Computer Science

Environment Setup

Appalachian State University

Updated: 08/22/2024

Author: Professor Willow Sapphire

Updates: Prof. Danielle Lapensee-Rankine

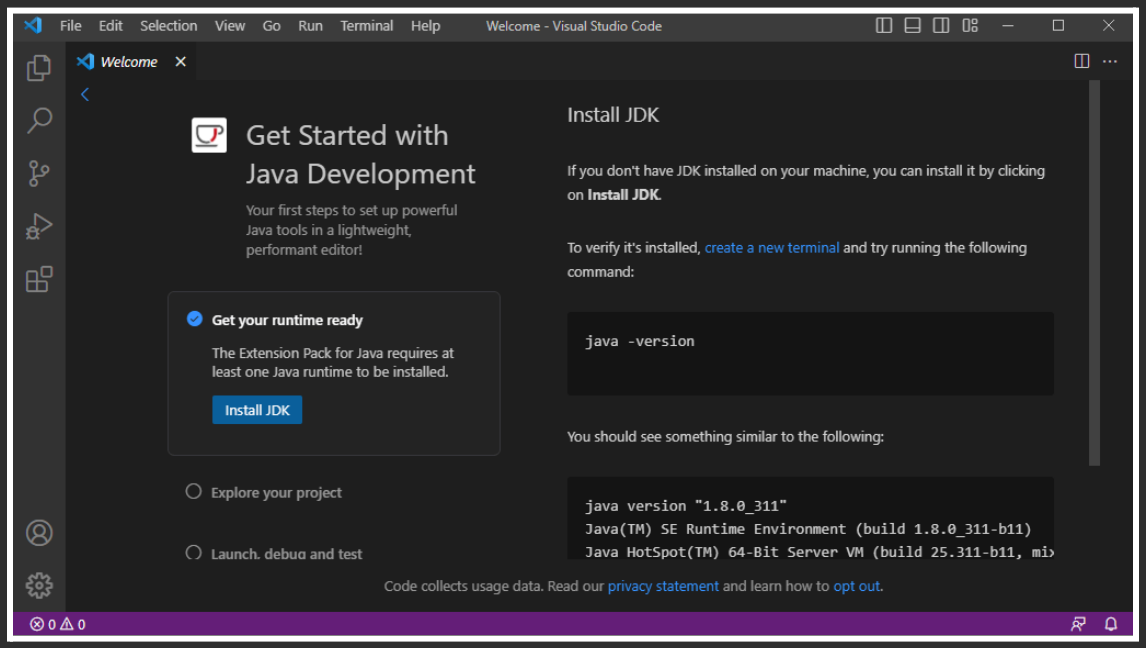
Table of Contents

Visual Studio Code

Throughout the labs in this course you will be using the Visual Studio Code (VSCode) text editor. This part will walk you through setting this up on your personal machine.

Installing VS Code

- Go to https://code.visualstudio.com and download the version for your operating system.
- Run the installation file that was downloaded and go through the installation instructions (default options are fine).
- Launch VSCode to ensure that it is working. You should see something like this. We will assume you already have Java installed. Search for it or ask for help if you do not.



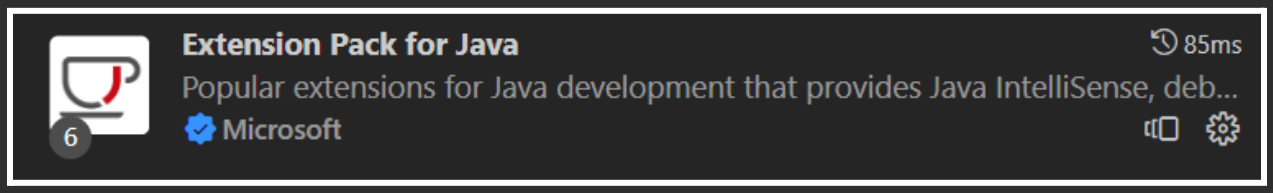
Step 2: Installing Extensions

One of the benefits of using VSCode is that it has many extensions that we can install to make our lives easier. This step will be split into two parts: required extensions and suggested extensions.

To install extensions, go to the panel on the left side that looks like the image to the left. Click on the icon that looks like 4 squares to open the extensions menu (it will also say extensions if you mouse over it). You can use the search bar to search for extensions and each extension will have an install button on it that you can use to install it.

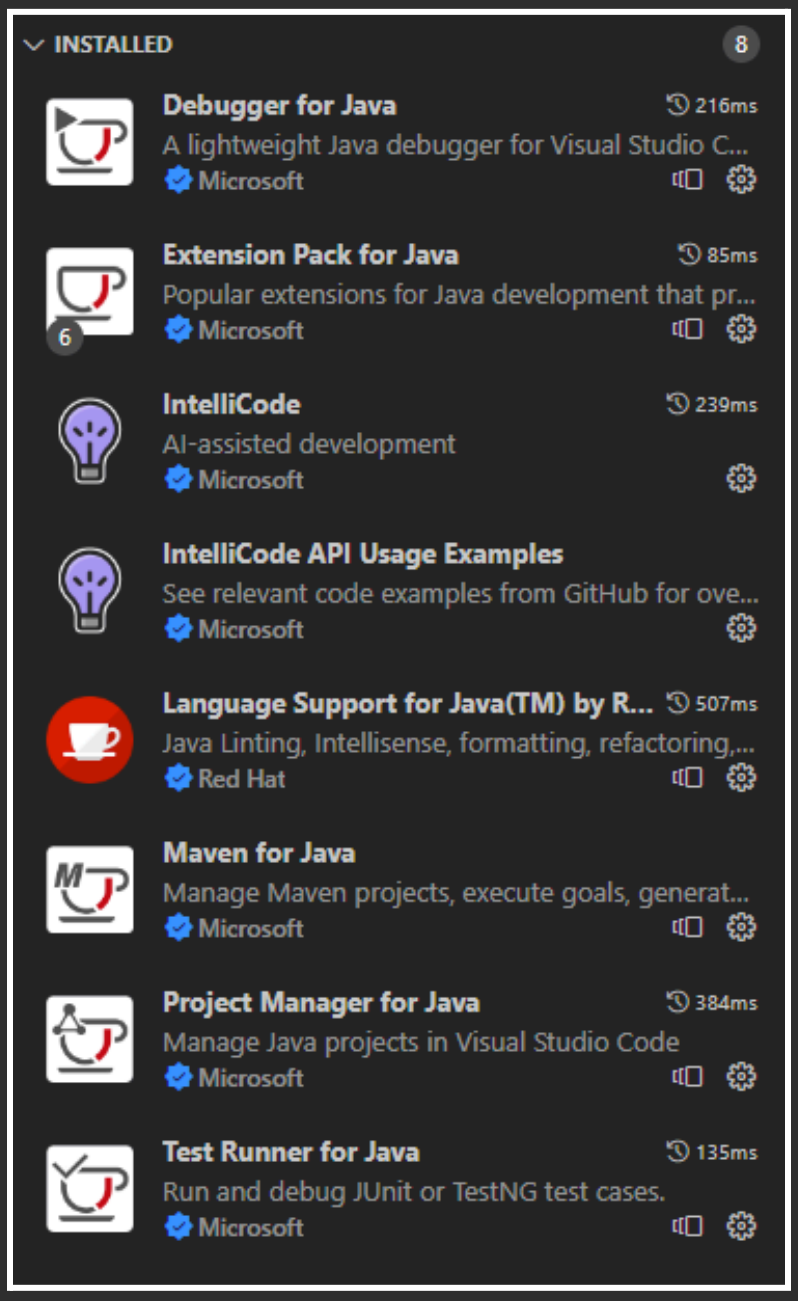
Step 2.1: Required Extensions

1. Extension Pack for Java



a. This installs several extensions to help with Java development.

b. If you clear your search bar you should see the following extensions installed. If you are missing any, search for them and install them.

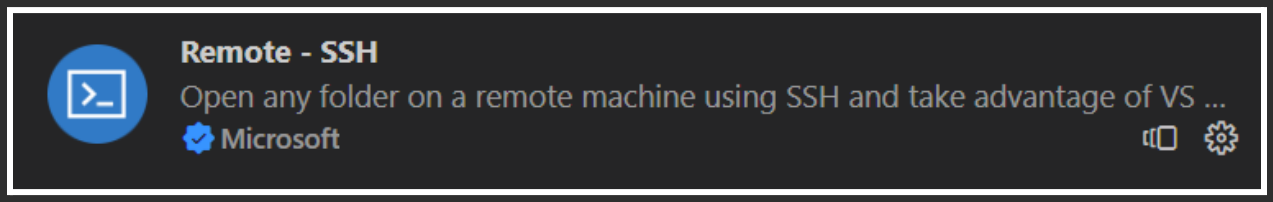


2. vscode-pdf

   a. This will let you view pdf files in vs code.

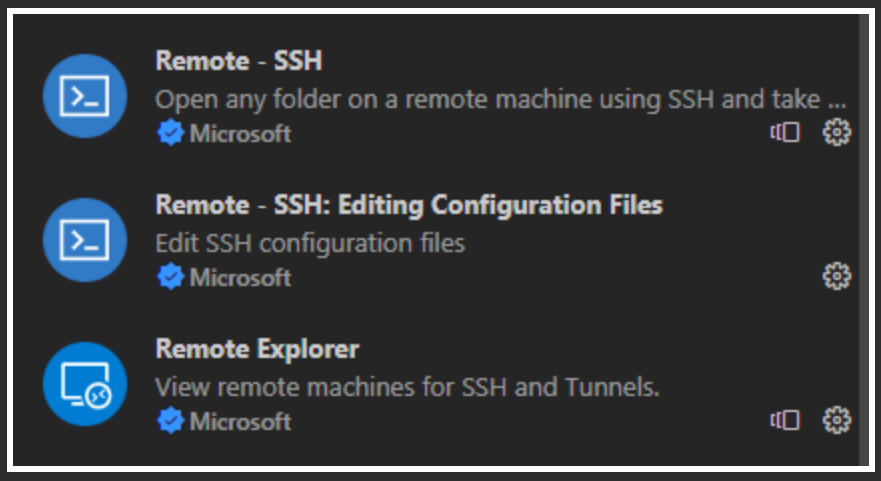
   b. Labs may sometimes include pdf documents. This will let you view them in VSCode rather than needing to open them somewhere else.

3. Remote - SSH



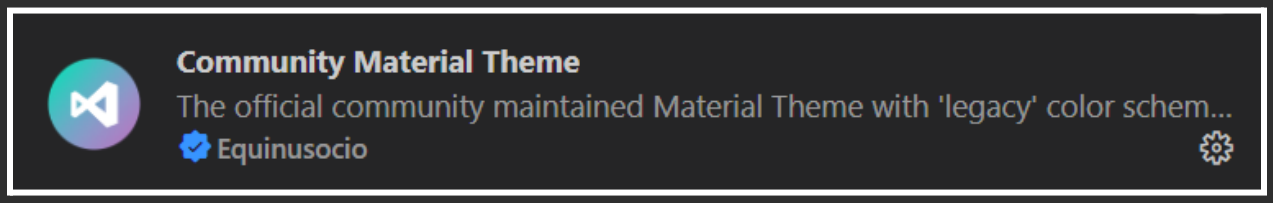
   a. This extension will allow you to connect to the student machine. You will learn more about this in the Remote section of the setup instructions.

   b. After installing, you should see the three following extensions installed. If you are missing any of them, search for and install them.

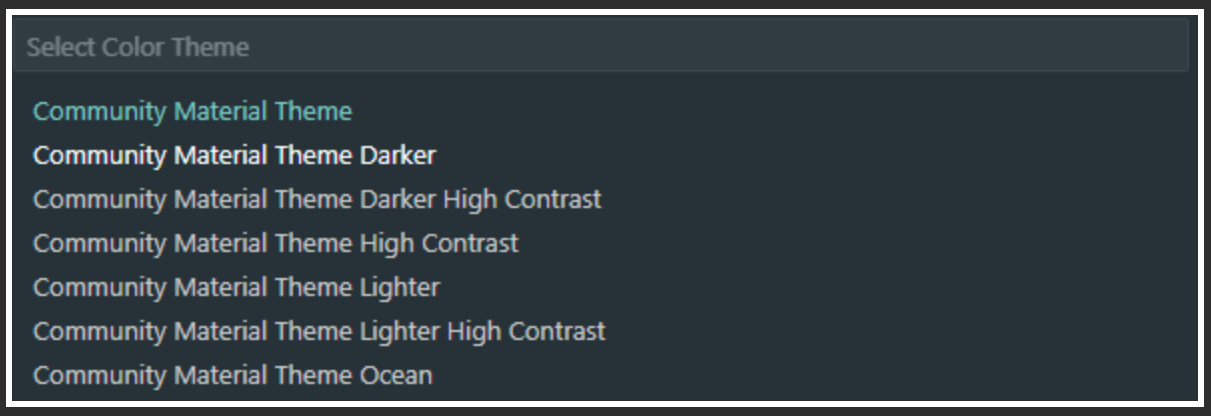


Step 2.2: Optional Extensions

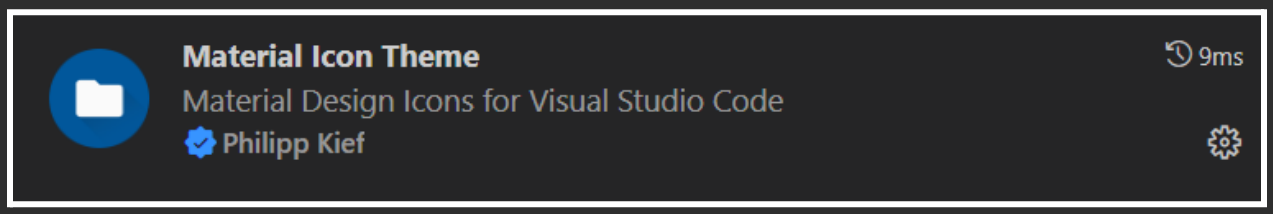
1. Community Material Theme



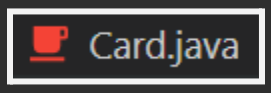
   a. This extension is a color theme for VSCode. You can use whatever theme you prefer, but I use the Community Material Theme Darker.
   b. After installing, you should see a dropdown at the top of VSCode that looks like the following image. Select Community Material Theme Darker if you want your VSCode to look like the screenshots in labs.



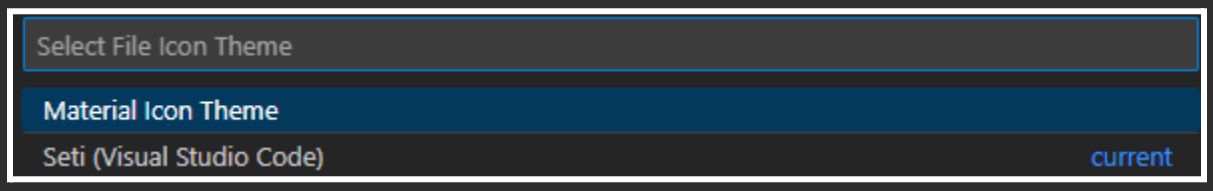
2. Material Icon Theme



a. This extension provides custom icons for files and folders in VSCode. Again, you may use any theme you wish, but this is what will be used in screenshots in the labs. Plus, who doesn't want a coffee cup to be their Java icon?

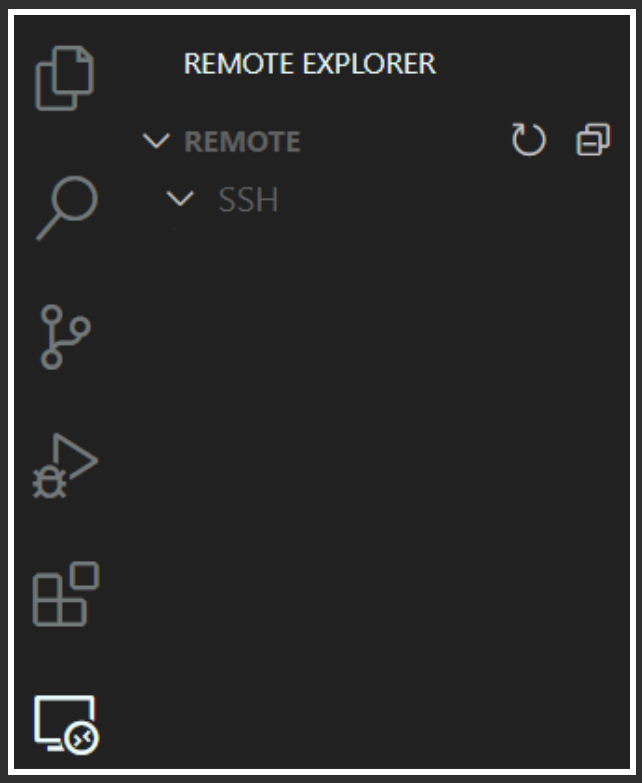


b. After installing, you should see a dropdown at the top of VSCode that looks like the following image. Select Material Icon Theme.



Step 3: Remote SSH Setup

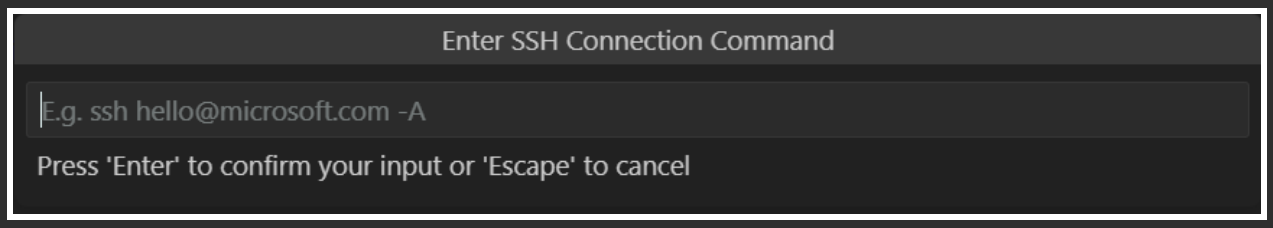
You must be either on asu-secure wifi or connected to the student vpn to use the cs3667 machine. On the left-hand menu in VSCode you should see a new icon after installing Remote - SSH that looks like a computer monitor. Select it and you should see a menu like the image on the right. Hover over the SSH part of the menu and you should see two icons appear as shown in the screenshot below.



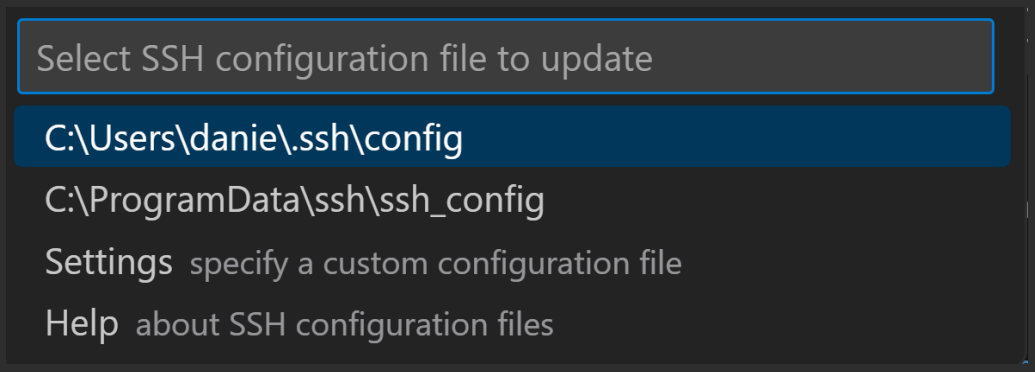
Click on the + icon to set up a connection to a new computer. You will see a menu like the screenshot at the top of the following page. Type in your student username followed by @cs3667.cs.appstate.edu

For example, I would type

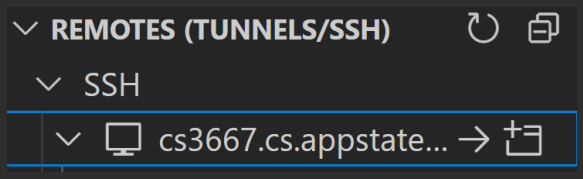
lapenseerankinedl@cs3667.cs.appstate.edu



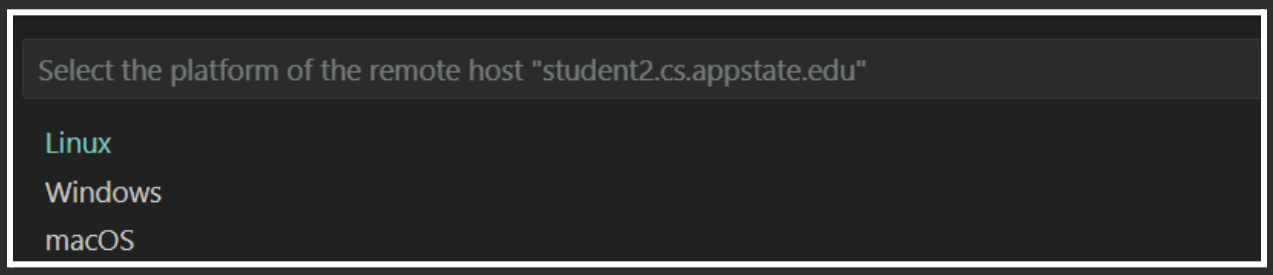
Hit enter and you should see a menu like the following appear.

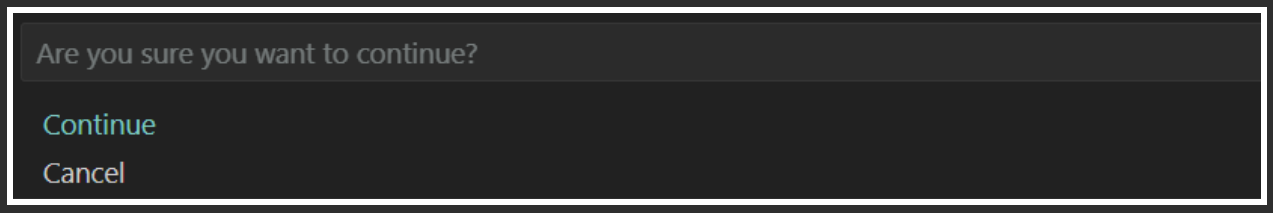


Click enter again (selecting the first option). In the remote menu click the refresh button.

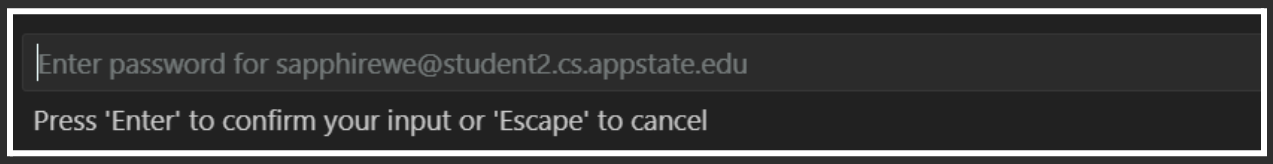


Expand the ssh menu and you should see the new connection in the list. Hover over it and you will see two icons appear. Click the arrow icon. VSCode will refresh and it will ask you the platform of the host. Select Linux.





Yes, you want to continue.



Enter your AsULearn password for the cs3667 machine. If everything goes well, you will now be connected to the cs3667 machine. It may hang, if it does, close VS Code and log in again.

Step 4: Installing Extensions Remotely

You should now be connected to the cs3667 machine, now click the extensions icon on the left-hand menu in VSCode.
You should see a dropdown menu which says “Local - Installed”. If the dropdown is “open” then you should see a list of the extensions you installed in step 2. Go ahead and “close” that dropdown to hide that list of extensions.

Now, you should see another dropdown become visible (it was there all along just far below the list of extensions). This dropdown

is for your remote (cs3540). To the right of that line is a “cloud download” icon. Click that to install all of your local extensions (step 2) onto the remote machine.

Understanding the Remote

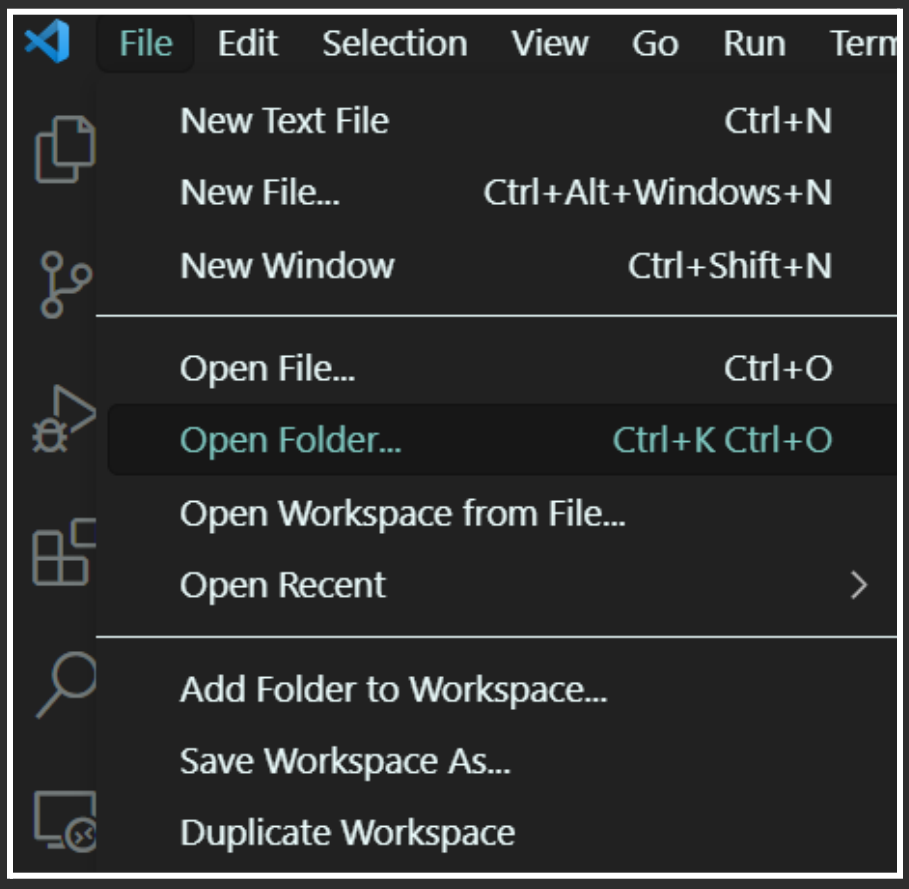
The cs3667 machine that you have connected to through VSCode is a completely different computer. It is not part of VSCode and it is not a program running on your computer. Here are a few notes about what it means to be connected to a remote machine.

- Everything you do on the machine is literally on a different computer.
- You do not need to "save" the remote just like you do not "save" your computer.
- Any files that you create or edit on the remote you do need to save just like you need to save files on your computer
- Treat the remote just like you would treat any computer, just understand that it is somewhere else
- You can only interact with the remote through the terminal. The next section will explain how to do this.

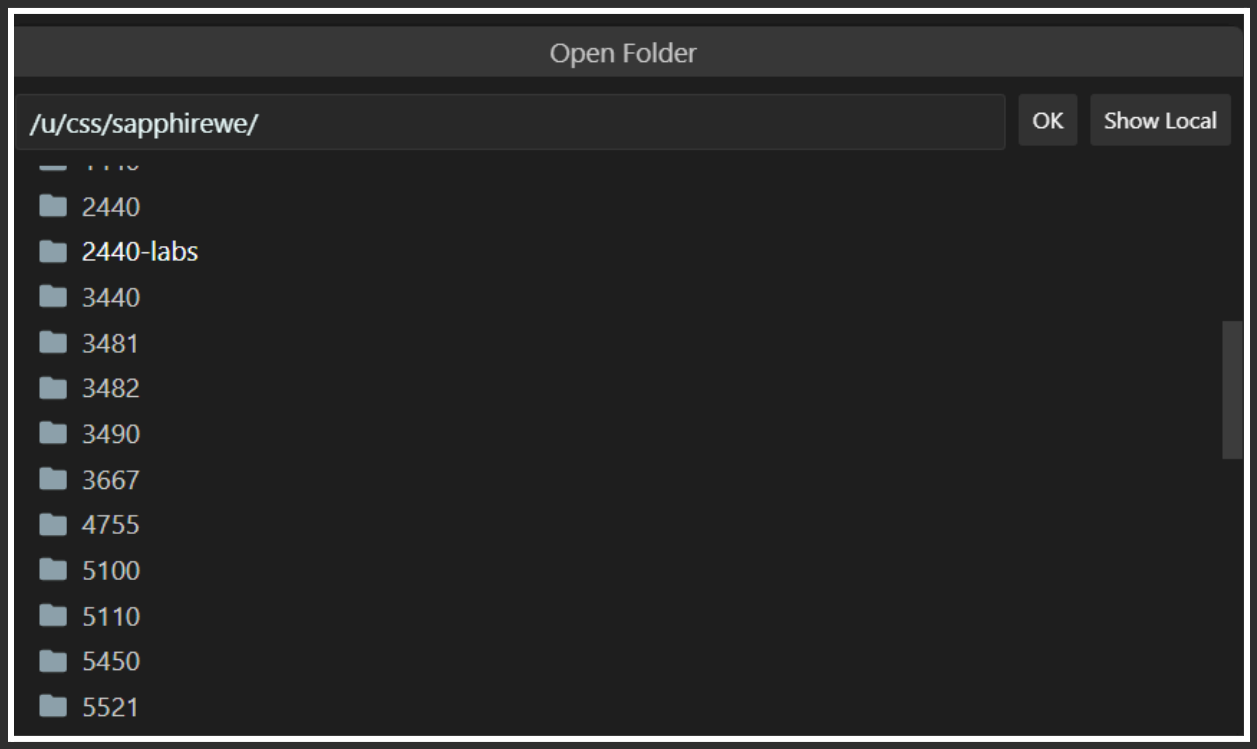
Navigating the Remote Machine

Go here to read the instructions for navigating the cs3667 machine in the terminal. This is very important, so do not continue until you have read through that document.

You should have some folders in your home directory already. You could also create them using the mkdir command (see previous document). To open a folder in VSCode, select the Open Folder option under the File dropdown



You will then see a dropdown like the following image. Click on the folder that you want to open and then click ok.



VSCode will reload and you will likely be prompted for your password again. You will then be able to start creating and editing files in that folder.

Setting Up Git & GitHub

Introduction

We will be using the GitHub website and Git program to manage our code in these labs. You will not be required to learn most git commands, but it is recommended since you will need to learn them in future classes. For this class, we will only be using it to download the starter code for labs. See the links on AsuLearn for some resources to learn more about git.

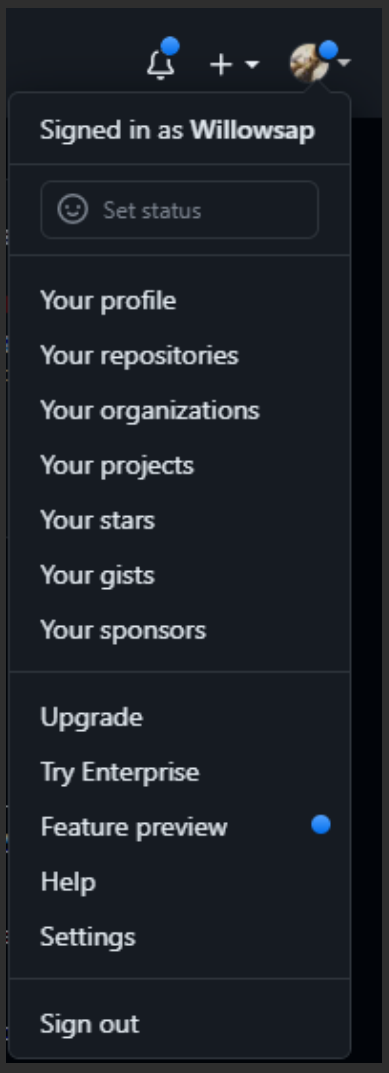
Setting Up GitHub

Creating An Account

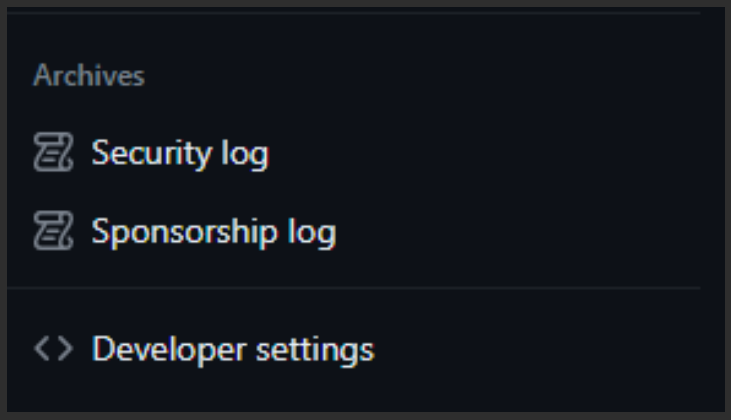
GitHub is a popular website that uses git. Go to https://github.com and create an account if you do not already have one. It is recommended to create one with your student email for school purposes, but you may also use a personal email. The process is fairly straightforward, so there will not be more specific instructions. You do NOT need to select any paid options.

Getting a Personal Authentication Token

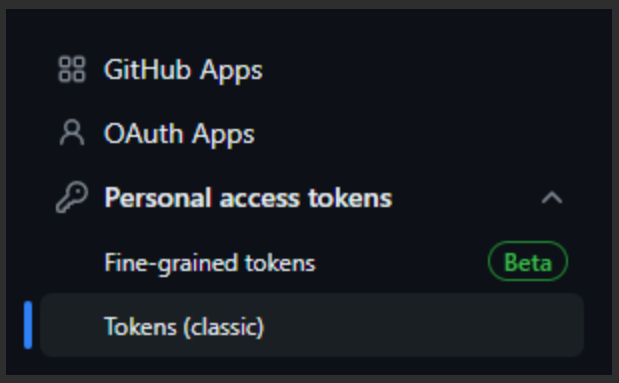
The password that you created with your GitHub account will be used when logging into the website, but you will need something called a personal authentication token to login from the cs3667 machine.



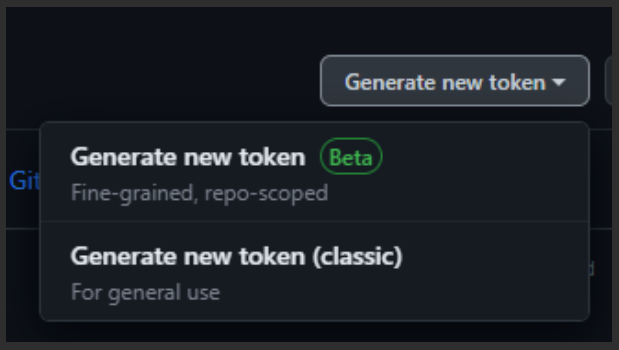
1. Log in to GitHub
2. Click on your profile image in the upper right corner and you should see the dropdown on the right. Click on settings near the bottom.
3. Scroll to the bottom of the menu on the left and select developer settings.



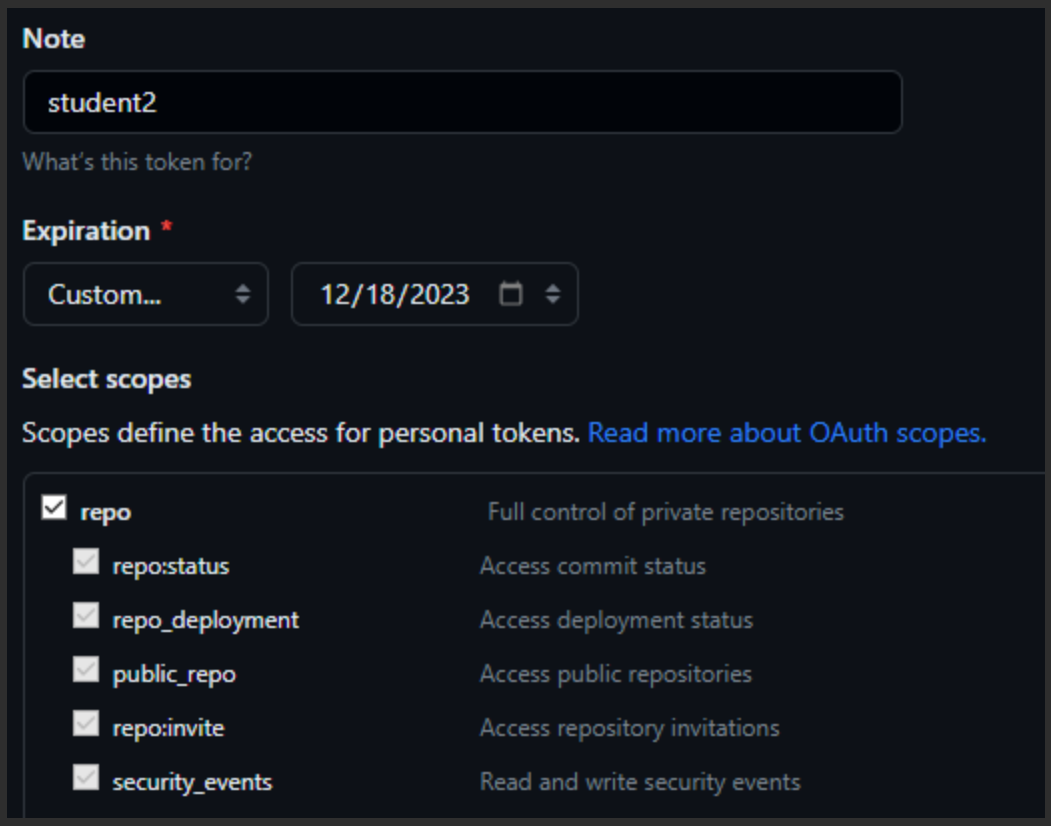
4. Expand the Personal Access Tokens option on the left menu and select Tokens (classic)



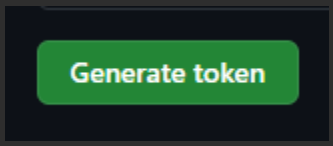
5. Click on Generate new token and select Generate new token (classic)



6. Give your token a name in the Note field. student2 would be a good name since you will use the token on the student2 machine.
7. Give the token an expiration date. It is strongly recommended that you do not select No Expiration. A good date would be December 18th since that is the end of the semester.
8. Under Selected scopes check the repo box.
9. Your screen should look something like the following image.



10. Scroll to the bottom of the page and click Generate token



11. After clicking generate token you will be sent to the page with your tokens and your new token will be visible. It will look like a long string of random characters. Immediately copy this somewhere. You will not be able to get back to this page. I recommend copying it into a text file until you are done with setup, then deleting the text file.

Setting Up Git

Installation

The cs3667 machine that you are connected to already has git installed. You may want to install it on your personal computer as well. Doing that is very simple. Go to the git website, download the installer, and run it. This is not necessary though as we will be working on the cs3667 machine.

Configuration

Although you do not need to install git on the cs3667 machine, you do need to configure it for yourself. First, you need to tell git who you are. Use the following two commands in a terminal on the cs3667 machine.

```
git config --global user.name "<username>"
git config --global user.email "<email>"
```

Replace <username> with your github username and <email> with your github email address. Include the quotation marks but do not include the angle brackets.

Credential File

You should create a credential file so that you will not need to enter your authentication token every time git requires it. Here are the steps to set this up.

1. In your home directory, create a file named .git-cred
    a. You can use `touch ~/.git-cred`
    b. The tilde (~) means your home directory.
    c. Do NOT forget the period at the beginning of the filename. This makes the file invisible.
2. Change the permissions of the file so that you are the only one able to access it.
    a. You can use `chmod 600 ~/.git-cred`
    b. THIS STEP IS IMPORTANT FOR YOUR SECURITY
3. Open the file for editing.
    a. You can use `vim ~/.git-cred`
4. Enter a url in the following form:

    `https://<username>:<token>@github.com`

    a. Replace <username> with your github username and <token> with your personal authentication token
    b. Do not include the angle brackets

    c. In vim, hit 'i' to enter insert mode to start typing in the file
5. Save and exit the file. In vim, hit escape to exit insert mode. Then type :wq and hit enter. This stands for "write" and "quit"
6. Use the git credential helper to store your information using the following command

   ```
   git config --global credential.helper 'store --file ~/.git-cred'
   ```

    a. Note that this is all one line and there is a space between --file and ~/.git-cred
    b. If you named your file something other than .git-cred, use the name of your file.
7. Your credentials are now stored in git.