

GitHub Guide

Using Git in your Dev Environment for the First Time!

Wow! Good job setting up your dev-environment for the assignments – now it's time to set up Git. If you have not set up the dev-environment yet, go to: [Setup Guide](#) before returning here.

Steps to set up Git for your docker environment:

(Note: If you already have Git installed locally, you can keep using your local version of Git and the changes will be propagated into your dev-environment as well! If you choose to use your local Git, then you can skip this initial GitHub setup.)

1. Set up your git email by running the command:

```
git config --global user.email "you@example.com"
```

(Note: replace you@example.com with your Github email address)

(Note 2: Make sure you keep the quotation marks around the email)

2. Set up your git name by running the command:

```
git config --global user.name "Your Name"
```

(Note: replace Your Name with... you guessed it: your name)

(Note 2: Make sure you keep the quotation marks around your name)

3. Now when you try to make a push to a GitHub repository, you will encounter this response:

```
Username for 'https://github.com':
```

Type in your GitHub username, and then press enter.

4. Then, you will see this response:

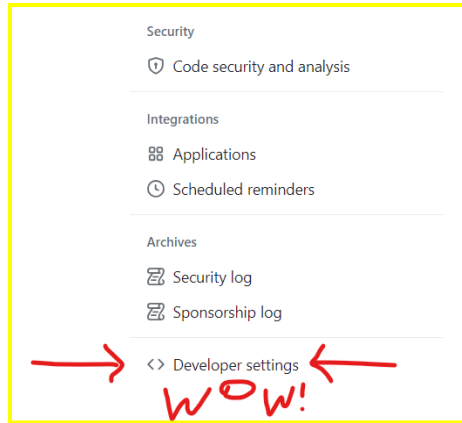
```
Password for 'https://<YOUR_USERNAME>@github.com':
```

Copy and paste over your personal access token into password, and then press enter. If you don't have a personal access token set up, follow the guide below!

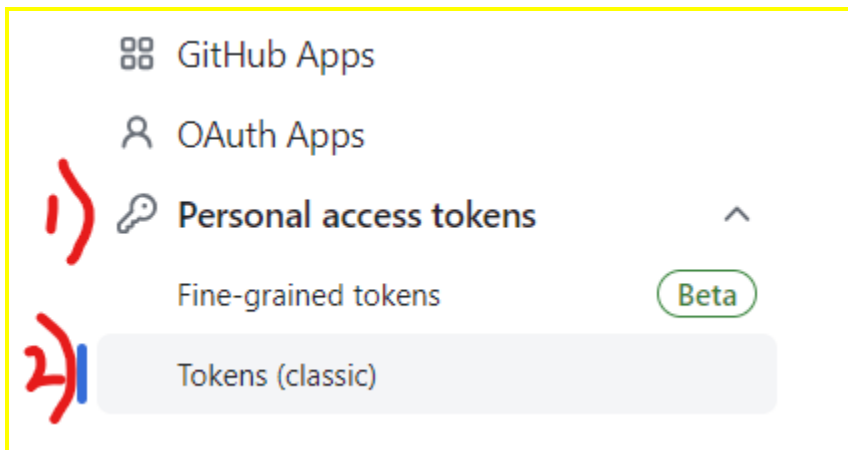
Guide to Setting up Your GitHub Personal Access Token!

Note: If you have not set up a personal access token, follow these steps:

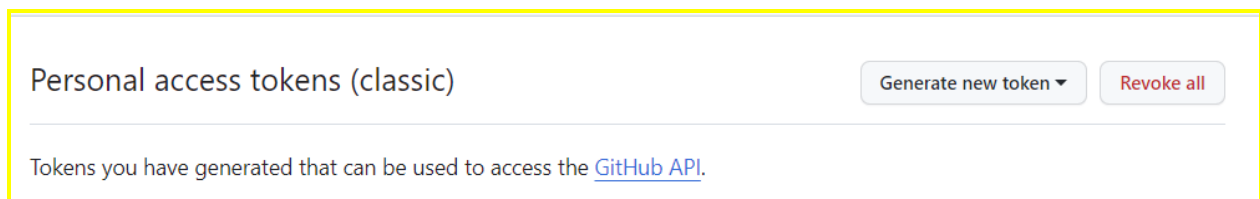
1. Go to your GitHub profile settings
2. Select the **Developer Settings** tab on the bottom left



3. Click **Personal access tokens**, and then **Tokens (classic)**



4. Click **Generate new token** and then **Generate new token (classic)**



5. Type in a note as a description of the token, and select the **repo** box

GitHub Apps

OAuth Apps

Personal access tokens Beta

Fine-grained tokens

Tokens (classic)

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

ENTER WHATEVER YOU WANT HERE LOL

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.](#)

☒ repo	Full control of private repositories
☒ repo:status	Access commit status
☒ repo_deployment	Access deployment status
☒ public_repo	Access public repositories
☒ repo:invite	Access repository invitations
☒ security_events	Read and write security events
☐ workflow	Update GitHub Action workflows
☐ write:packages	Upload packages to GitHub Package Registry

OPTIONAL!!

2

1

6. Click **generate token** at the bottom of the page, and now **WRITE DOWN/COPY DOWN YOUR TOKEN!!!!** You will never be able to see it again after this 🤪🤪

Congratulations, you have successfully created your own personal access token to use as a password for git!

Project Setup

We use GitHub invites to distribute the assignments throughout the semester. You will be working with just one repository for most projects, which will contain a directory for each project. Let's walk through how to set up your assignments. We will use the first project, Go, as an example.

To begin, accept the [GitHub invites assignment for Assignment 1: Go](#), and then clone the repo into the **home** directory of your dev-environment:

1. First accept the google classroom assignment.
2. Clone the repo. Do this by going to the GitHub Repo link created by classrooms, and clicking clone. Copy the url in the box, navigate into the home directory of your dev-environment in a terminal, and then use the following command:

```
$ git clone <github invites stencil url>
```

Note: Make sure to use the correct url for ssh or https

Once you cloned the repository, if you open it you should see the directories `cmd/dinodb`, `pkg`, and `test`, as well as some other supporting files. The only directories you will have to touch in this course is `pkg` and `test`.

The implementation work you will do for this course will be in the `pkg` directory, and the `test` directory is where you will write any of your own `tests` for your projects. If you open `pkg` you will see the directories `list` and `rep1`. These contain the files you will need to edit to implement the first project, `Go`. Throughout the course, we will add more directories to `pkg` for future projects you will complete.

For each project, you will repeat the above steps with a fresh Github Classroom link.

Updates to Assignments

Note:

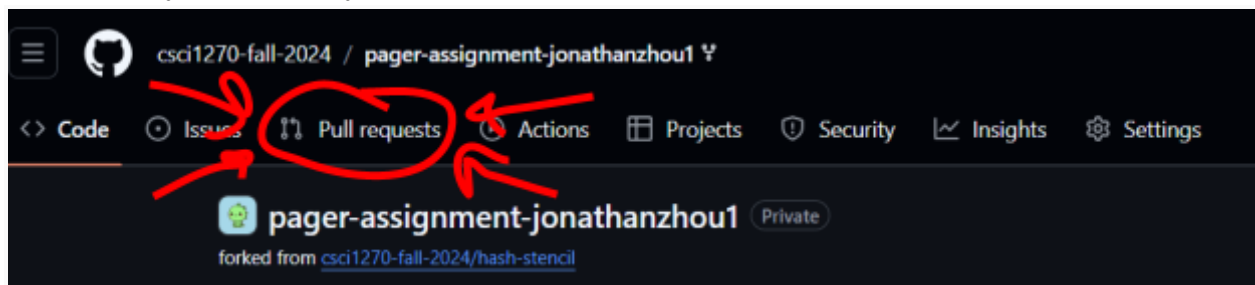
If GitHub Classroom does not automatically create a pull request in your stencil, please follow the instructions listed in the [Fallback Github Guide](#).

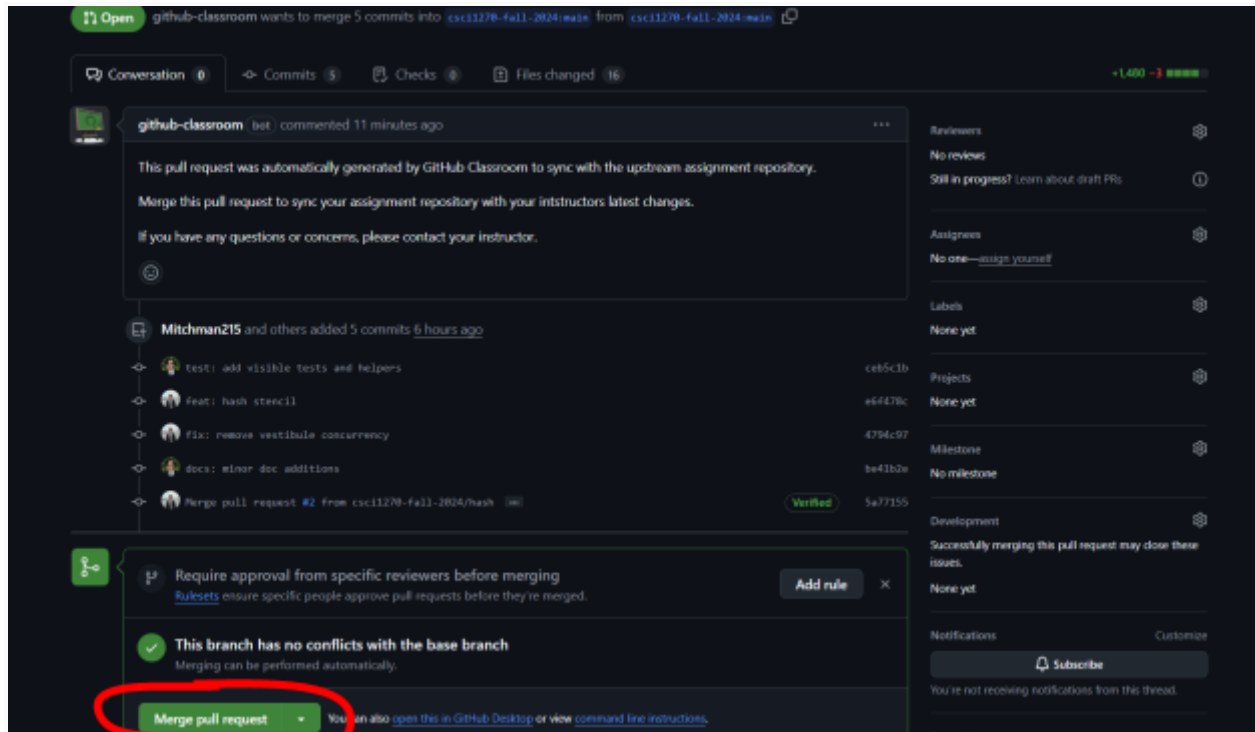
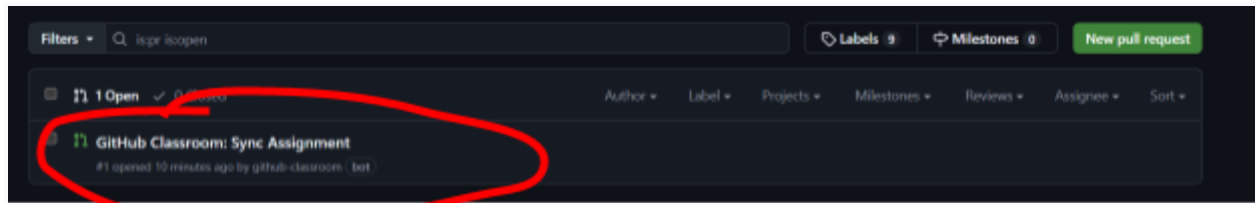
Pulling changes for projects

In the event that we update the stencil, we will use GitHub to create a pull request in your repository. After merging this pull request (and figuring out any merge conflicts as necessary), make sure to call 'git pull' on your local repository to catch up to the most recent version of the repository on GitHub, and then you are set to go!

Pull Requests

We expect you to be somewhat familiar with Git from the intro course and systems course you took as prerequisites for this course, but here is a quick picture guide about how to merge a pull request into your repository.





GitHub Classroom: Sync Assignment #1

Merged jonathanzhou1 merged 5 commits into csc11270-fall-2024:main from csc11270-fall-2024:main now

Conversation 0 Commits 5 Checks 0 Files changed 16 +1,480 -3

github-classroom bot commented 12 minutes ago

This pull request was automatically generated by GitHub Classroom to sync with the upstream assignment repository.

Merge this pull request to sync your assignment repository with your instructor's latest changes.

If you have any questions or concerns, please contact your instructor.

Mitchman215 and others added 5 commits 6 hours ago

- test: add visible tests and helpers
- feat: hash stencil
- fix: remove vestibule concurrency
- docs: minor doc additions
- Merge pull request #2 from csc11270-fall-2024/hash

jonathanzhou1 merged commit 23b966a into csc11270-fall-2024:main now

Reviewers: No reviews

Assignees: No one—assign yourself!

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues.

Notifications: Unsubscribe

Went to

And you are done merging!!