# DCML-CPS - Lab

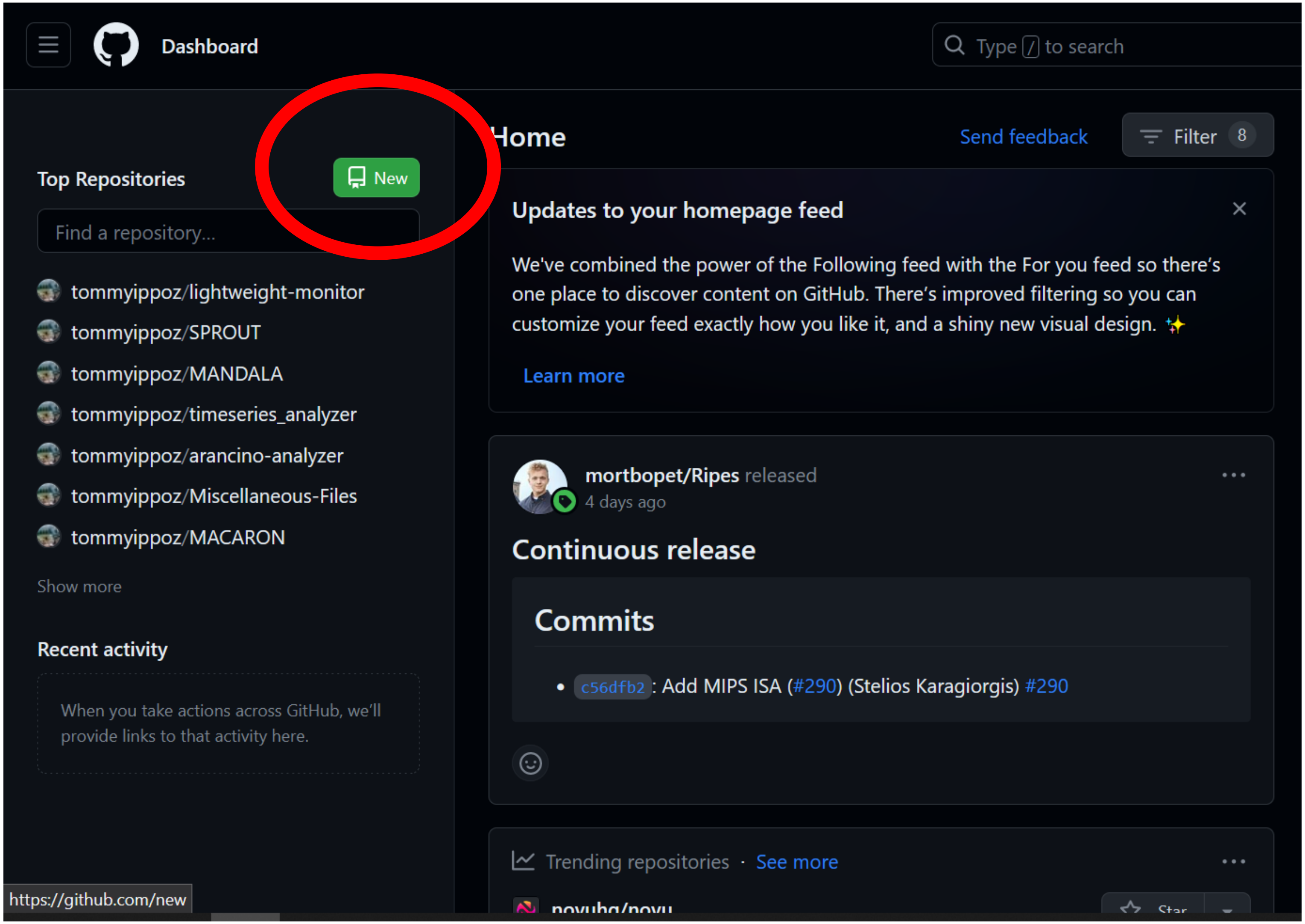
# Setting up a Git-Python Env

Tommaso Zoppi
University of Trento - Povo (IT)
tommaso.zoppi@unifi.it
tommaso.zoppi@unitn.it

# Lab Prep

- Prerequisites
- Having a GitHub Account
- Downloading a Python interpreter
- Then, either of the two
  - Downloading an IDE and a Git plugin
  - Download a Git client

# Lab Prep



Top Repositories

Find a repository...

- tommyippoz/lightweight-monitor
- tommyippoz/SPROUT
- tommyippoz/MANDALA
- tommyippoz/timeseries_analyzer
- tommyippoz/arancino-analyzer
- tommyippoz/Miscellaneous-Files
- tommyippoz/MACARON

Show more

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Home

Updates to your homepage feed

We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨

Learn more

mortbopet/Ripes released
4 days ago

Continuous release

Commits

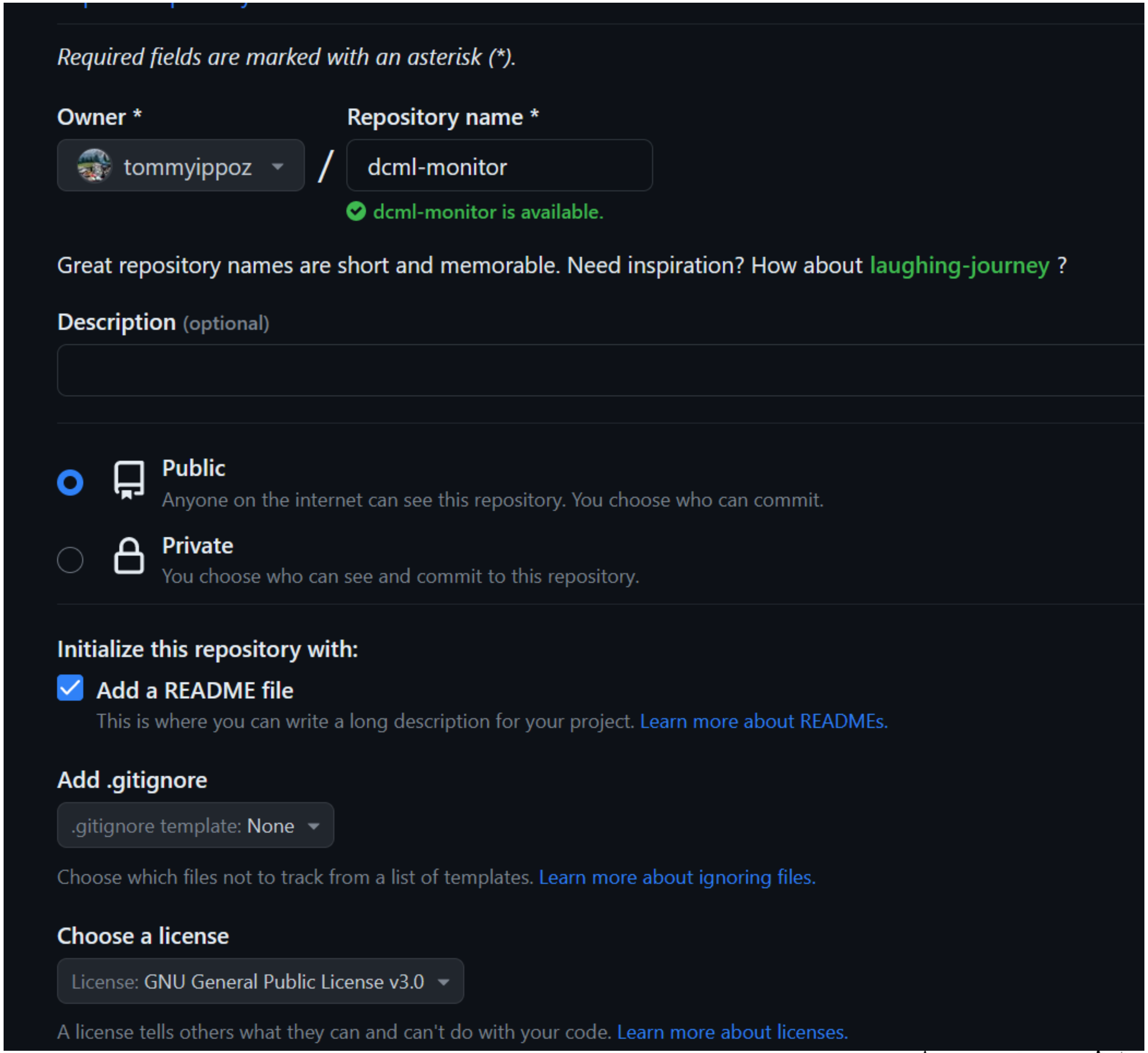
- c56dfb2: Add MIPS ISA (#290) (Stelios Karagiorgis) #290

Trending repositories · See more

novuhq/novu

https://github.com/new

# Lab Prep



*Required fields are marked with an asterisk (*).*

Owner * tommyippoz / Repository name * dcml-monitor

✓ dcml-monitor is available.

Great repository names are short and memorable. Need inspiration? How about laughing-journey ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

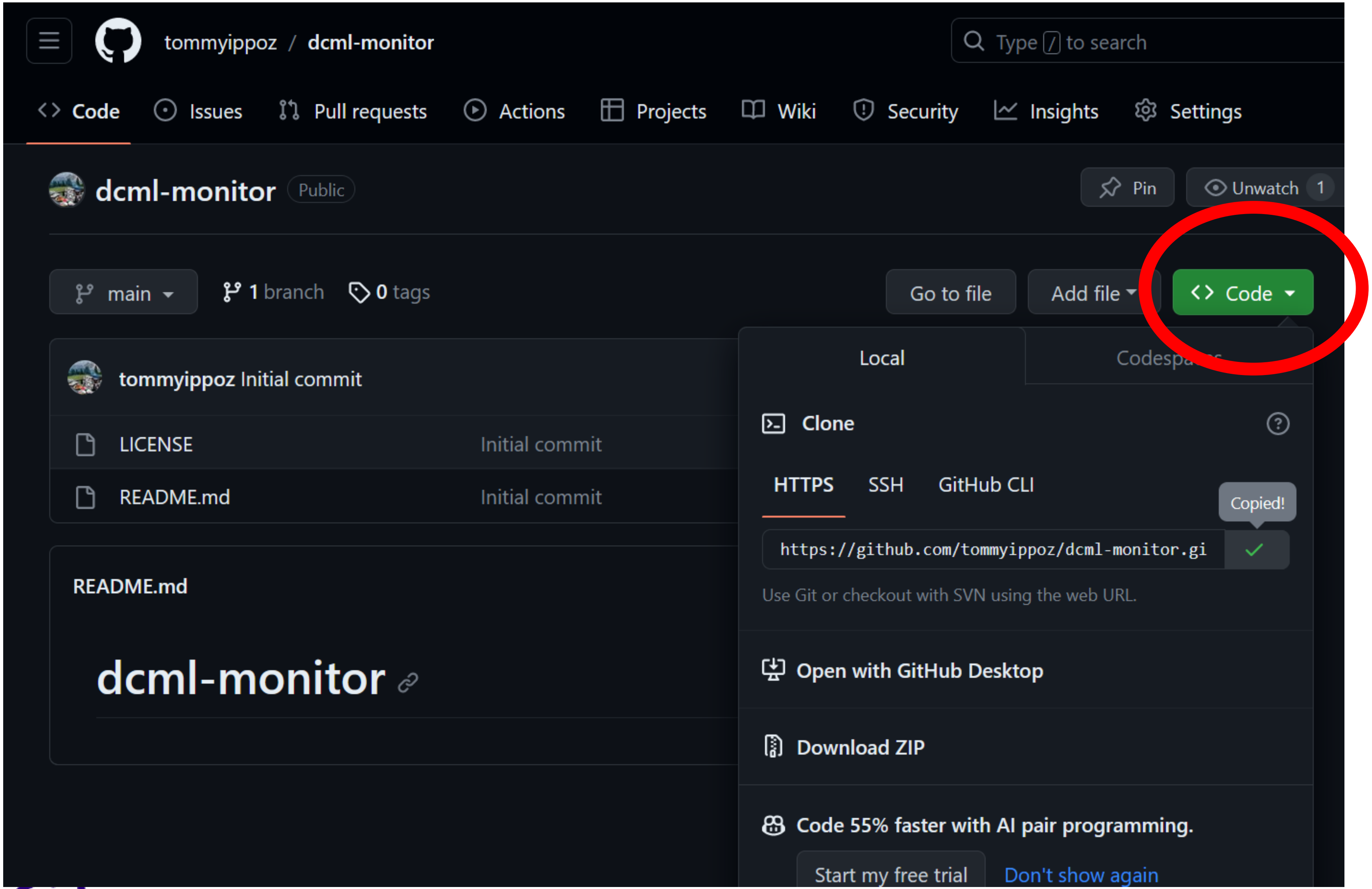
Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: GNU General Public License v3.0

A license tells others what they can and can't do with your code. Learn more about licenses.

# Lab Prep



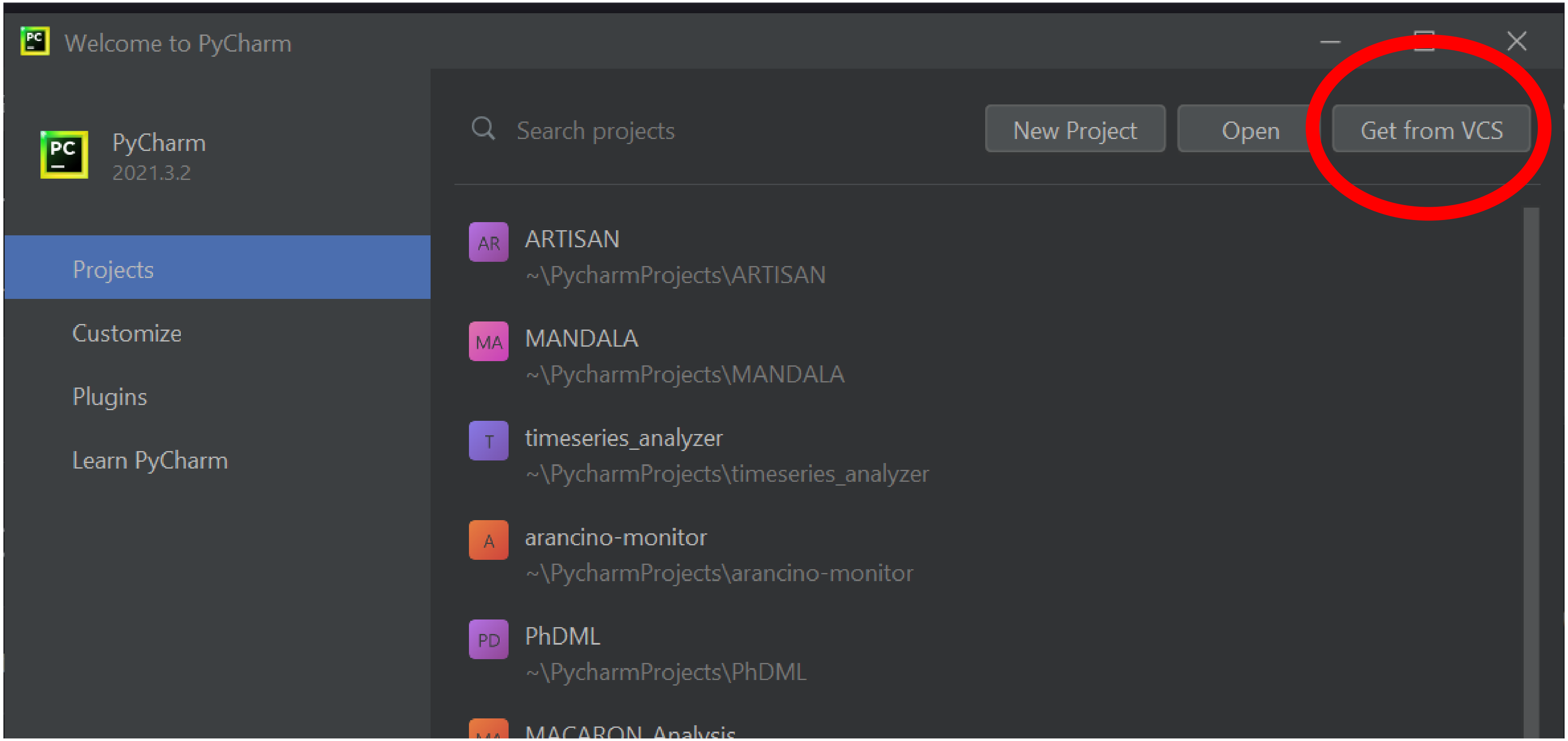
```
https://github.com/tommyippoz/dcml-monitor.gi
```

Use Git or checkout with SVN using the web URL.

Code 55% faster with AI pair programming.

# Lab Prep

- Now we move to pycharm



Welcome to PyCharm

PyCharm
2021.3.2

AR ARTISAN
~\PycharmProjects\ARTISAN

MA MANDALA
~\PycharmProjects\MANDALA

T timeseries_analyzer
~\PycharmProjects\timeseries_analyzer

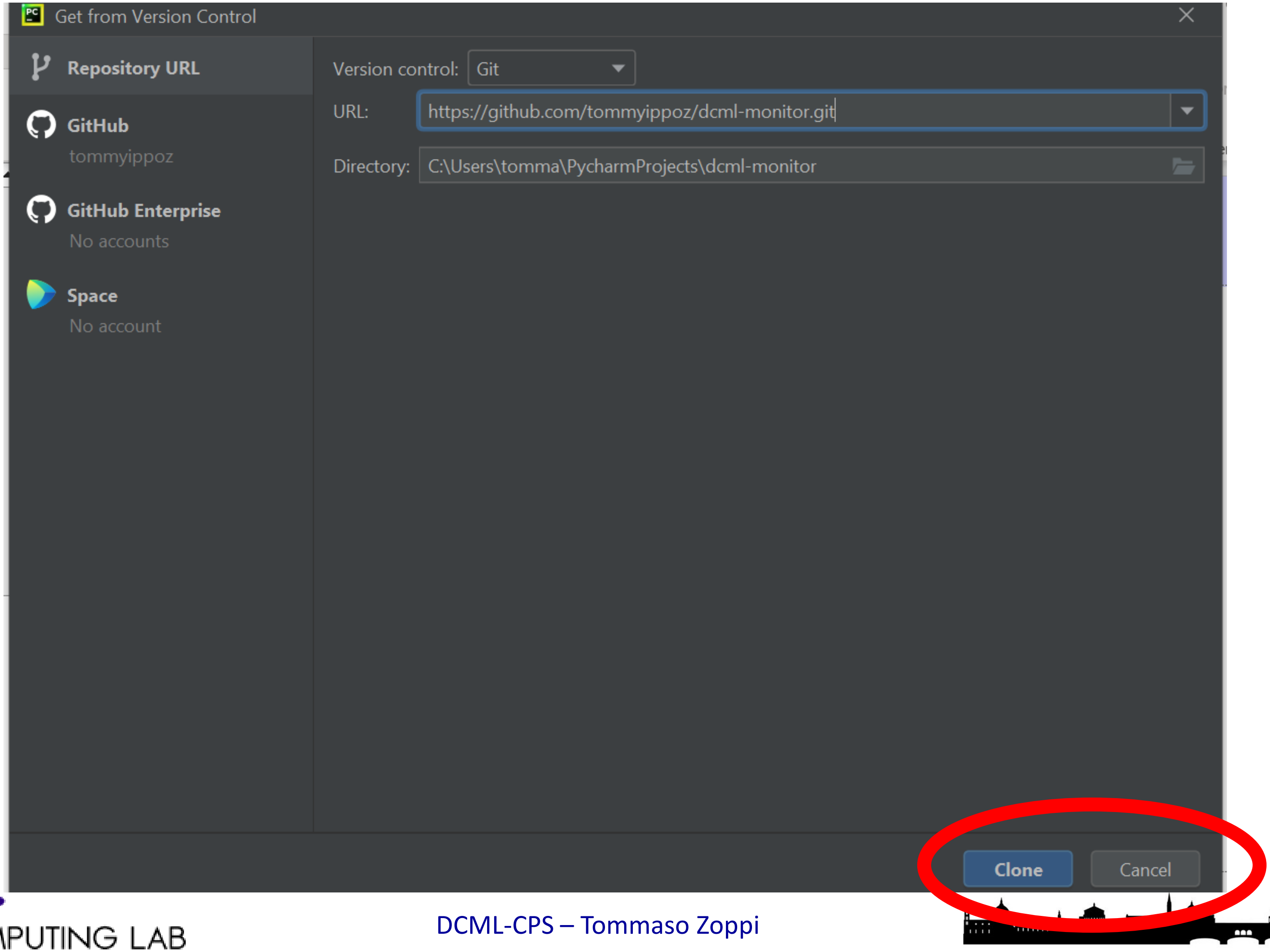
A arancino-monitor
~\PycharmProjects\arancino-monitor

PD PhDML
~\PycharmProjects\PhDML

MA MACARON_Analysis

# Lab Prep

- We paste what we copied before and clone



Get from Version Control

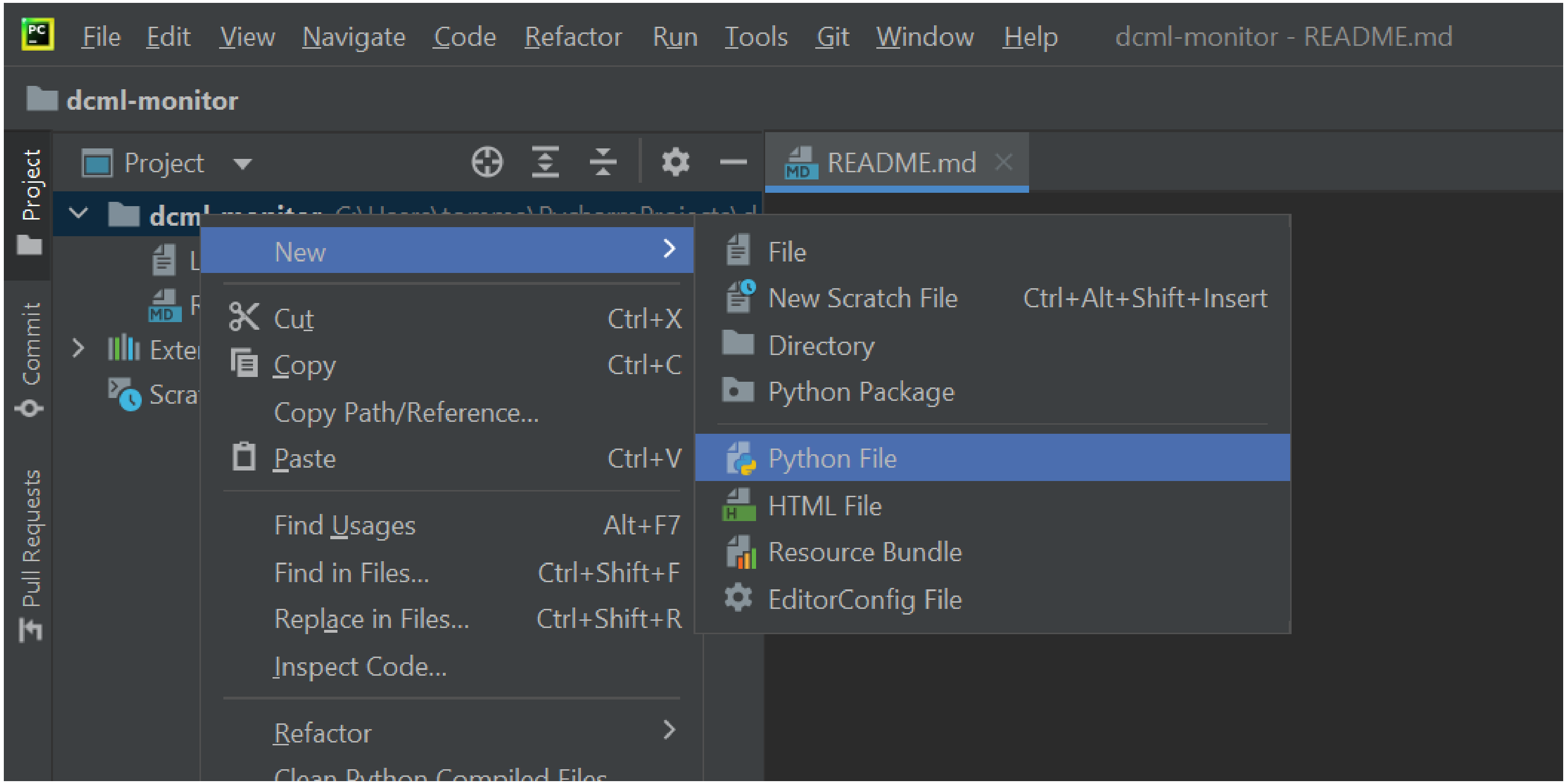
Version control: Git

URL: https://github.com/tommyippoz/dcml-monitor.git

Directory: C:\Users\tomma\PycharmProjects\dcml-monitor

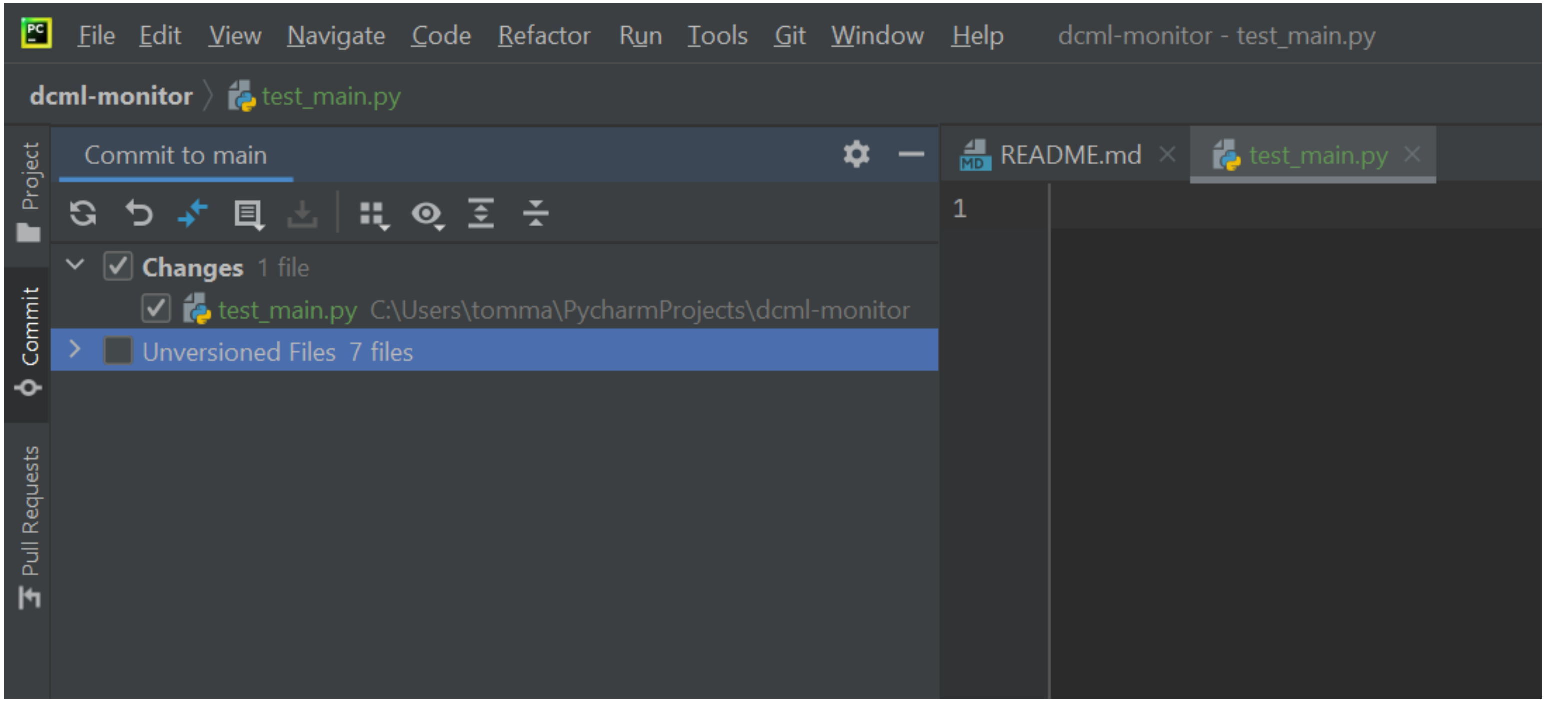
# Lab Prep

- Working env is ready, now lets add a new py file and then try to commit



# Lab Prep

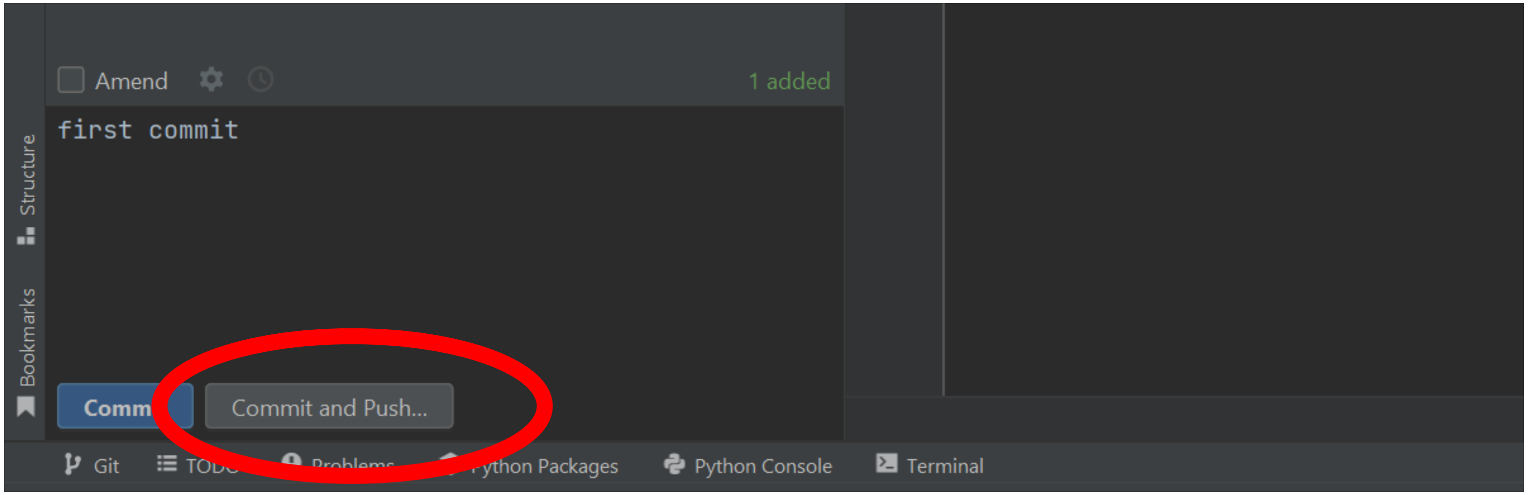
- Committing changes means finalizing them.
  - We commit the addition of the file


File Edit View Navigate Code Refactor Run Tools Git Window Help
dcml-monitor - test_main.py
dcml-monitor 〉 test_main.py
Project
Commit
Pull Requests
Commit to main
README.md ×
test_main.py ×
1

☑ Changes 1 file
☑ test_main.py C:\Users\tomma\PycharmProjects\dcml-monitor
□ Unversioned Files 7 files

# Lab Prep

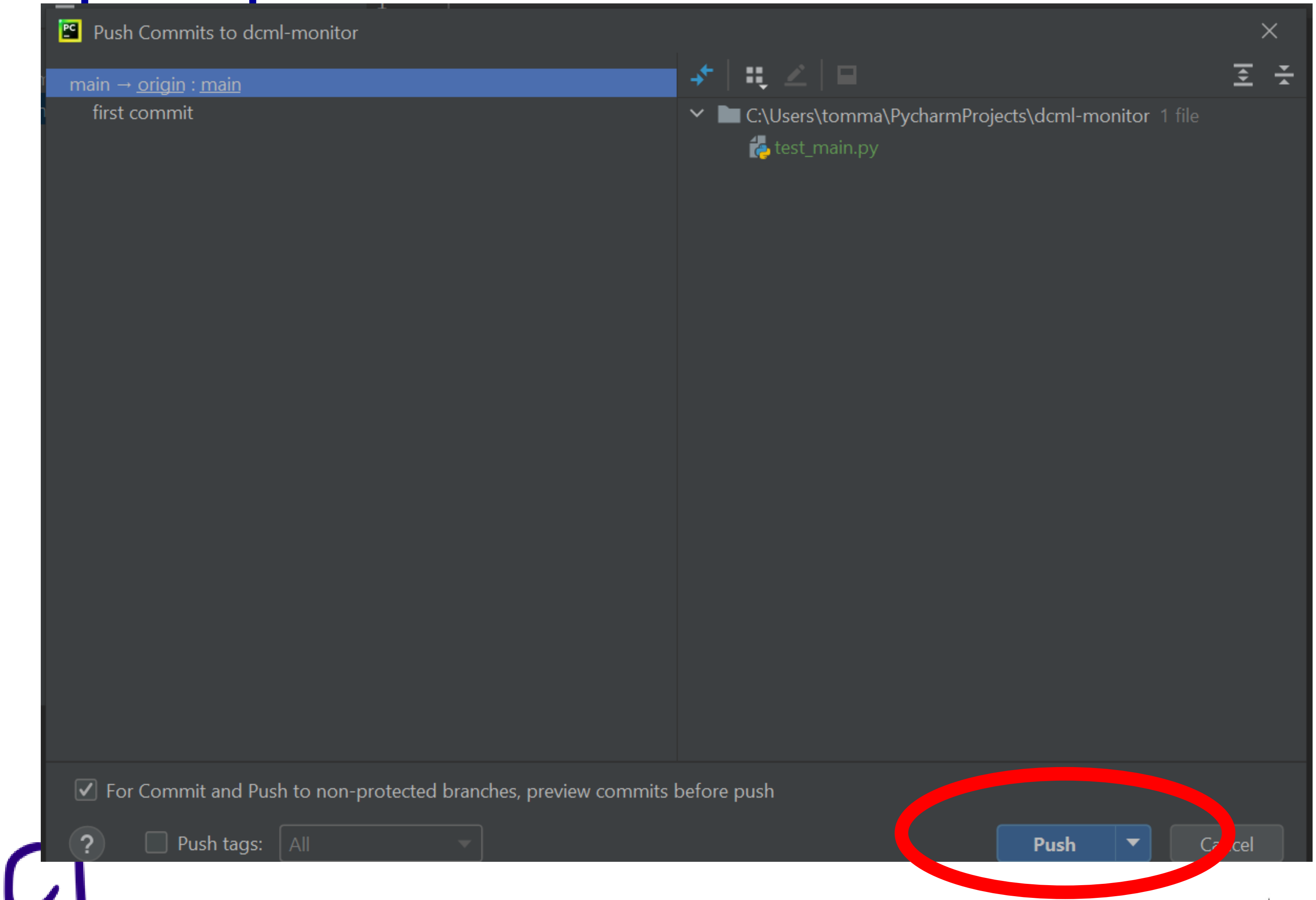
- Committing changes means finalizing them.
  - We commit the addition of the file
  - Write a textual tag for it



```
first commit
```

# Remote Push

- It prompts a confirmation



Push Commits to dcml-monitor

main → origin : main

first commit

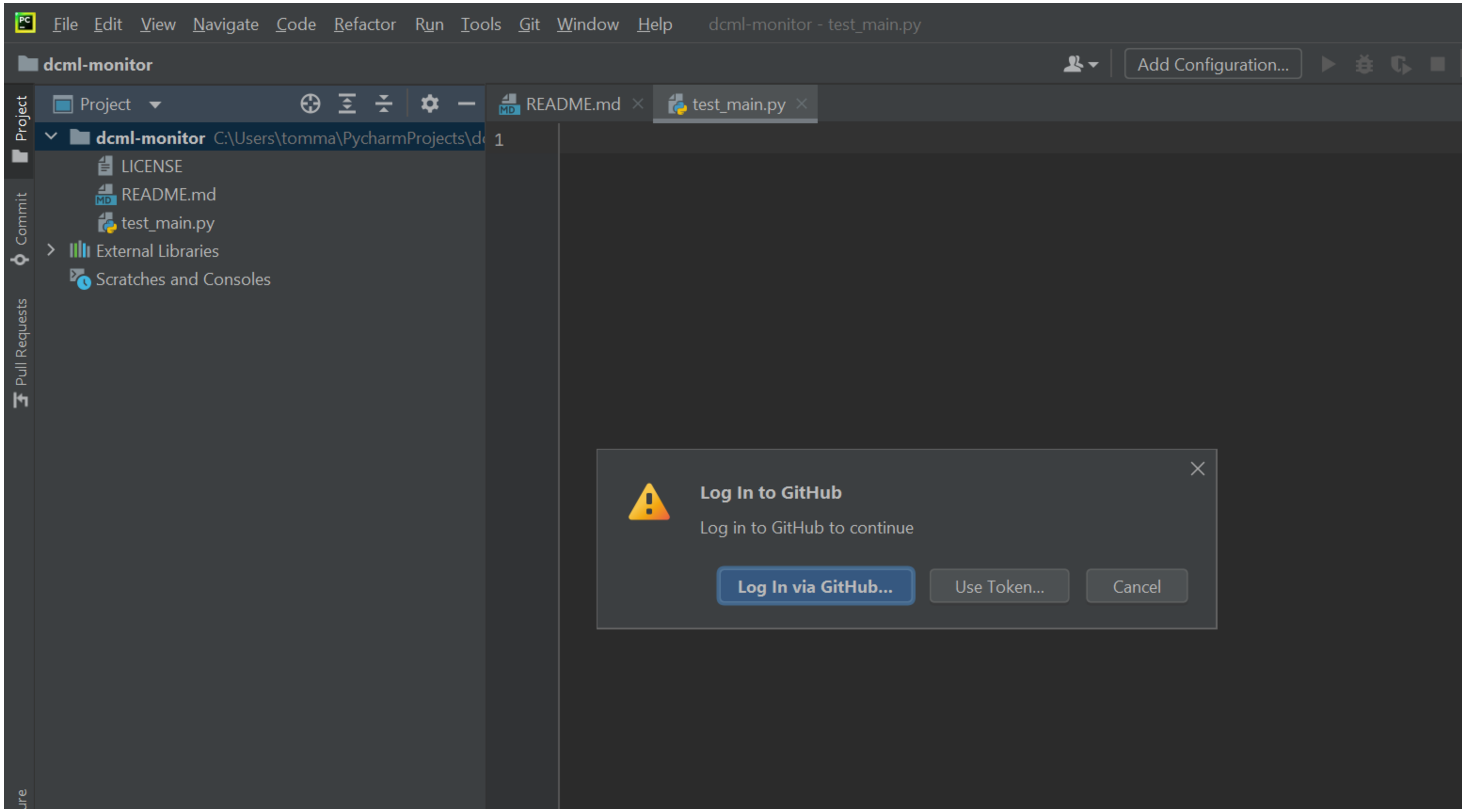
C:\Users\tomma\PycharmProjects\dcml-monitor 1 file

test_main.py

☑ For Commit and Push to non-protected branches, preview commits before push

# Remote Push

- Pushing to remote requires a token or authentication

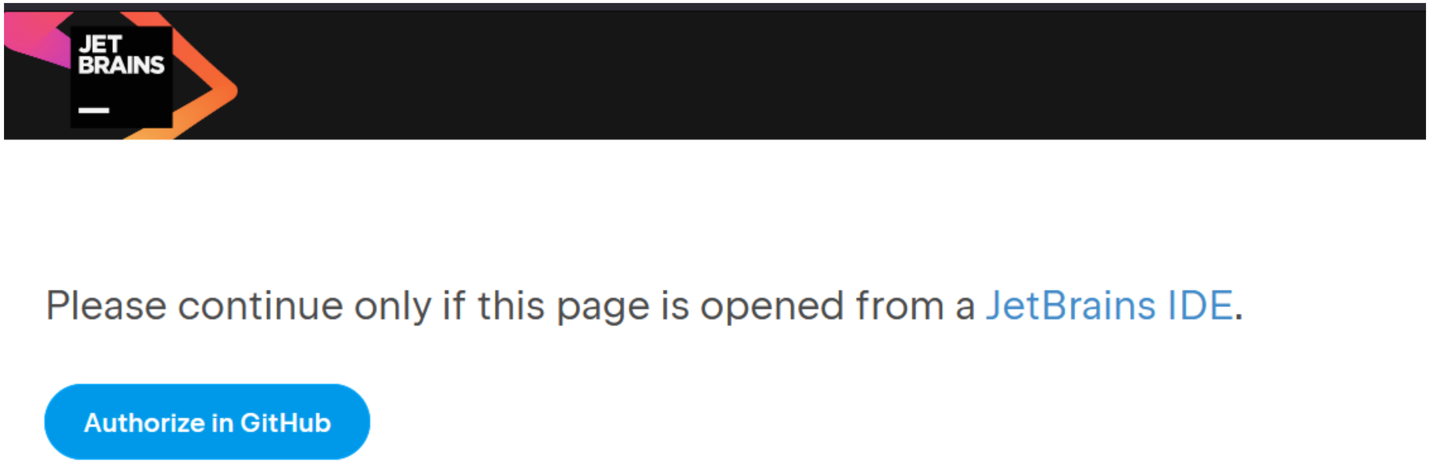


Log In to GitHub

Log in to GitHub to continue

# Remote Push

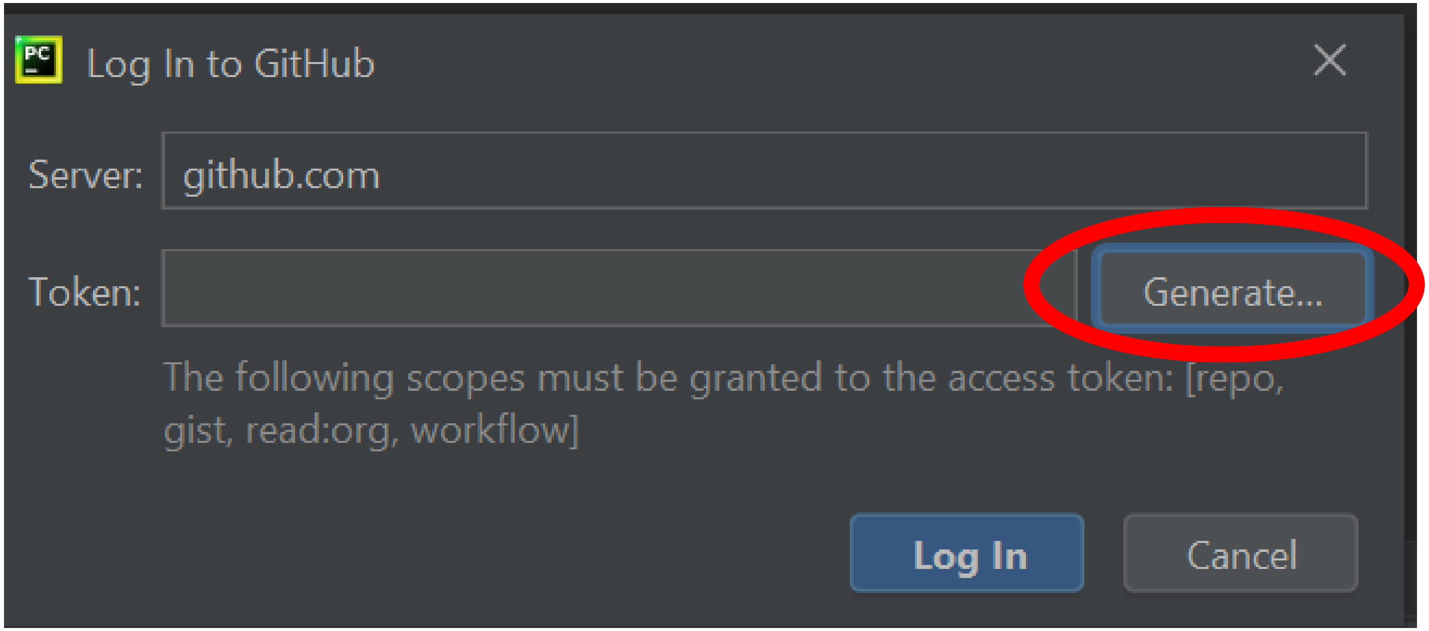
- Pushing to remote requires a token or **authentication**



Please continue only if this page is opened from a JetBrains IDE.

# Creating GitHub Tokens

- Pushing to remote requires a **token** or authentication



Log In to GitHub

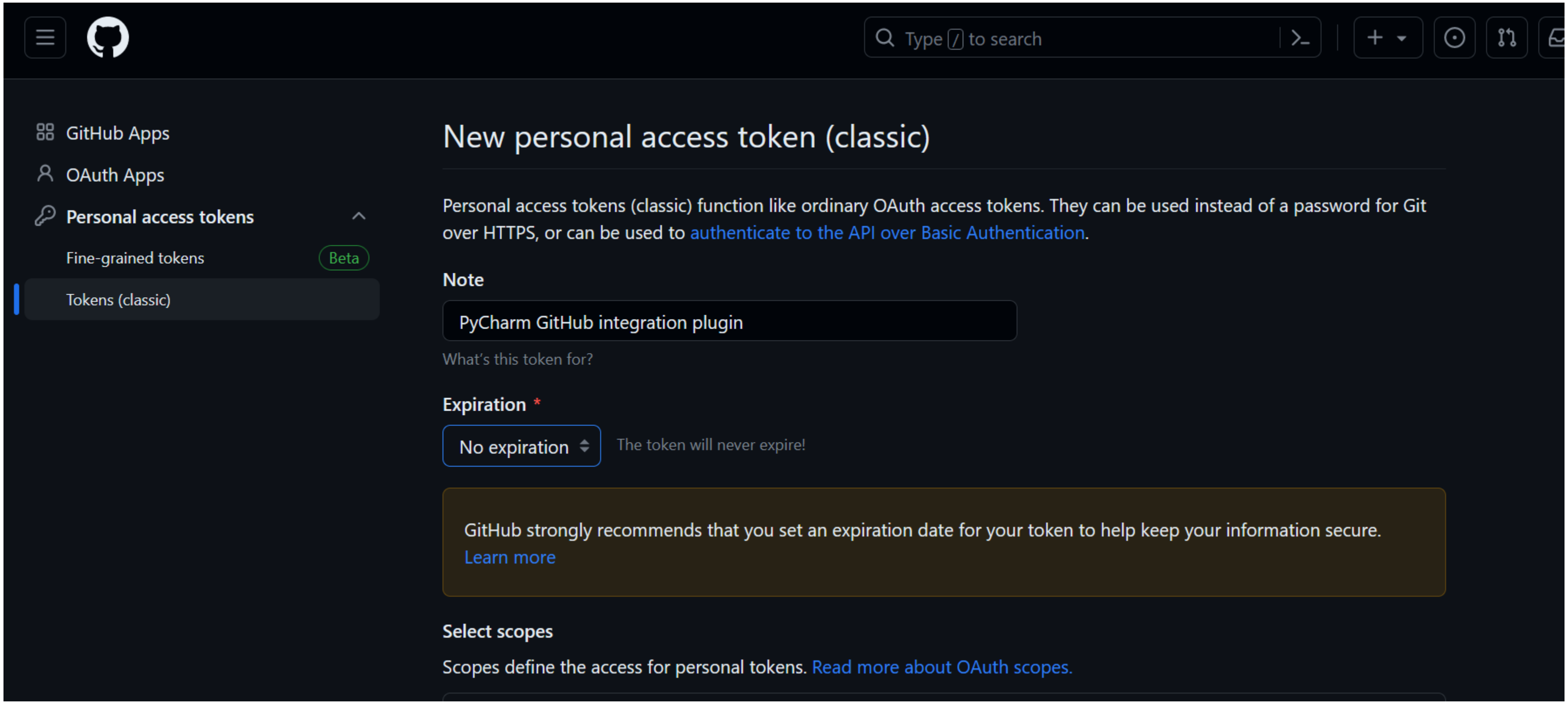
Server: github.com

Token:

The following scopes must be granted to the access token: [repo, gist, read:org, workflow]

# Creating GitHub Tokens

- Pushing to remote requires a **token** or authentication



## New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

**Note**

PyCharm GitHub integration plugin

What's this token for?

**Expiration** *

No expiration — The token will never expire!

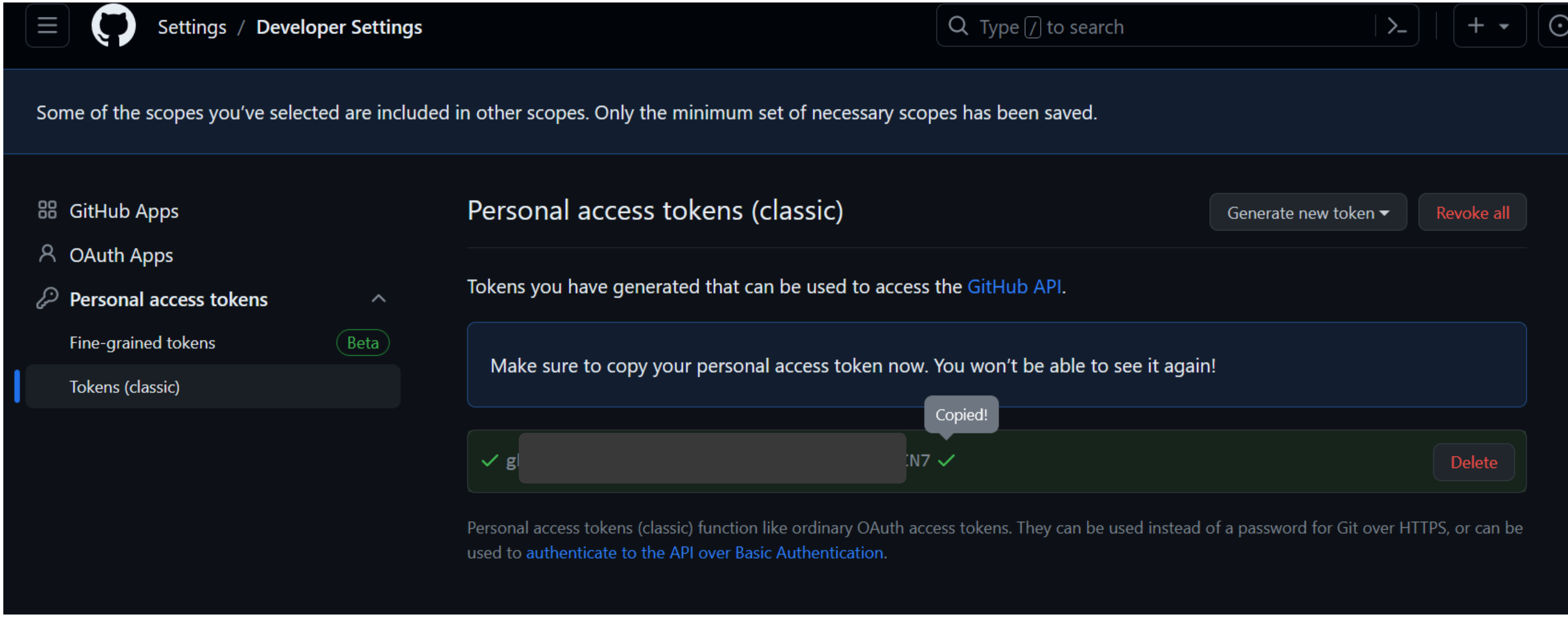
GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. Learn more

**Select scopes**

Scopes define the access for personal tokens. Read more about OAuth scopes.

# Creating GitHub Tokens

- Pushing to remote requires a **token** or authentication



Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic)

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ [redacted] ✓

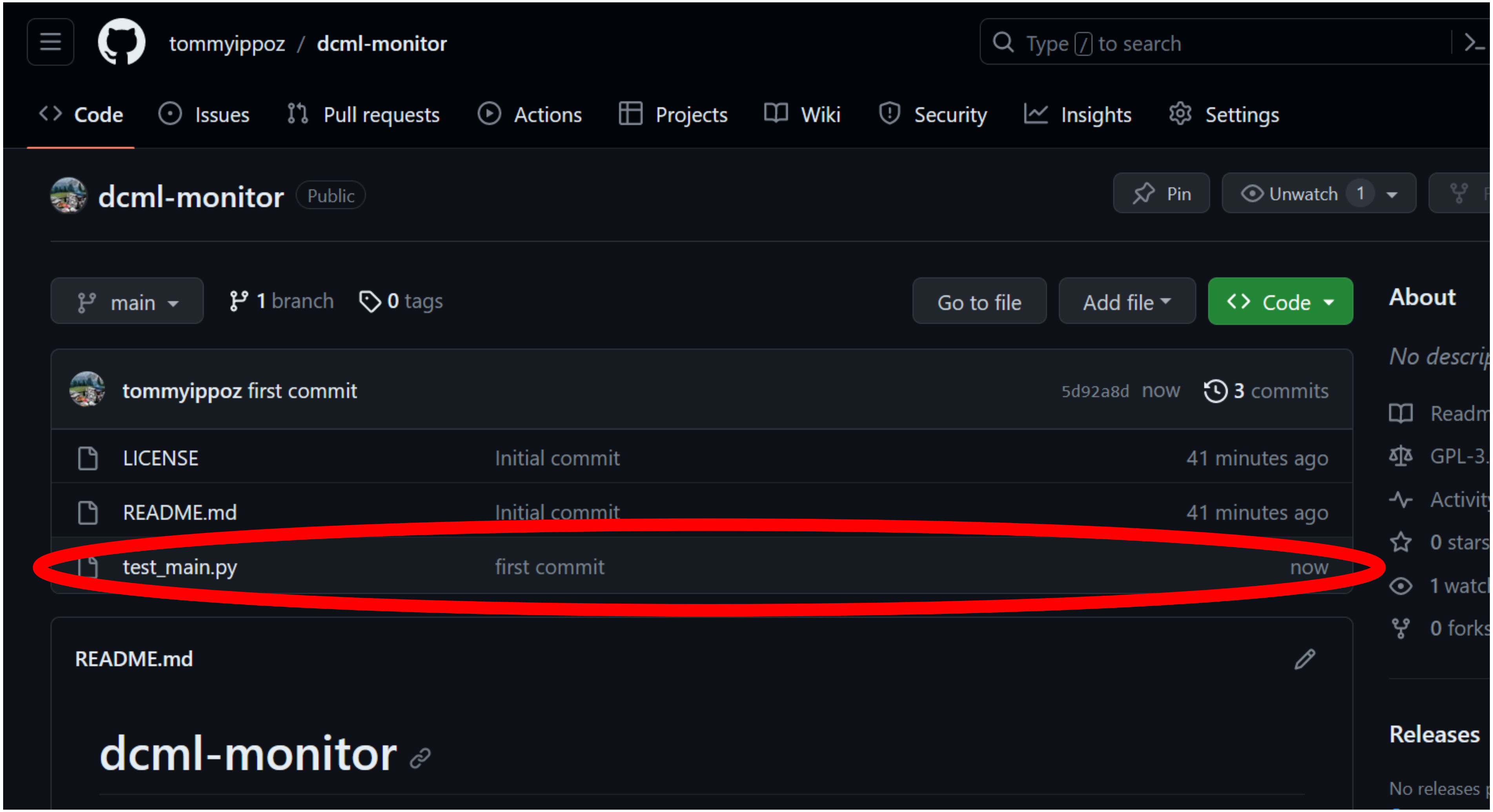
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

# Creating GitHub Tokens

- Now you are good to go with pushes and commits
  - Tokens are thought for a single-application, single-repo use, you should organize them accordingly

- Now you can copy-paste the token in PyCharm
  - The push will finalize

# Repo Update



tommyippoz first commit 5d92a8d now 3 commits

| | | |
|---|---|---|
| LICENSE | Initial commit | 41 minutes ago |
| README.md | Initial commit | 41 minutes ago |
| test_main.py | first commit | now |

README.md

dcml-monitor

# Lets Try it Together

- Hands-On Experimentation!