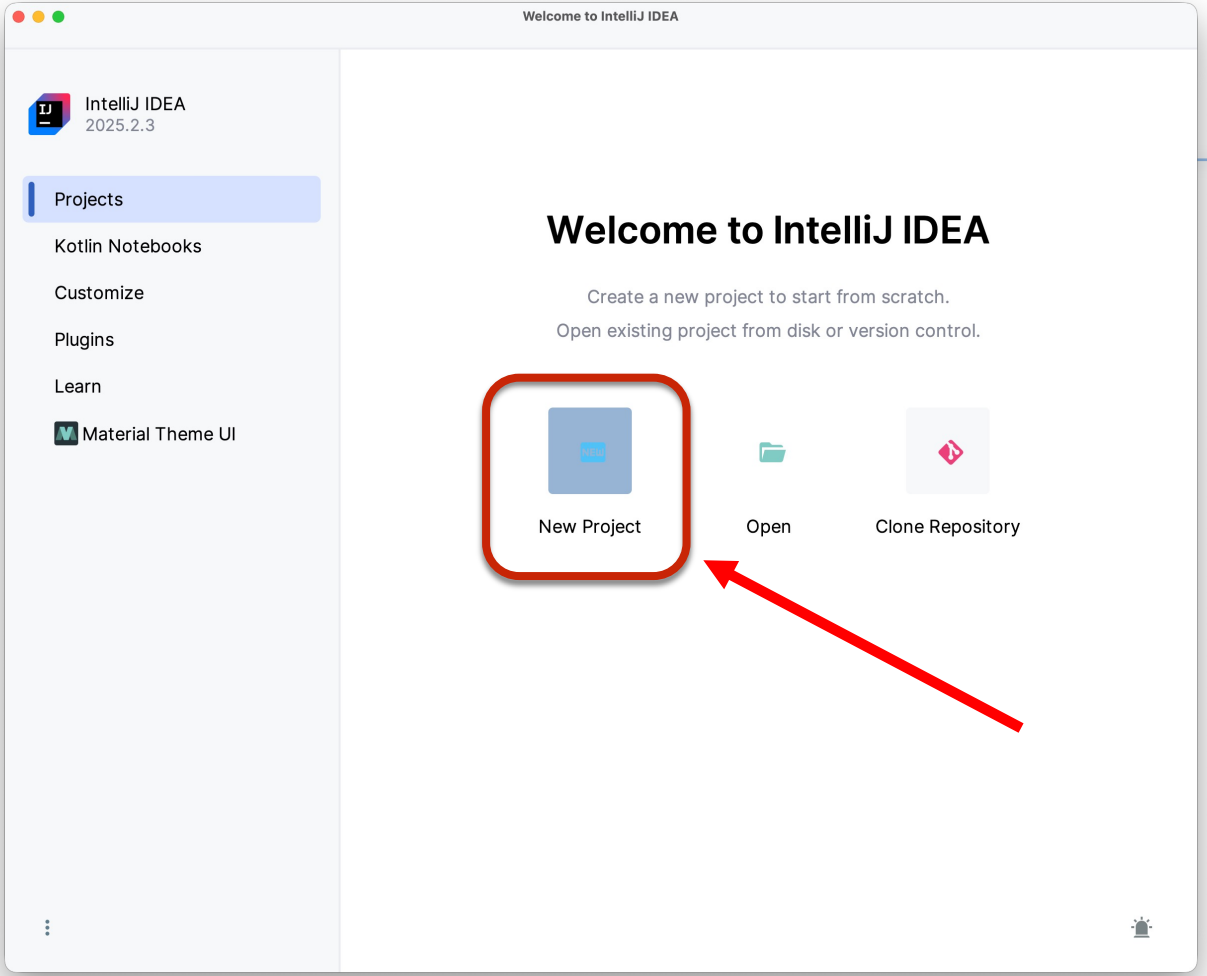
When you get this message return to Intellij



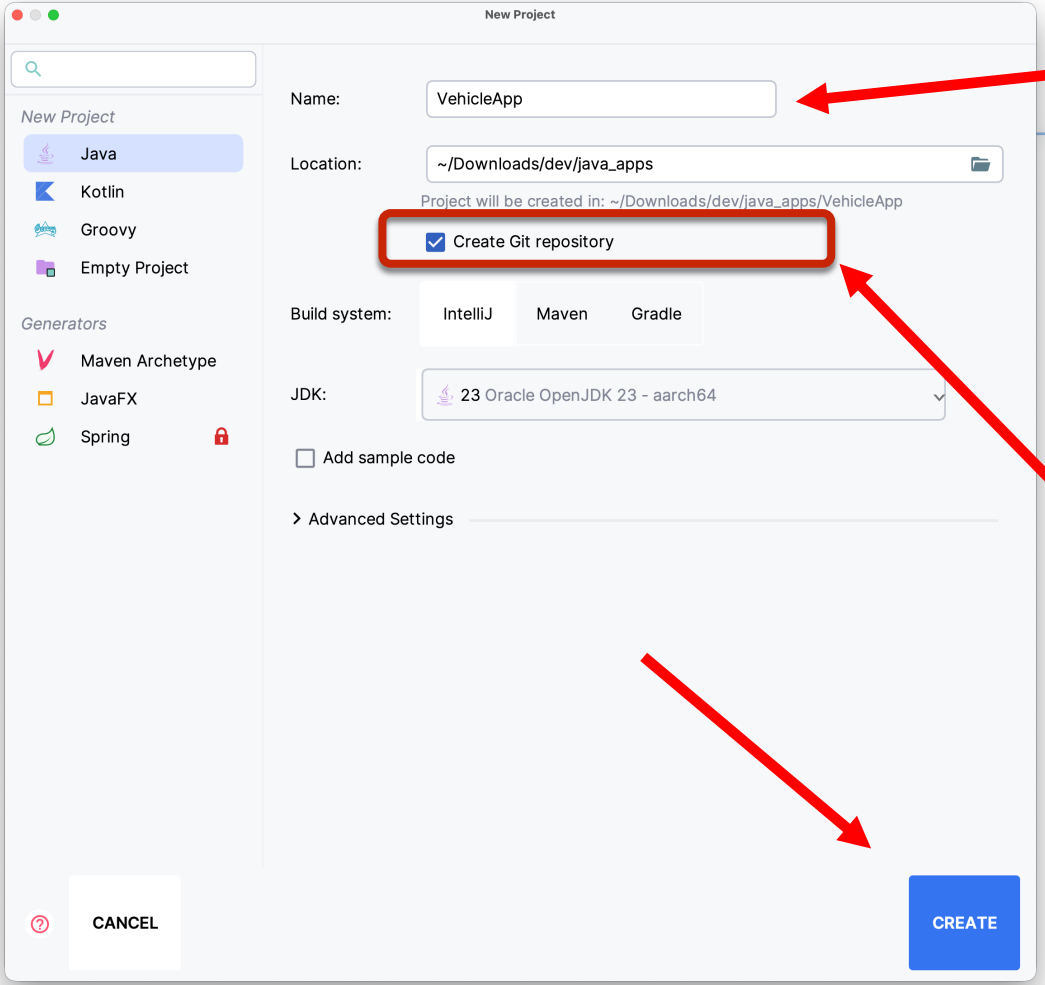
Your Github account is now connected to Intellij

Step 3 – Create your Intellij Java Project

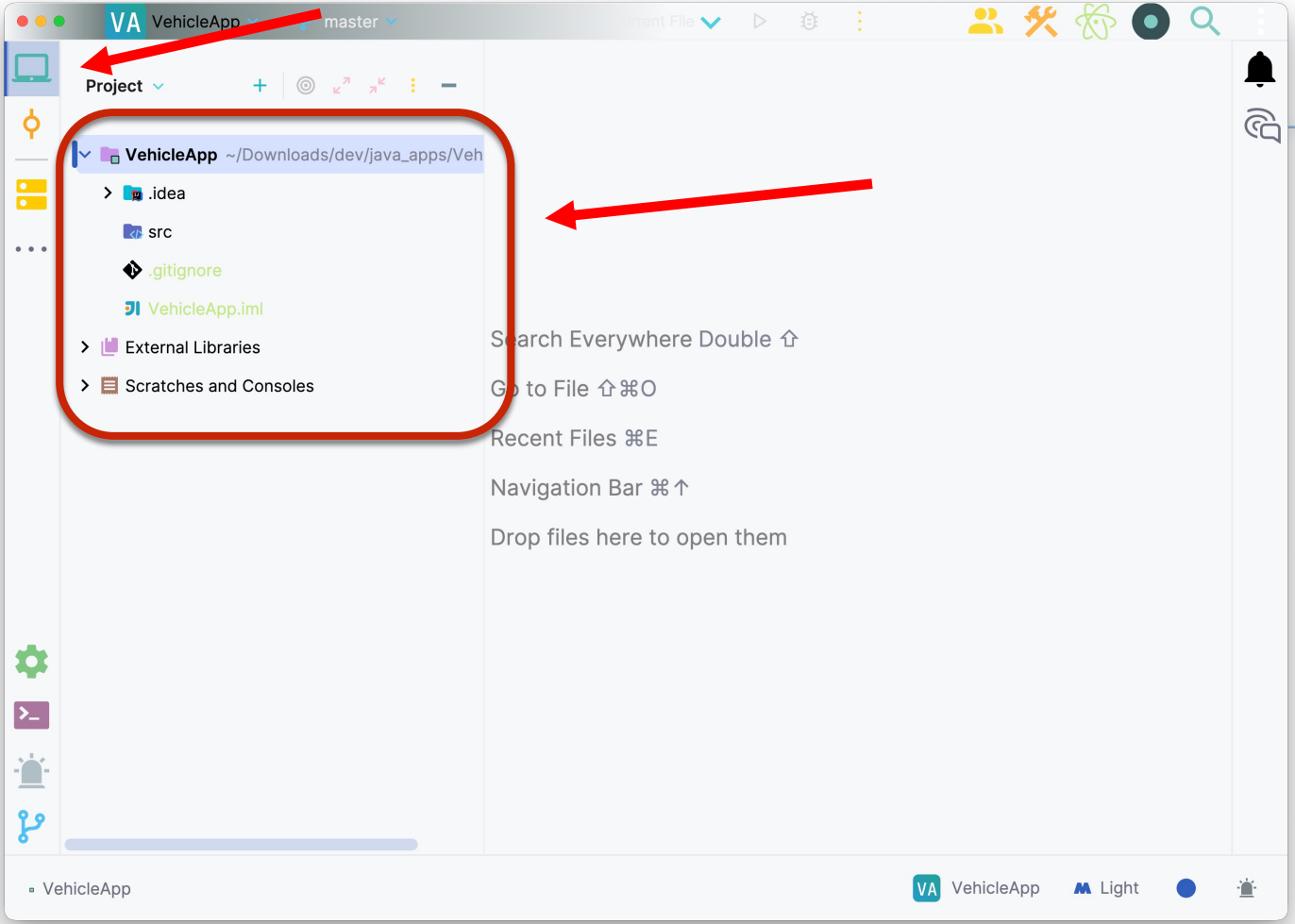




Select ‘New Project’

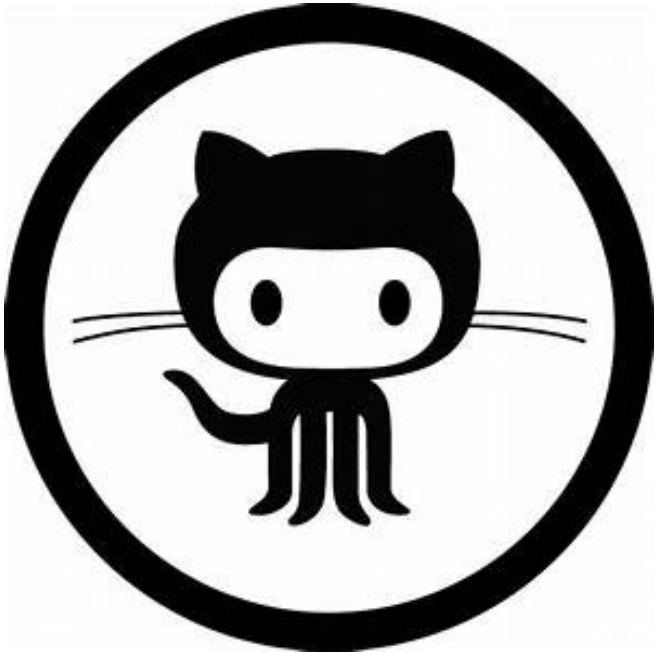


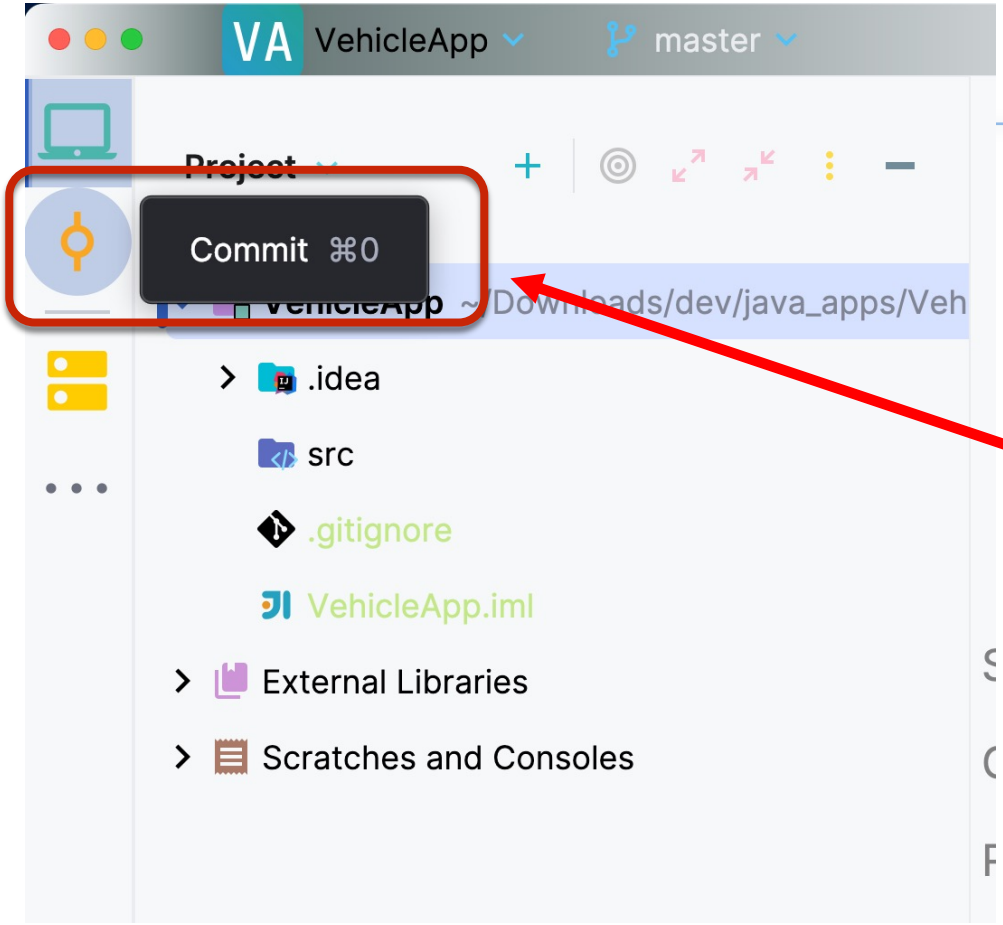
Name oyur project but also select Create Git repository and Create



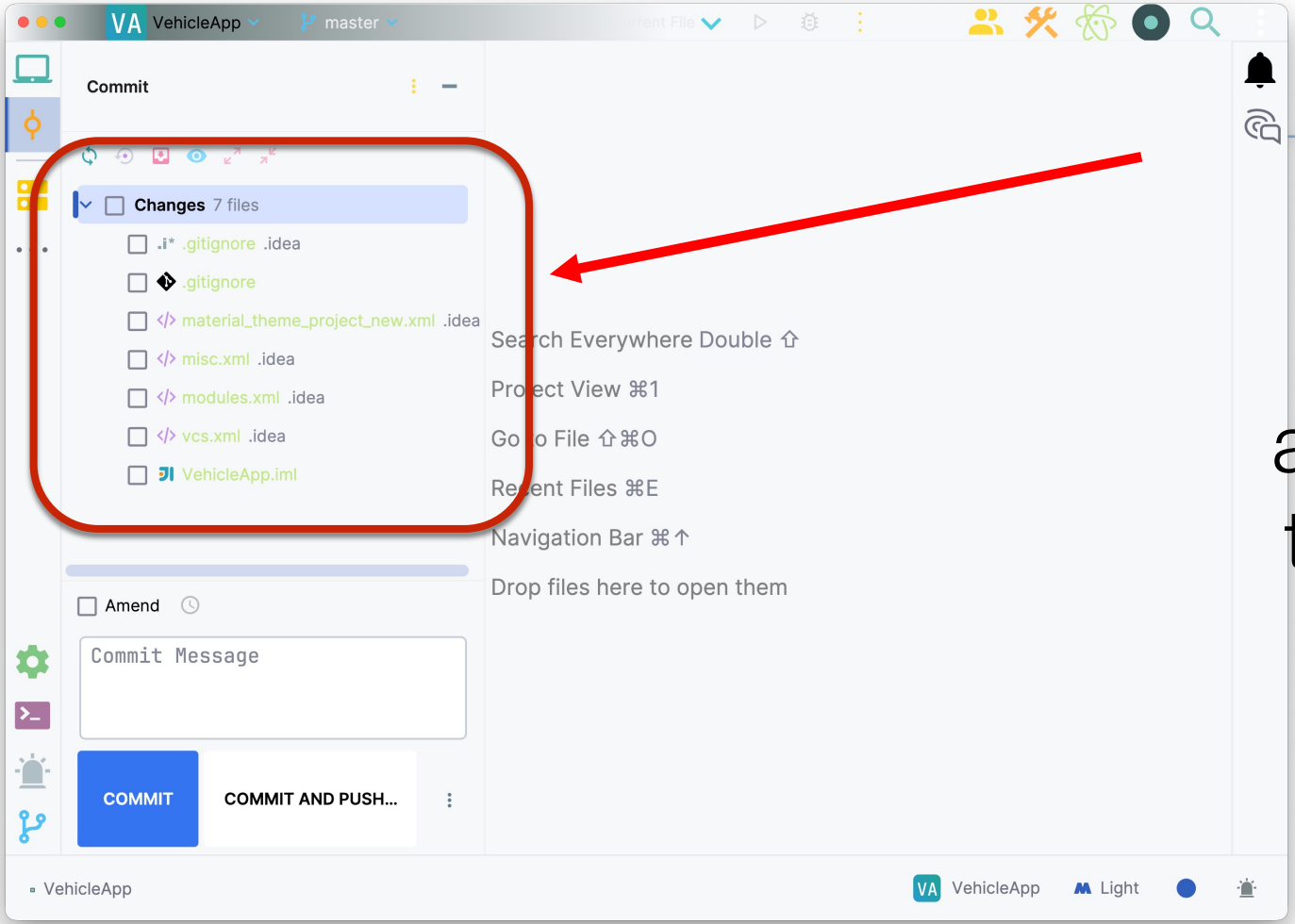
Your new project in the ‘Project’ Tab

Step 4 – Commit/Add your Project to Github

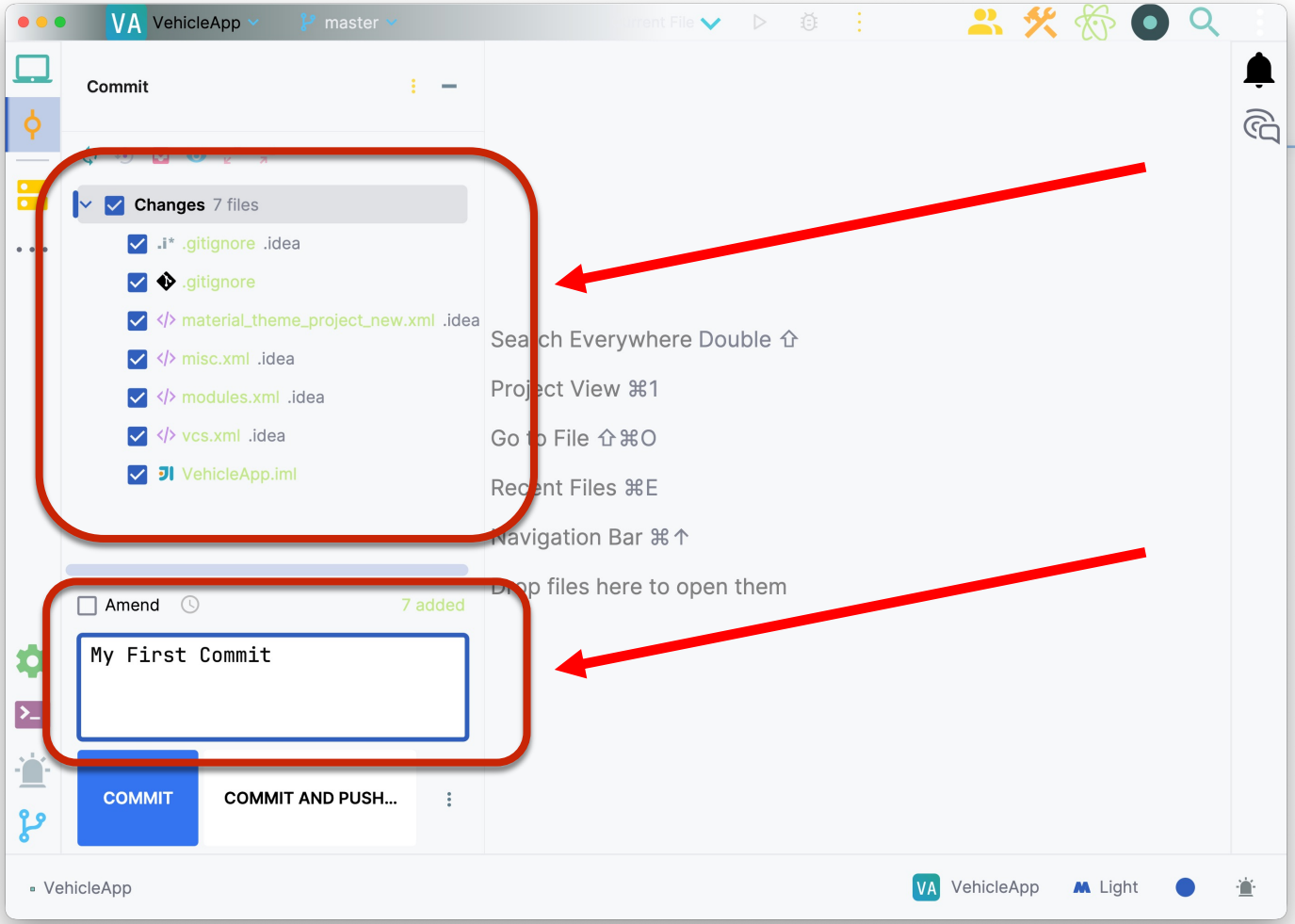




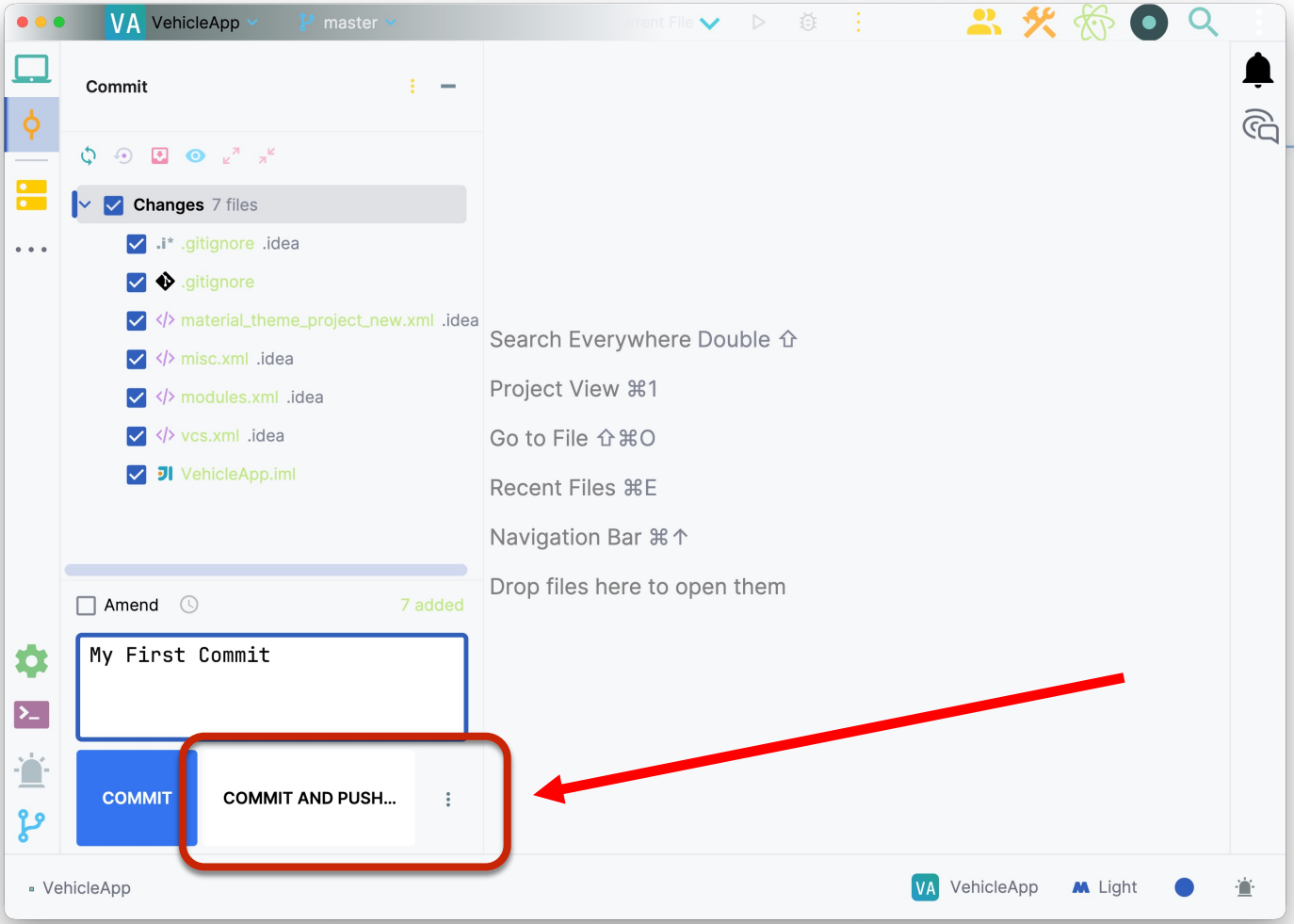
Select the ‘Commit’ Tab



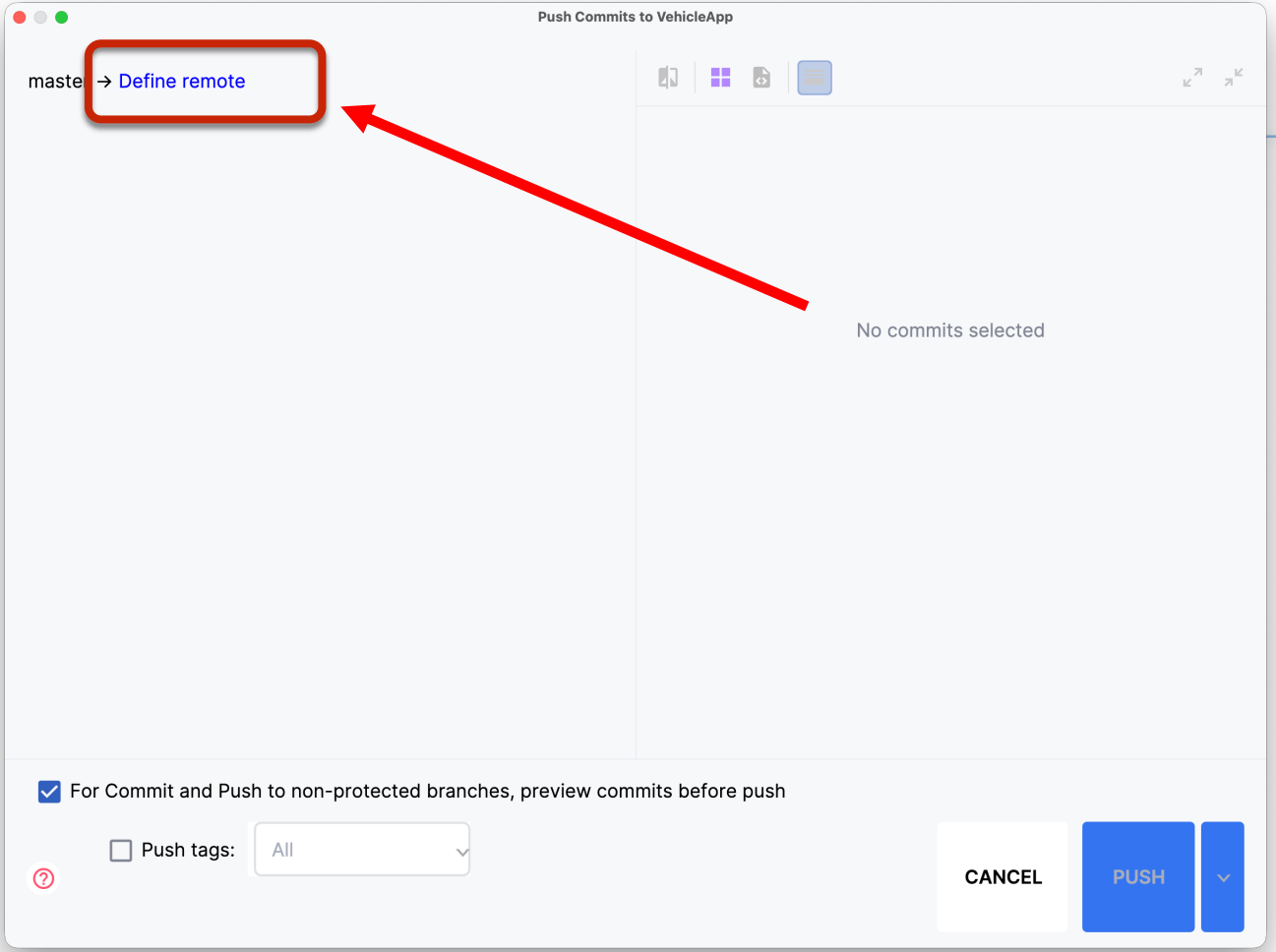
This shows any ‘Changes’ to files in your project



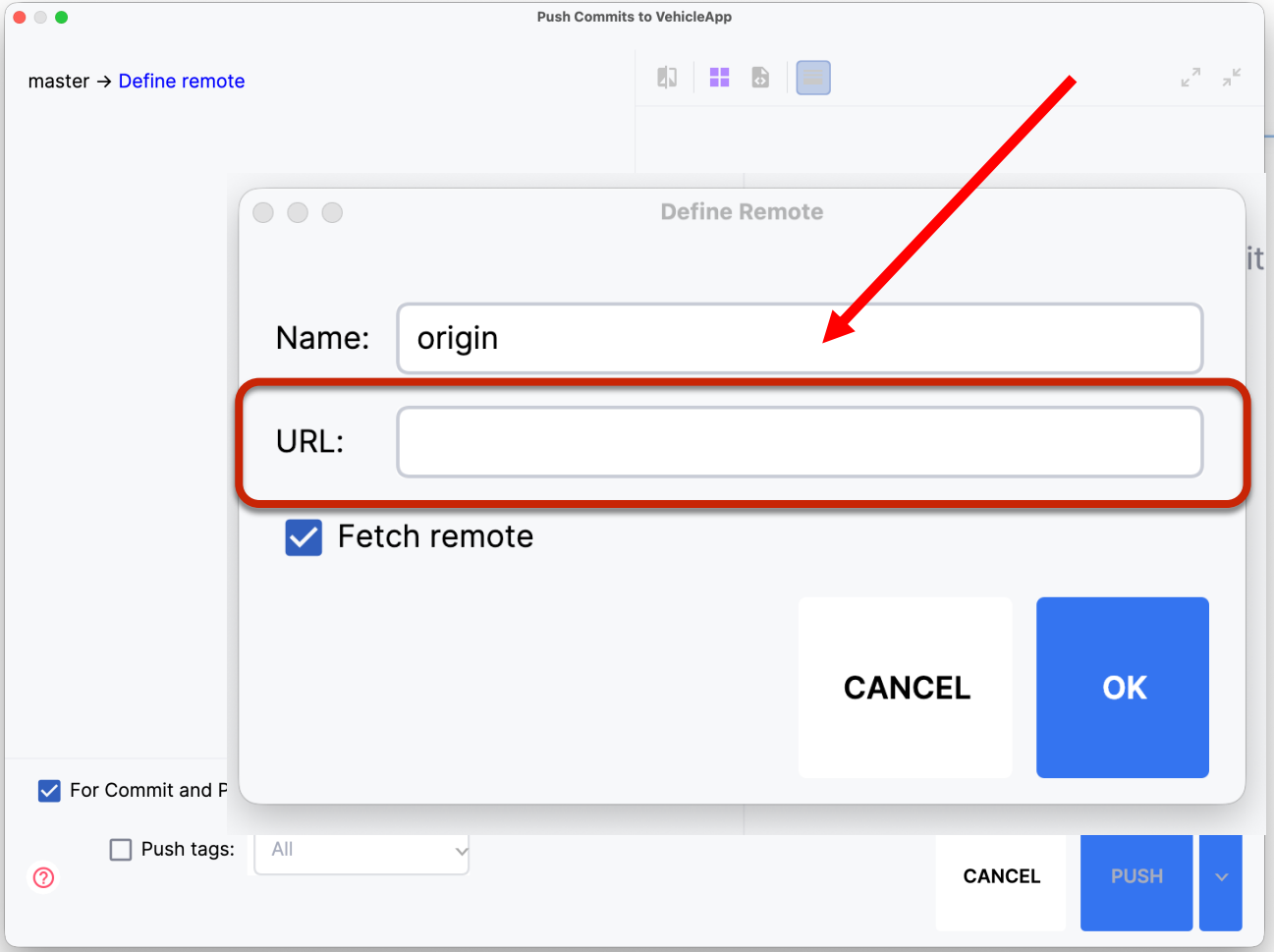
Select ALL Files to ‘commit’ and add a message



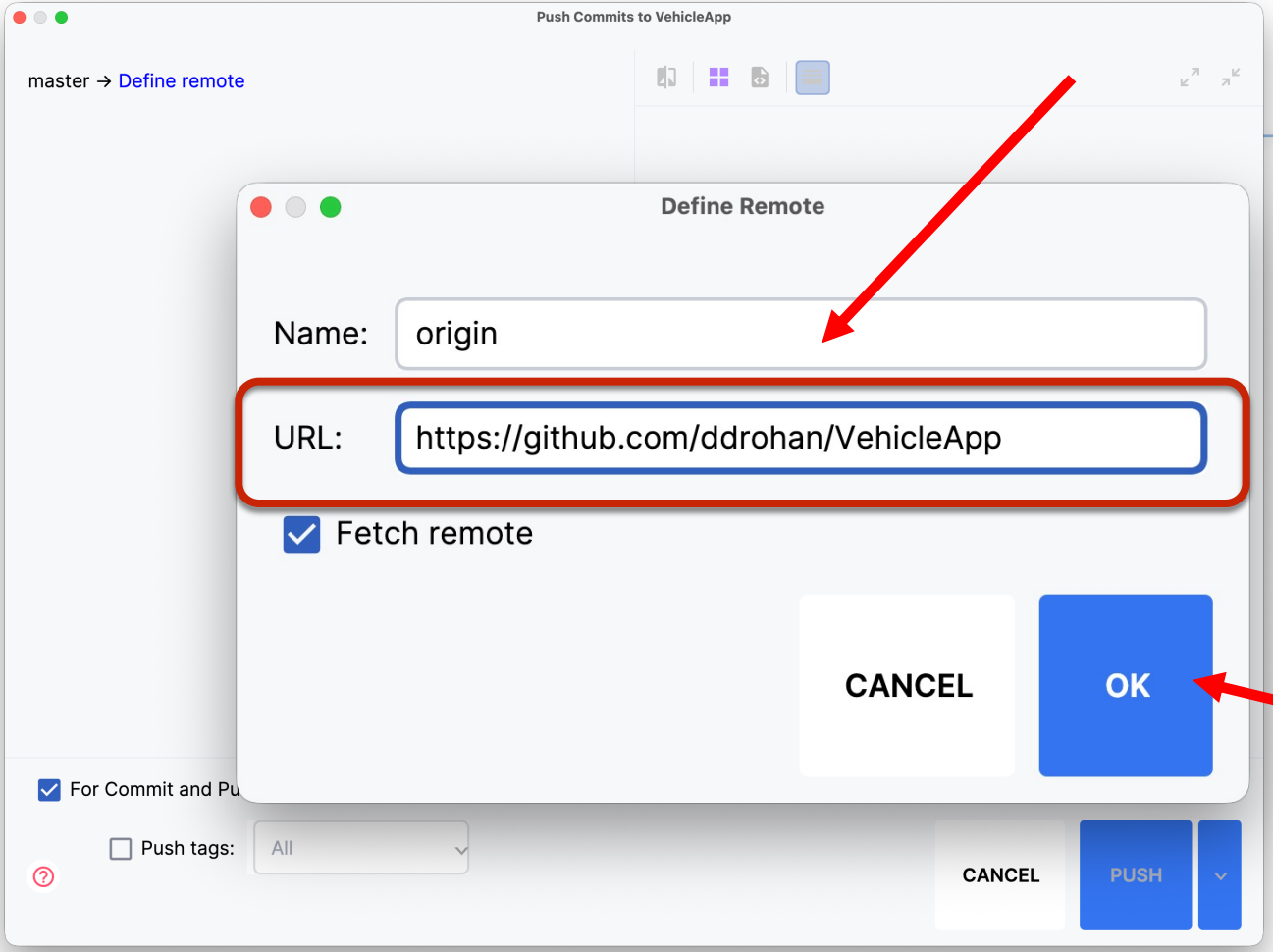
IMPORTANT
Select ‘Commit and Push’



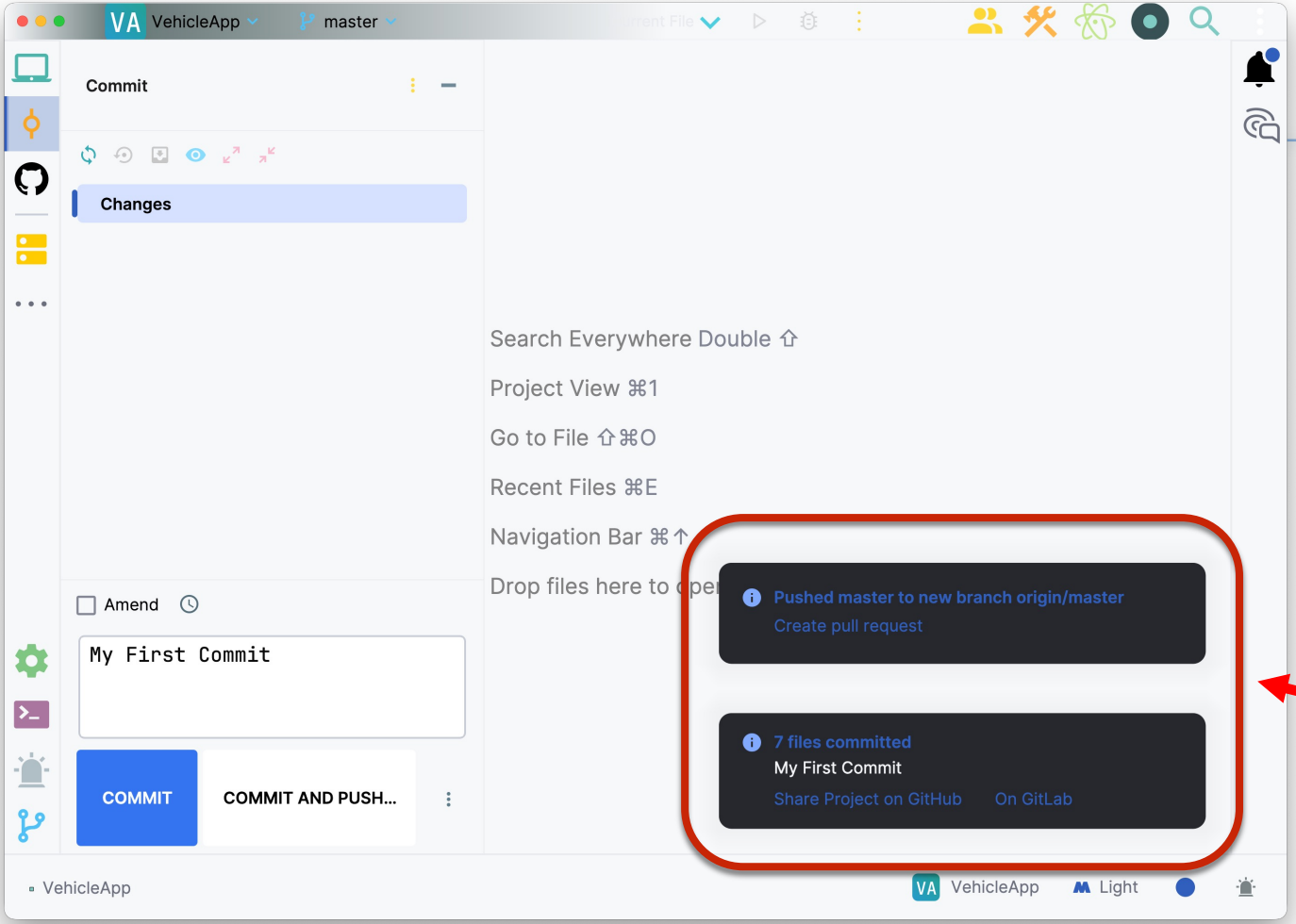
Select ‘Define remote’



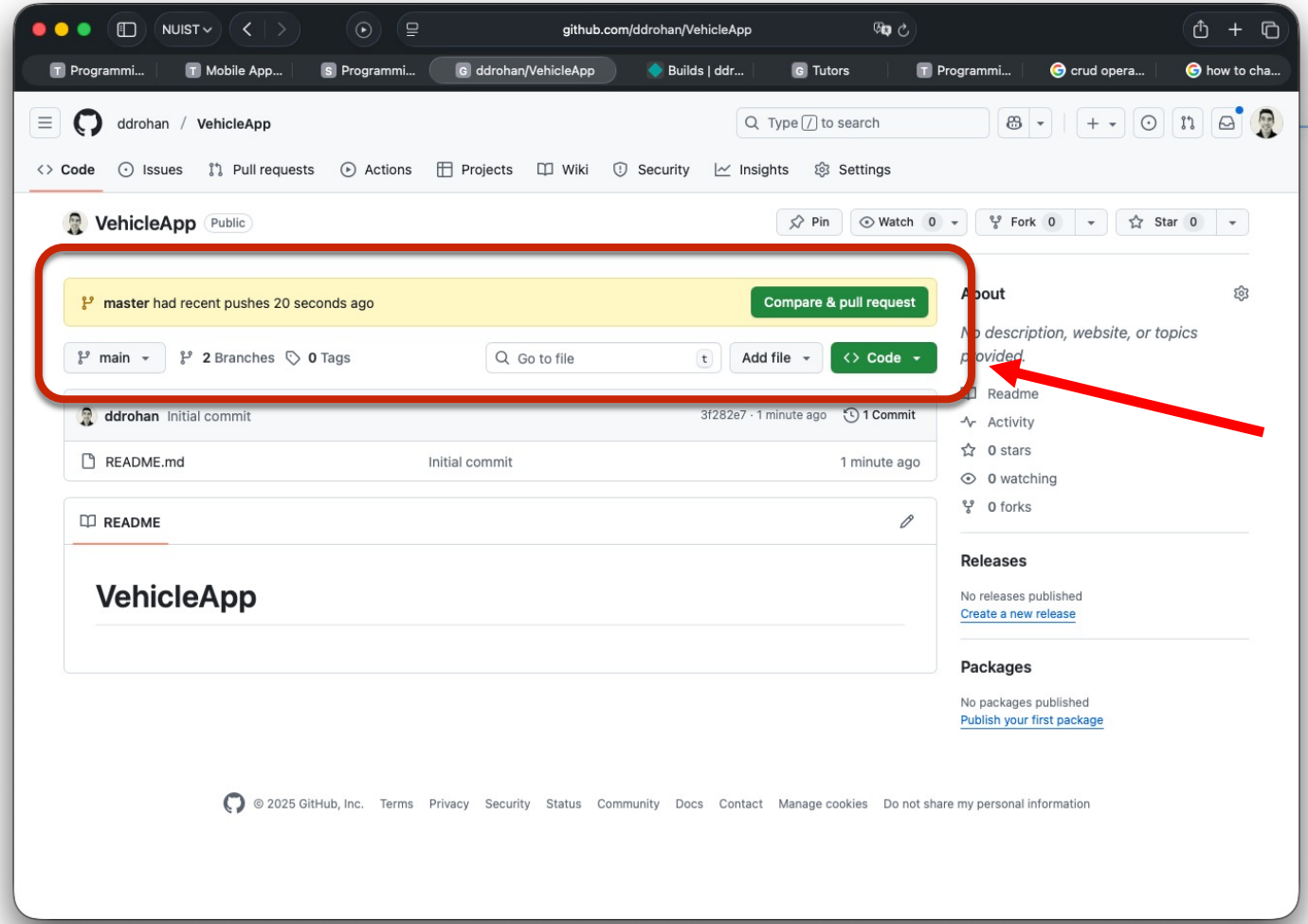
Enter the URL Link from your Github Repository


Push Commits to VehicleApp
master → Define remote
Define Remote
Name:
origin
URL:
https://github.com/ddrohan/VehicleApp
Fetch remote
CANCEL
OK
For Commit and Pu
Push tags:
All
CANCEL
PUSH

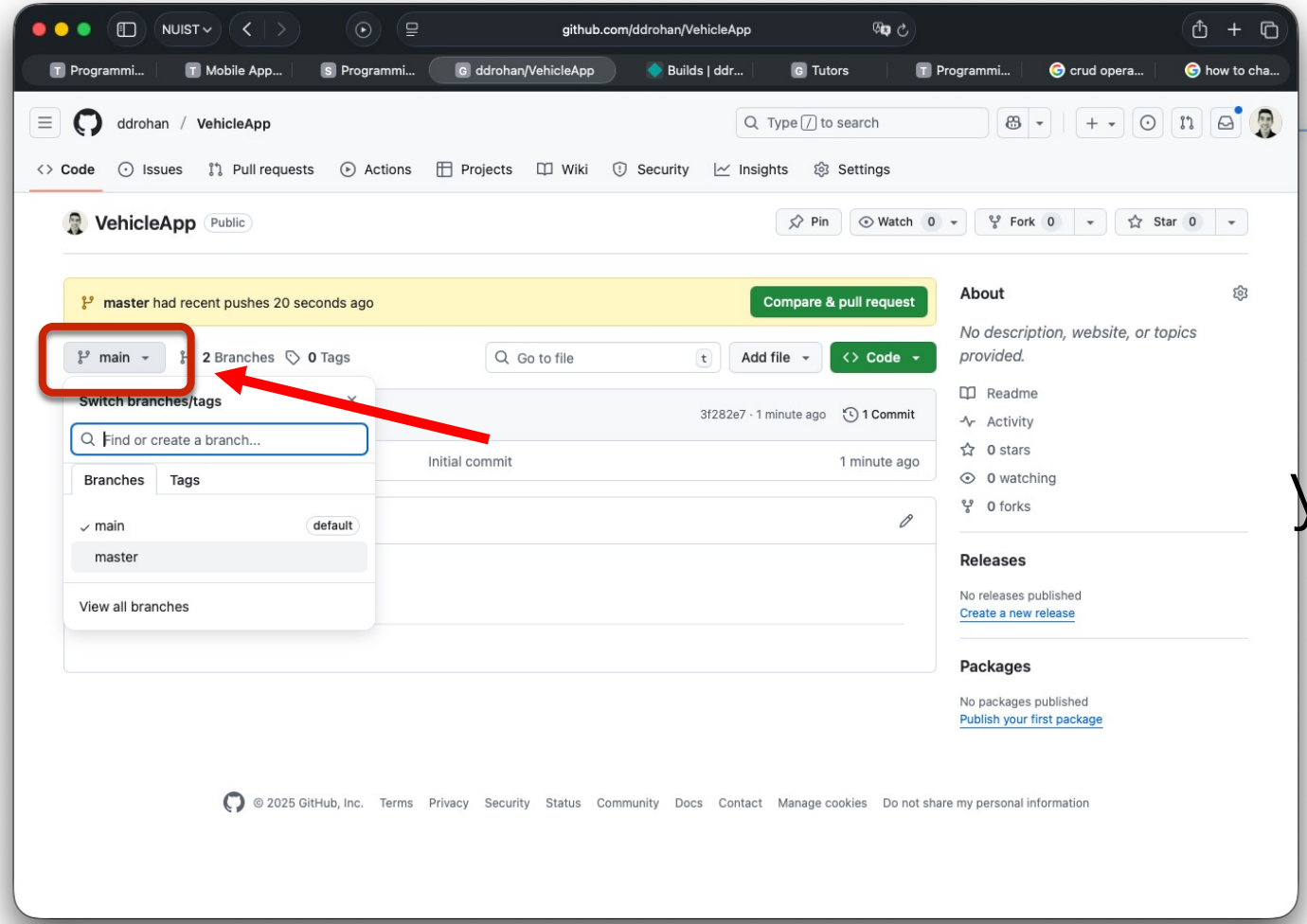
Select ‘OK’



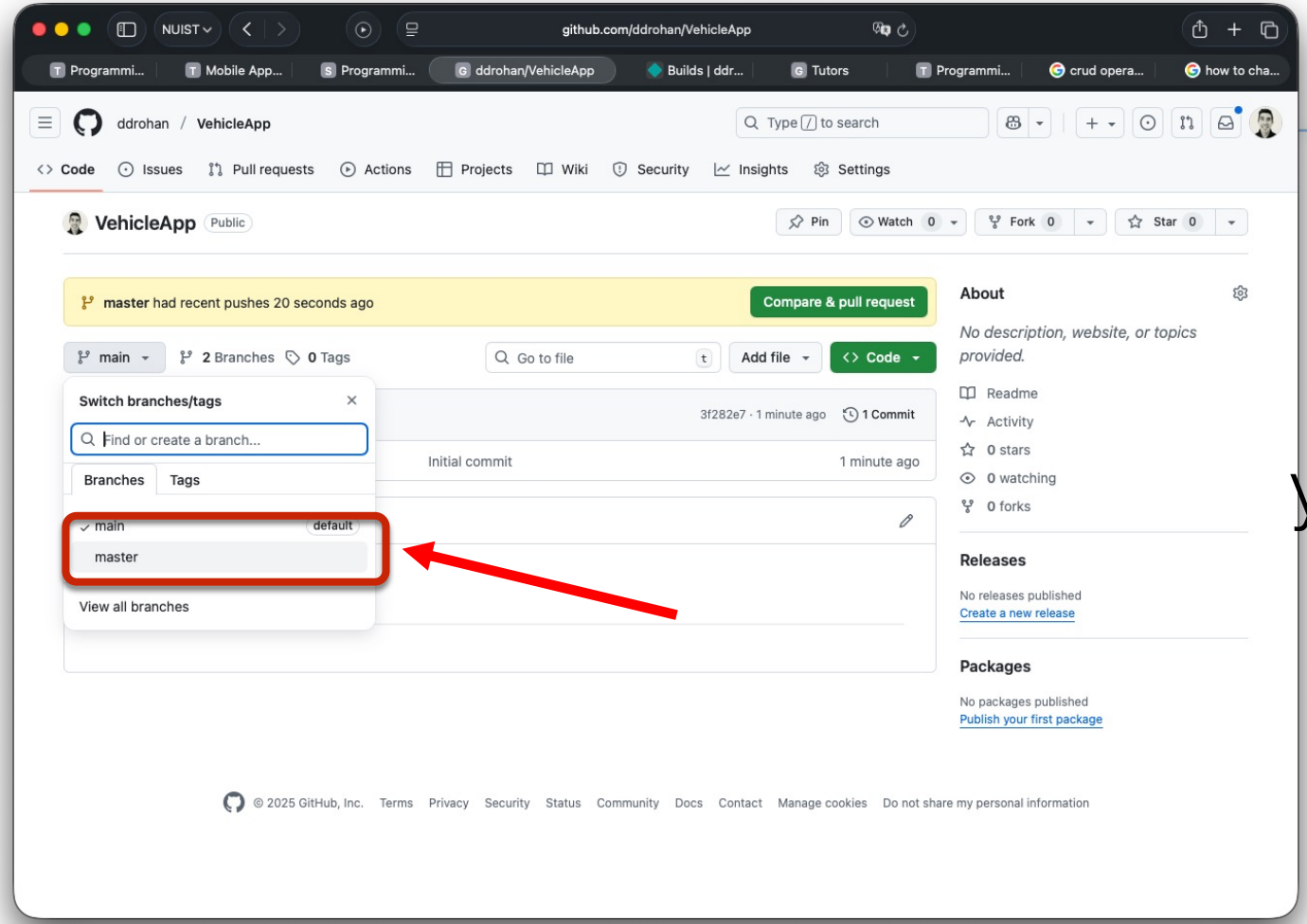
This will be your Initial Commit



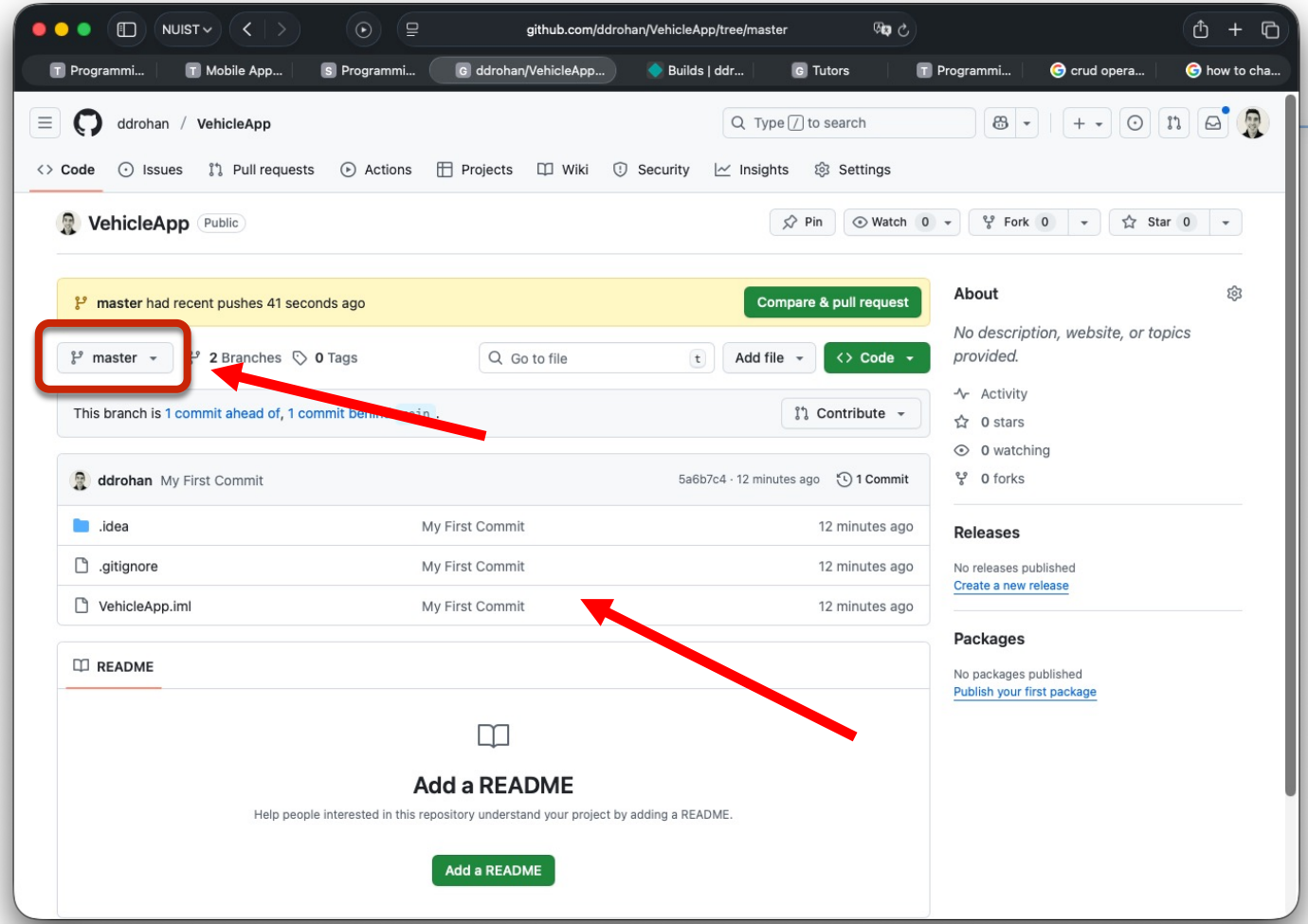
Confirm the changes on your Github Repository



Change to your ‘master’ Branch

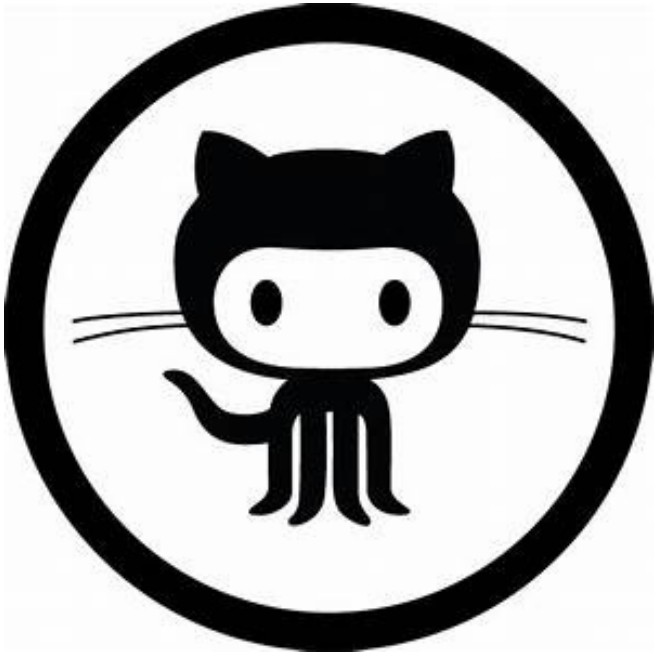


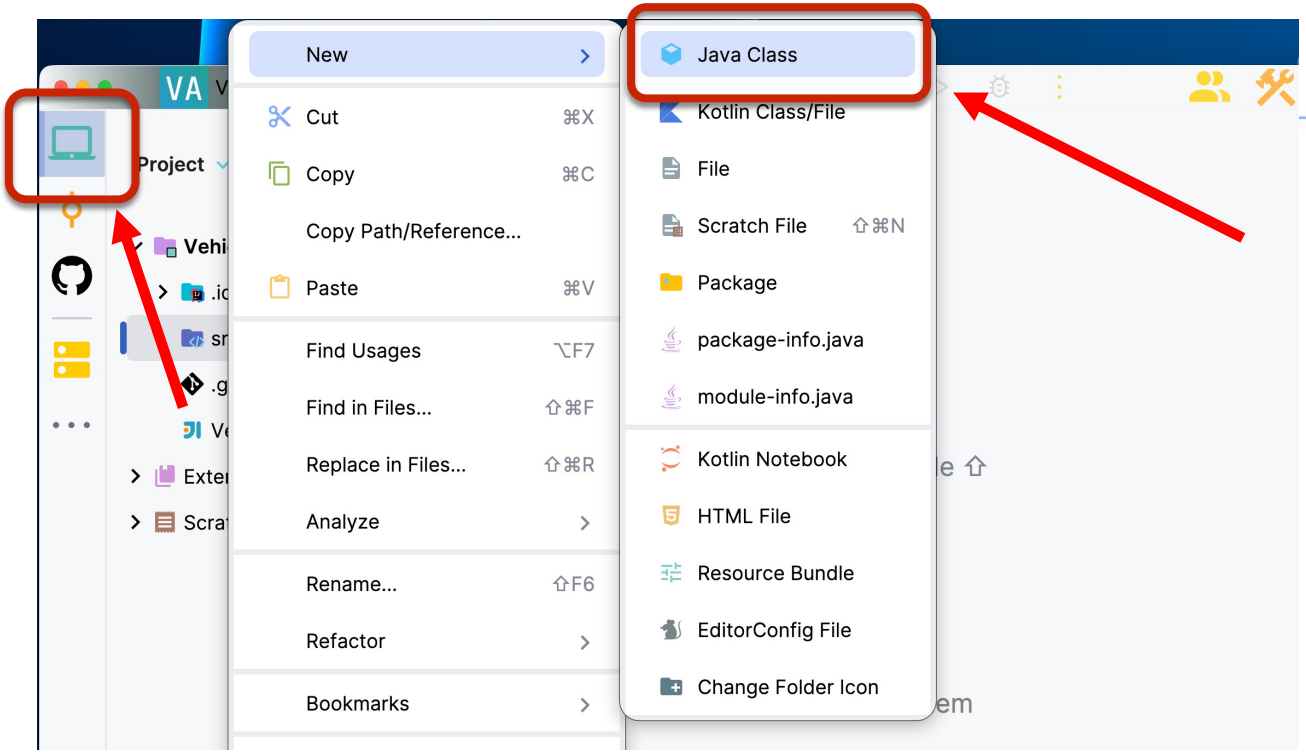
Change to your ‘master’ Branch



Your Initial Commit Changes

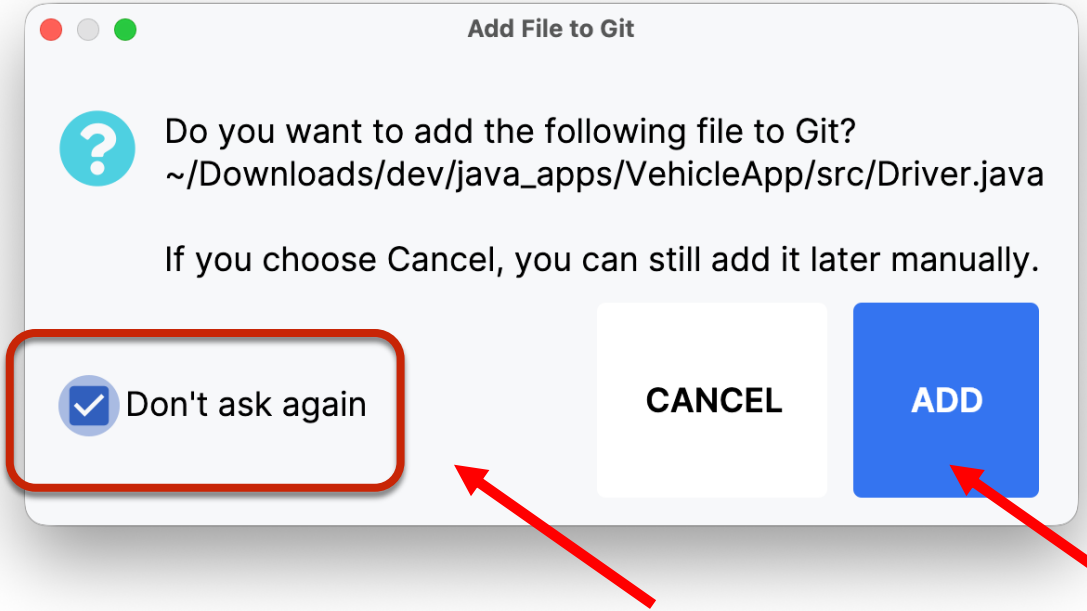
Step 5 – Make changes to your Project





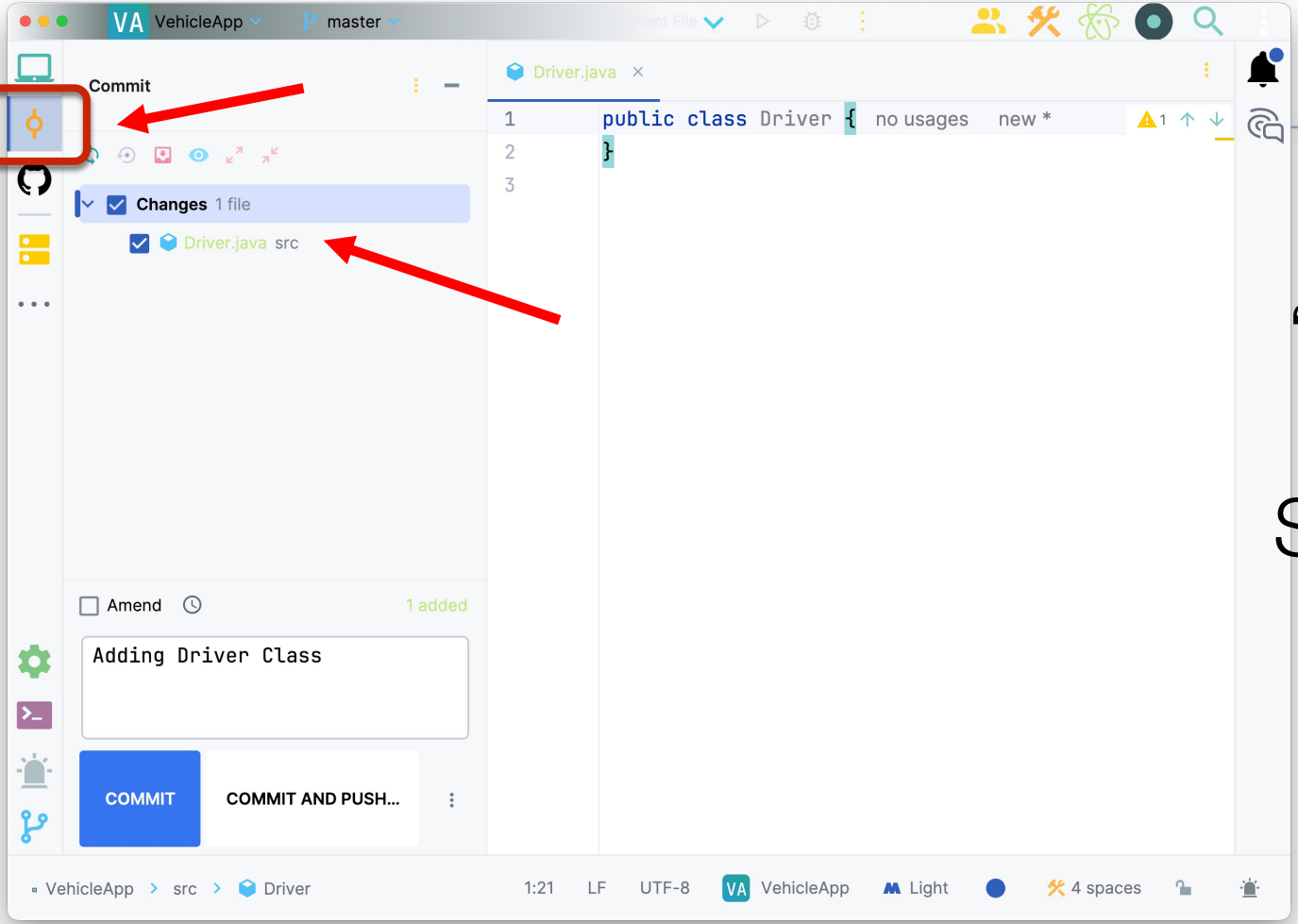
Select your ‘Project’ Tab and Add a new Java Class

(class Driver here)

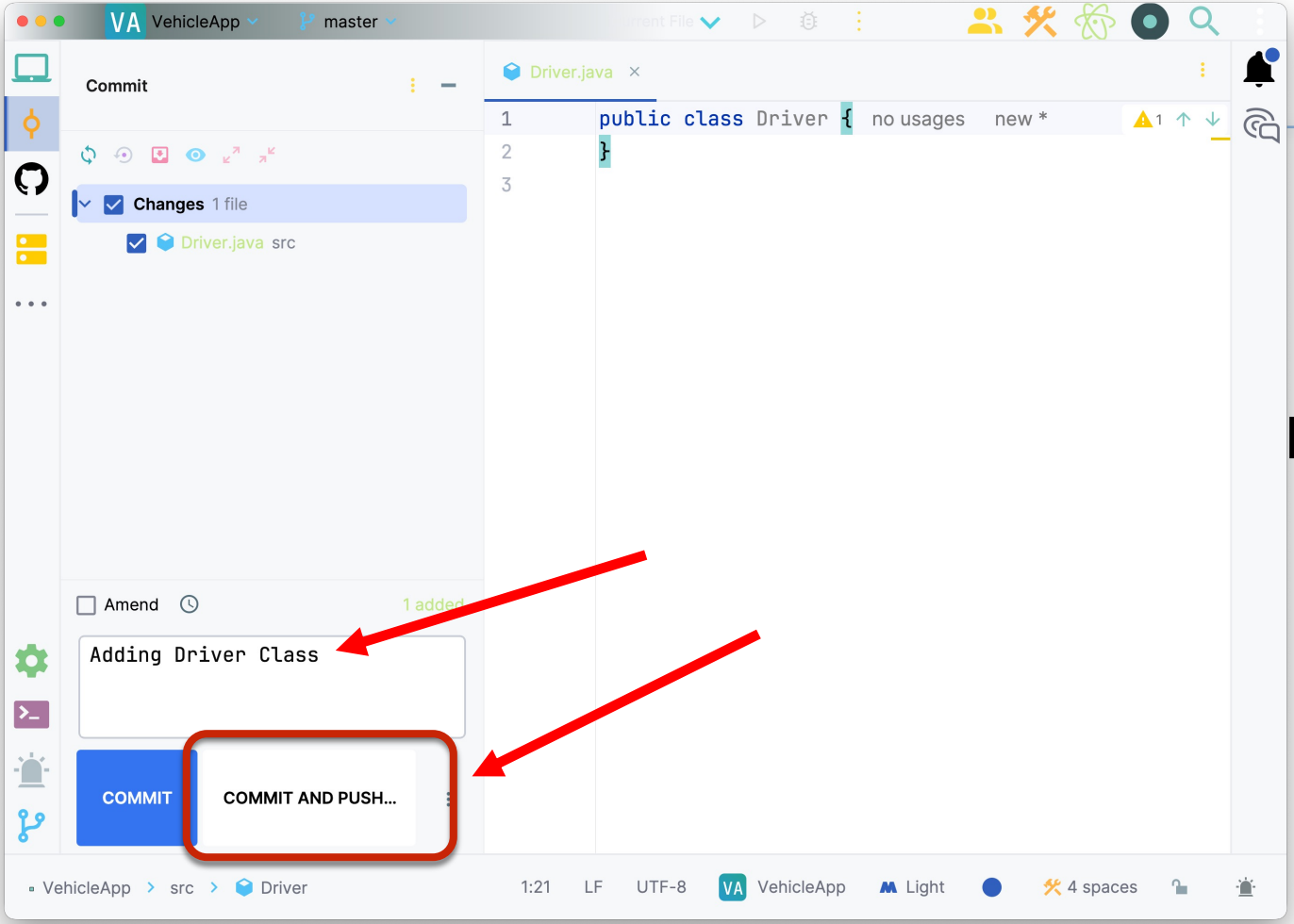


Most of the time you SHOULD add the file to git.

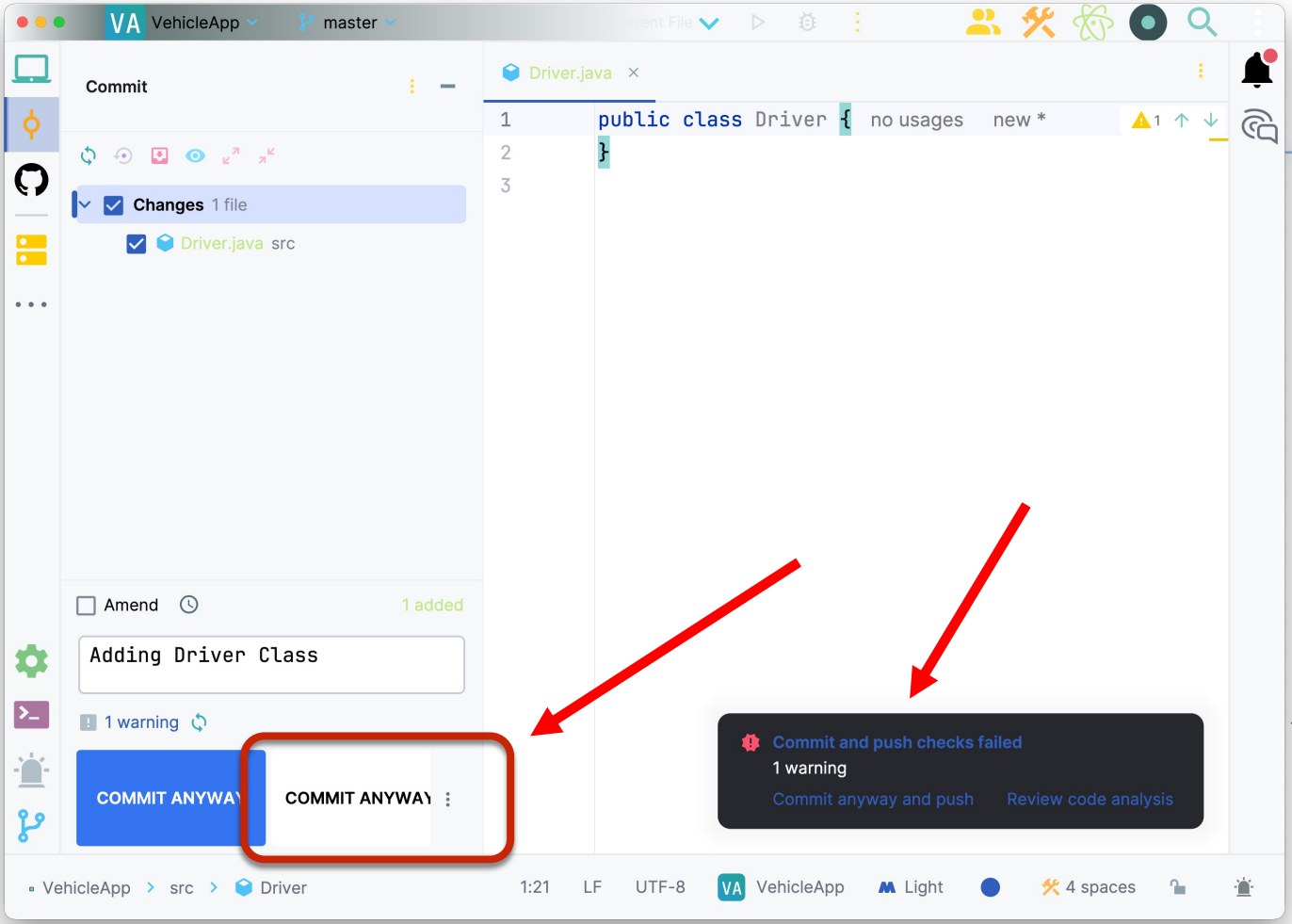
Choose ‘Don’t ask again’ and ‘ADD’



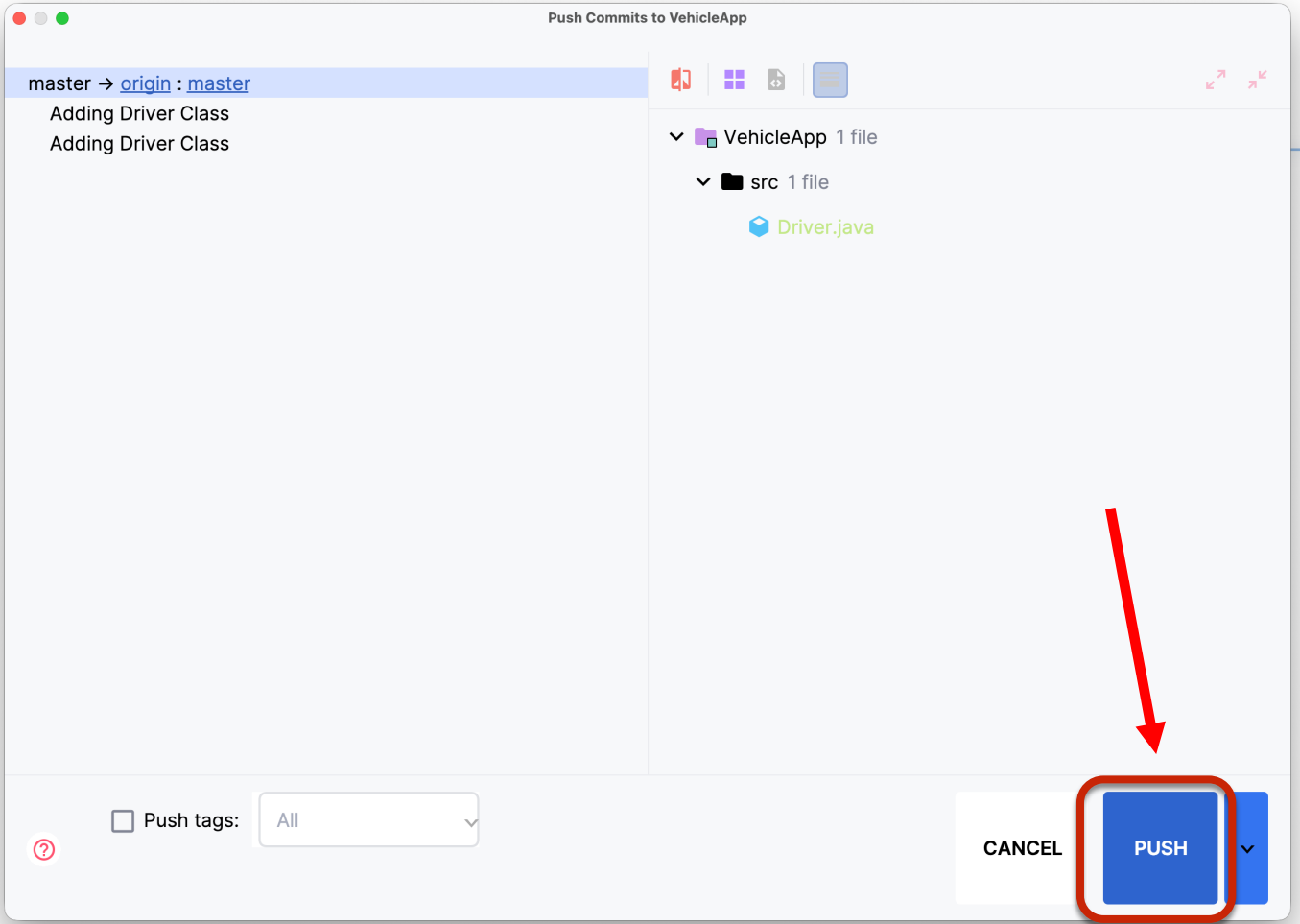
Select ‘Commit’ Tab again and Select the files to commit (Driver.java here)



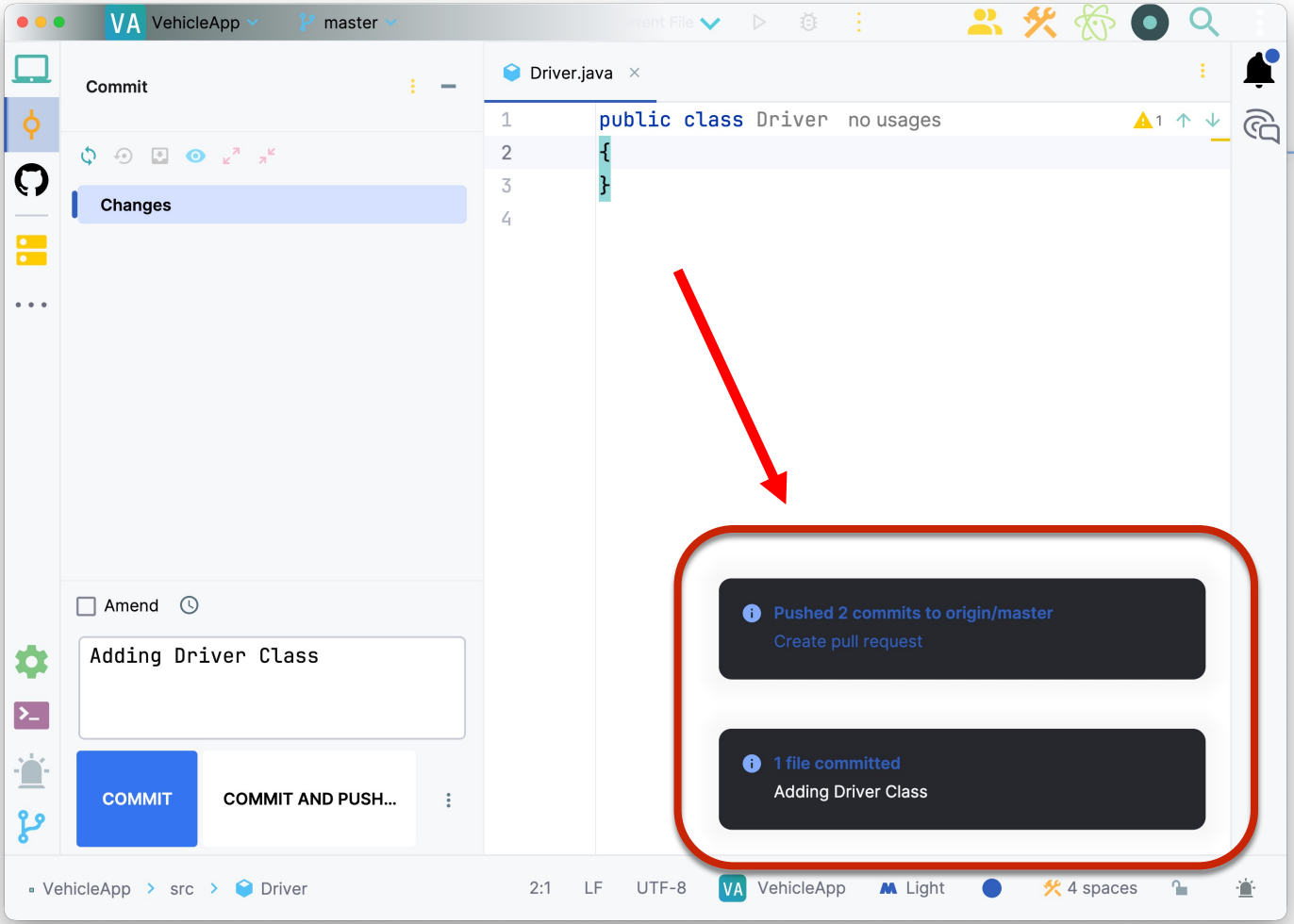
Add a message and ‘Commit and Push’ once again



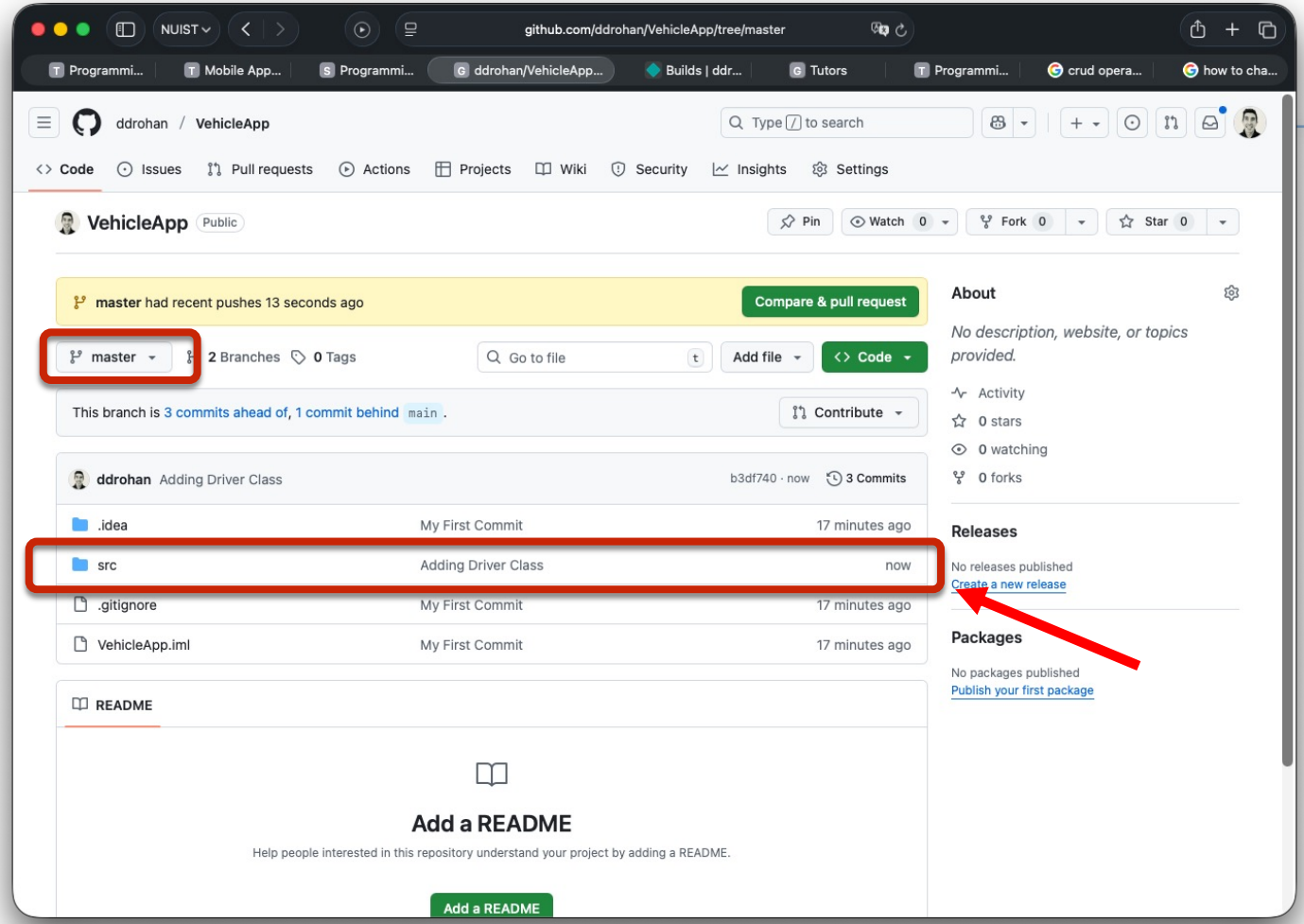
You can ignore the warning so ‘Commit Anyway And Push’


Push Commits to VehicleApp
master → origin : master
Adding Driver Class
Adding Driver Class
VehicleApp 1 file
src 1 file
Driver.java
Push tags:
All
CANCEL
PUSH

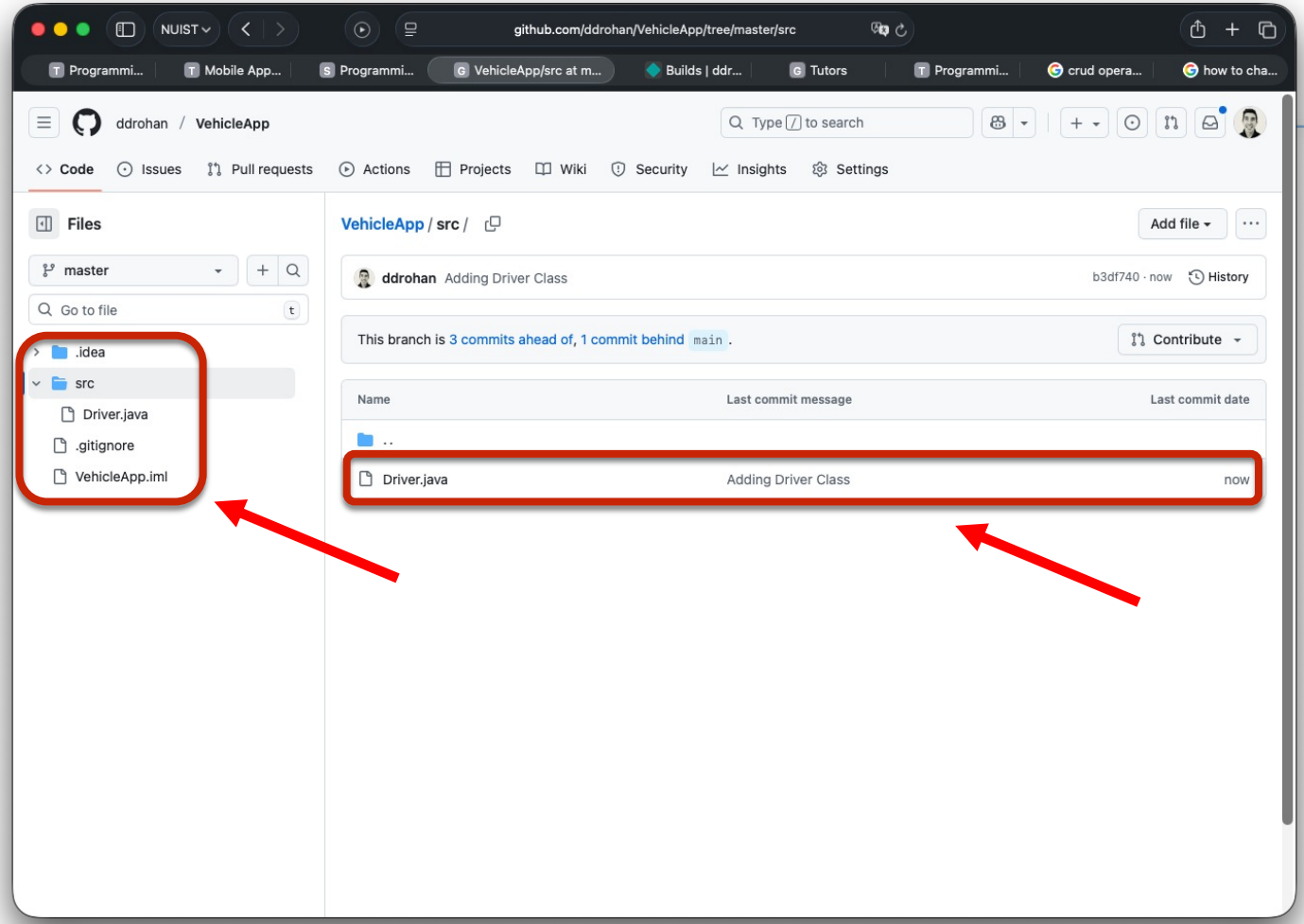
Select ‘PUSH’



Confirmation message of the commit and push



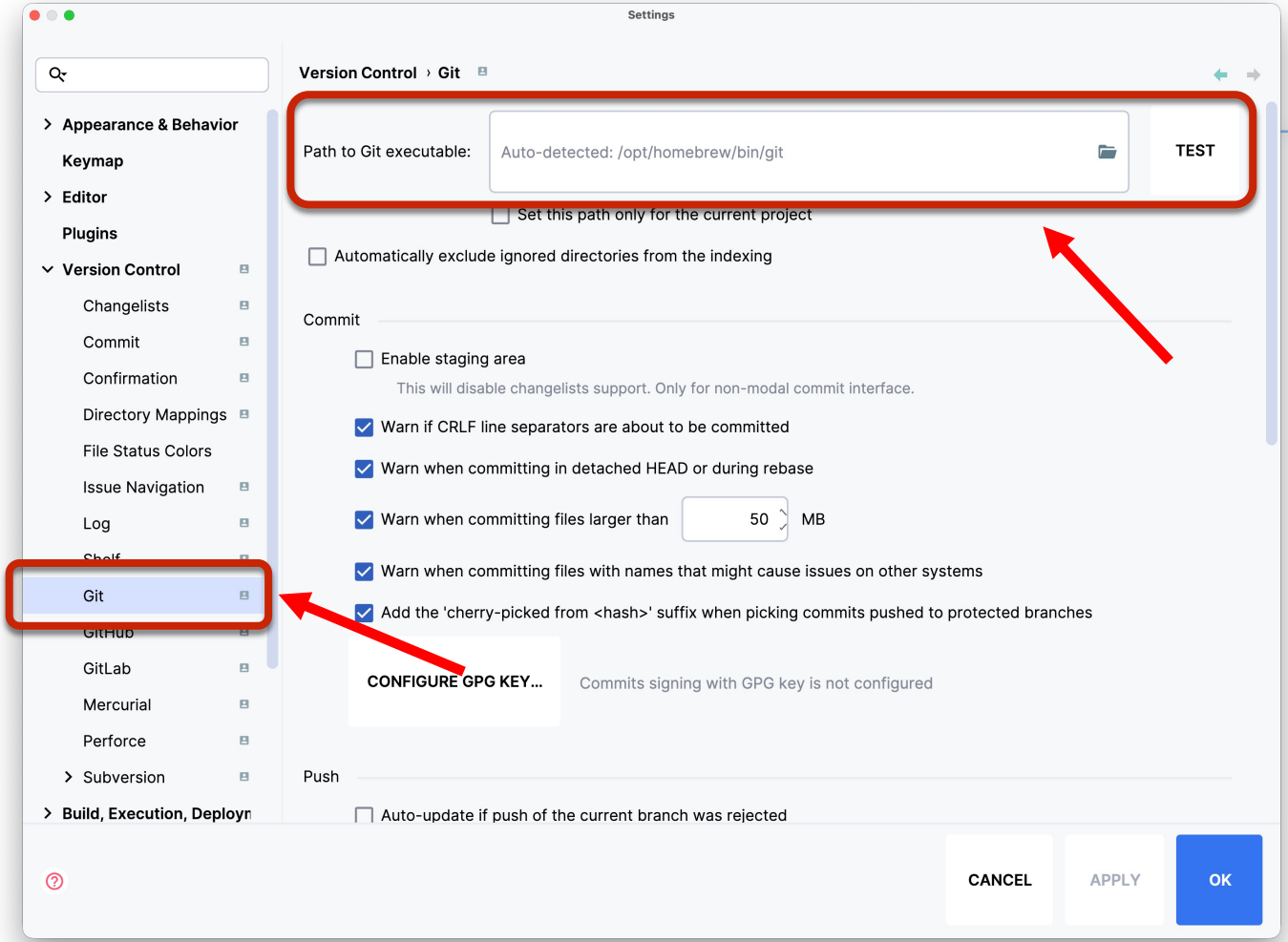
Confirm updates to your repository



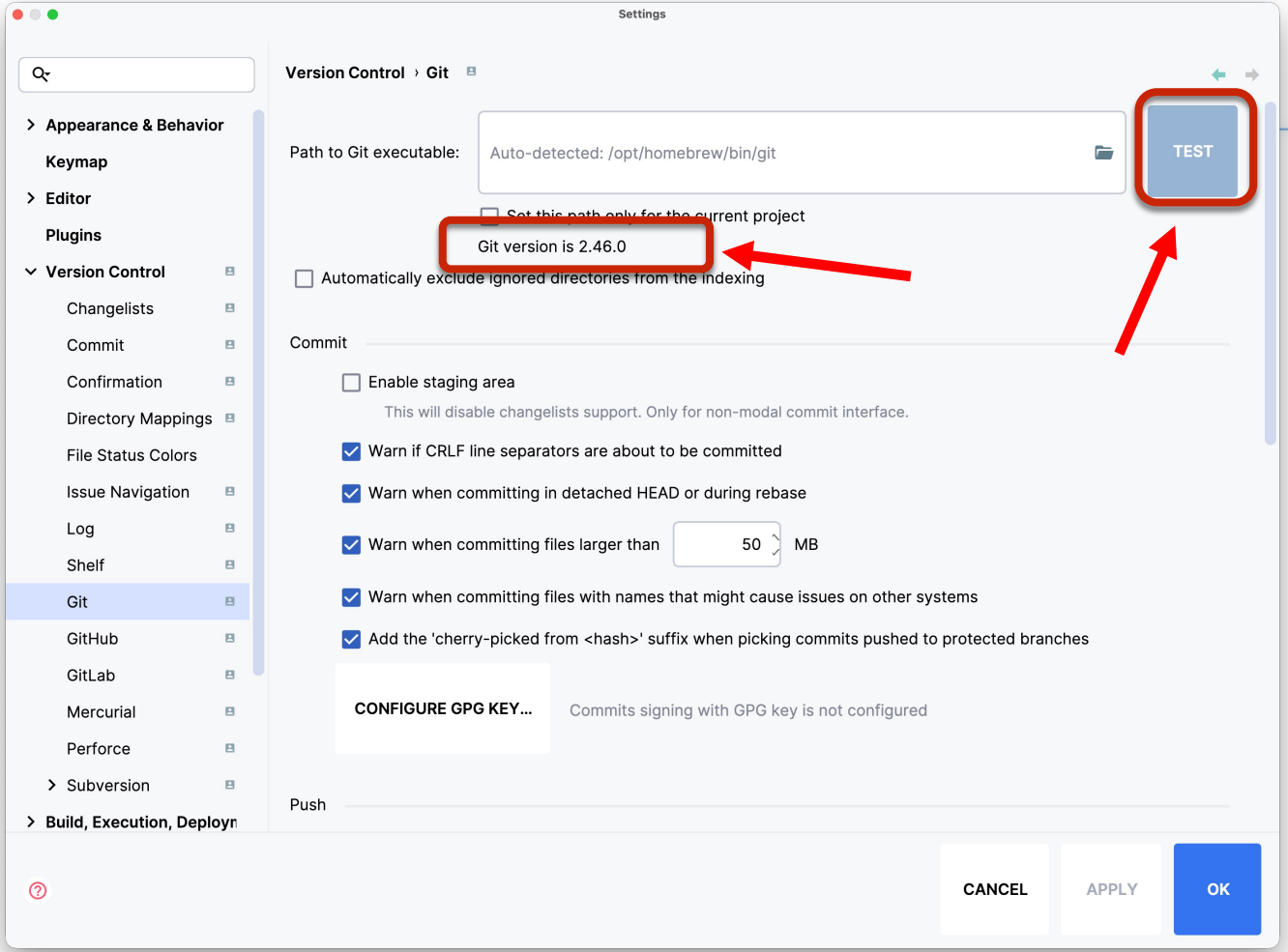
Confirm updates to your repository

Follow **Step 5** only for every additional change to your Project

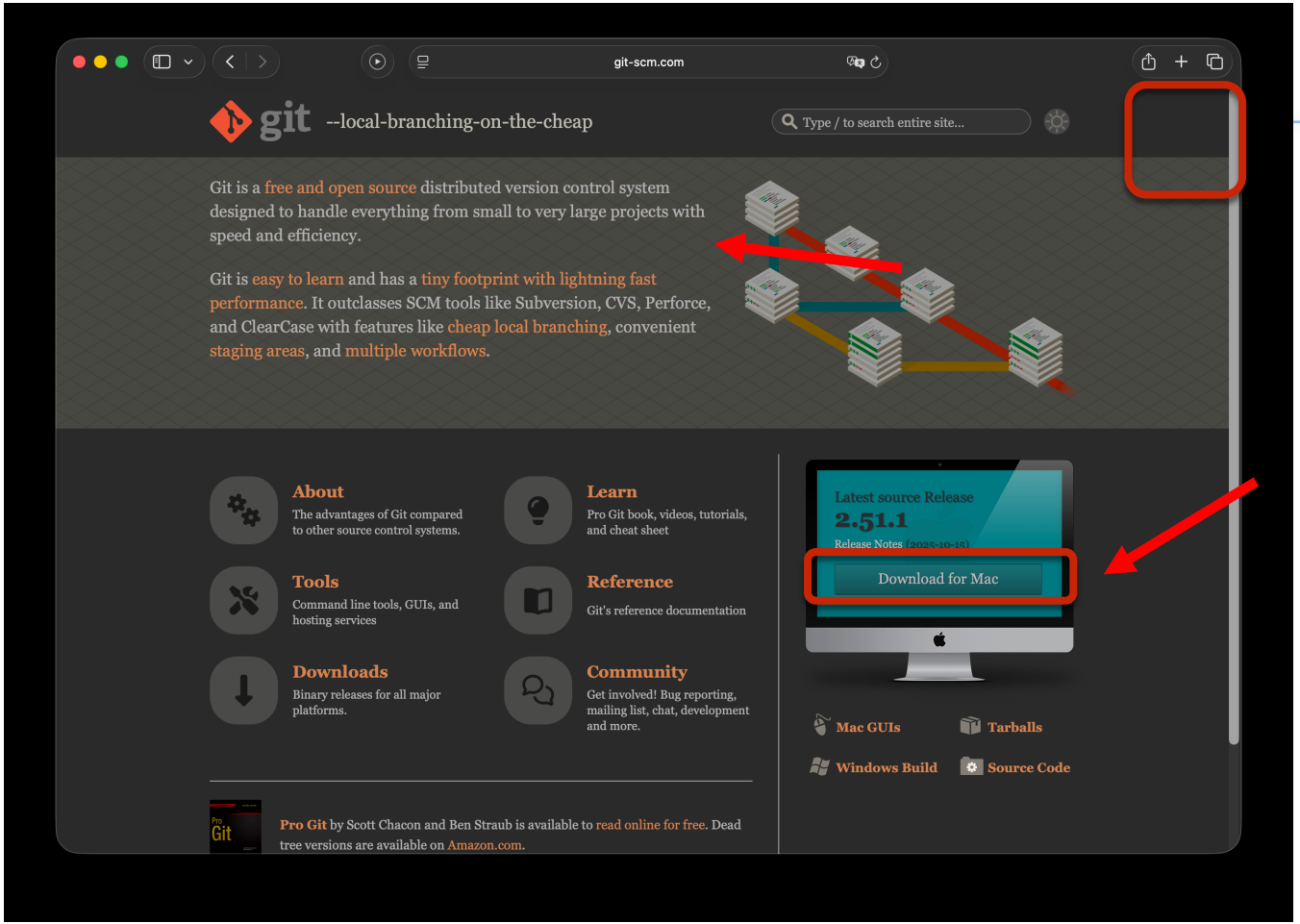
TROUBLE SHOOTING



Confirm your git is configured correctly



Confirm your git is configured correctly

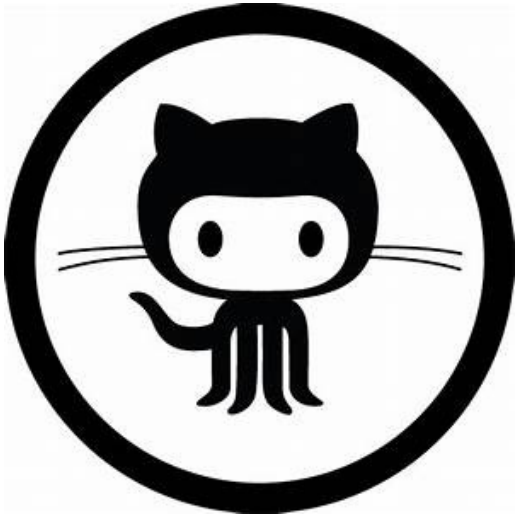


If not, install Git from

https://git-scm.com

and make sure path in Intellij points to this installation

Demonstration…



Questions?


Thanks.