Collaborating with GitHub and Plugins

The goal of this lab is to get more familiar with coding for your final group projects. I

Goals

- Clone a new repository
- Create a new branch
- Add a new Leaflet plugin
- Create a pull request
- Install and test the live share extension

Lab Overview

1. Cloning a new repository
2. Creating a new branch
3. Adding a new Leaflet plugin
4. Creating a pull request
5. Installing and testing the live share extension

This lab will start by cloning a new repository from the `git practicing repo` here:

- https://github.com/albertkun/23S-ASIAAM-191A-Git-Practicing/

The git link to be cloned is here:

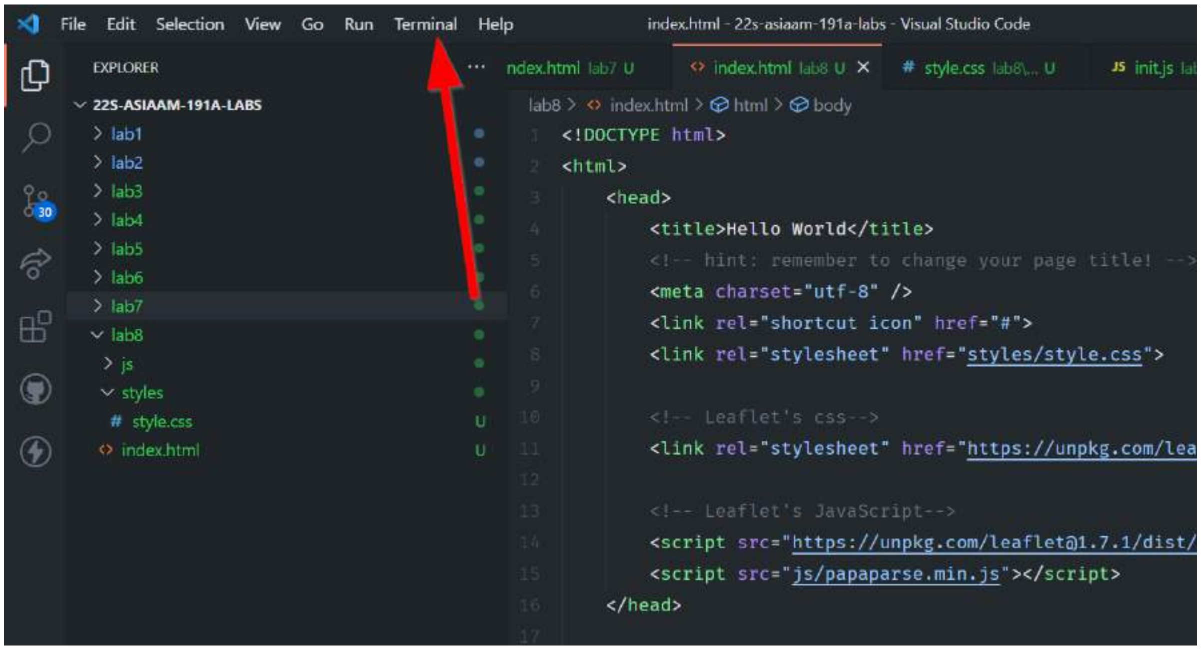
```
https://github.com/albertkun/23S-ASIAAM-191A-Git-Practicing.git
```

Note: This is similar to cloning your group projects repository to your local machine if you have not done so already.

Last update: 2023-05-25

Clone a new repo

We will be working with the terminal a bit today, so let’s open up the terminal by going to the menu bar:

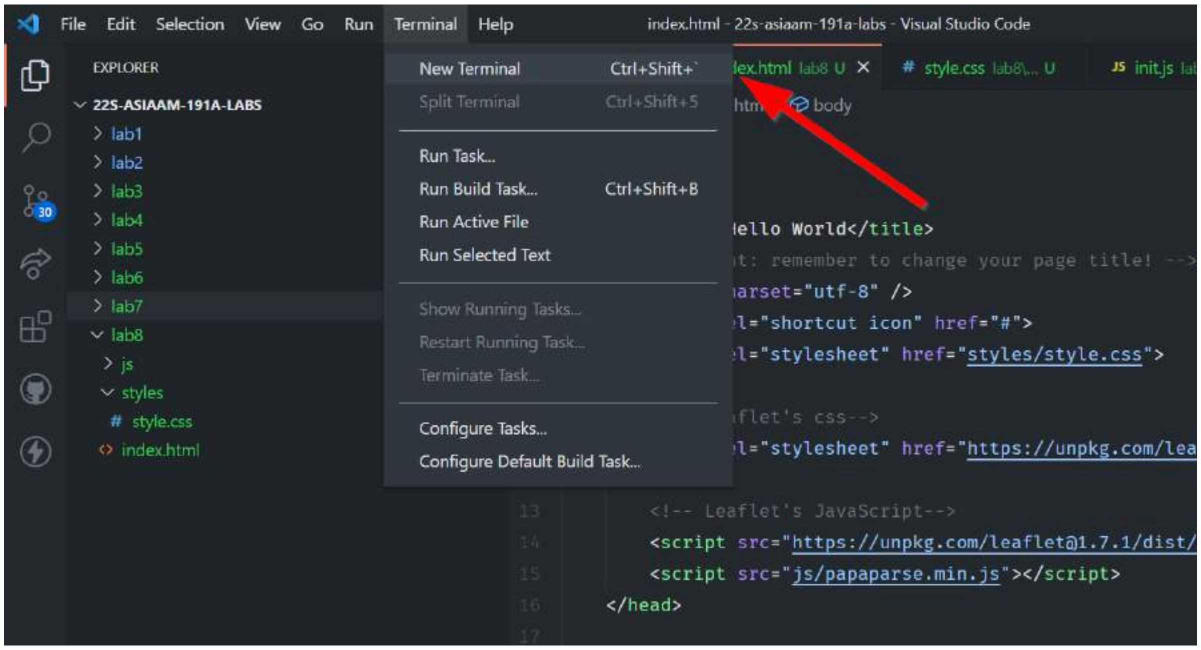


```
<!DOCTYPE html>
<html>
    <head>
        <title>Hello World</title>
        <!-- hint: remember to change your page title! -->
        <meta charset="utf-8" />
        <link rel="shortcut icon" href="#">
        <link rel="stylesheet" href="styles/style.css">

        <!-- Leaflet's css-->
        <link rel="stylesheet" href="https://unpkg.com/lea

        <!-- Leaflet's JavaScript-->
        <script src="https://unpkg.com/leaflet@1.7.1/dist/
        <script src="js/papaparse.min.js"></script>
    </head>
```

Then clicking on `New Terminal`:



```
lello World</title>
t: remember to change your page title! -->
arset="utf-8" />
l="shortcut icon" href="#">
l="stylesheet" href="styles/style.css">

flet's css-->
l="stylesheet" href="https://unpkg.com/lea

    <!-- Leaflet's JavaScript-->
    <script src="https://unpkg.com/leaflet@1.7.1/dist/
    <script src="js/papaparse.min.js"></script>
</head>
```

Start by cloning this repo:

```
git clone https://github.com/albertkun/23S-ASIAAM-191A-Git-Practicing.git
```

Watch where you run terminal commands!

Make sure to note that where you run the terminal command is where the `git clone` will run and thus copy the folder into. DO NOT run this inside of another repository or you will create a lot of problems and break your git capabilities.

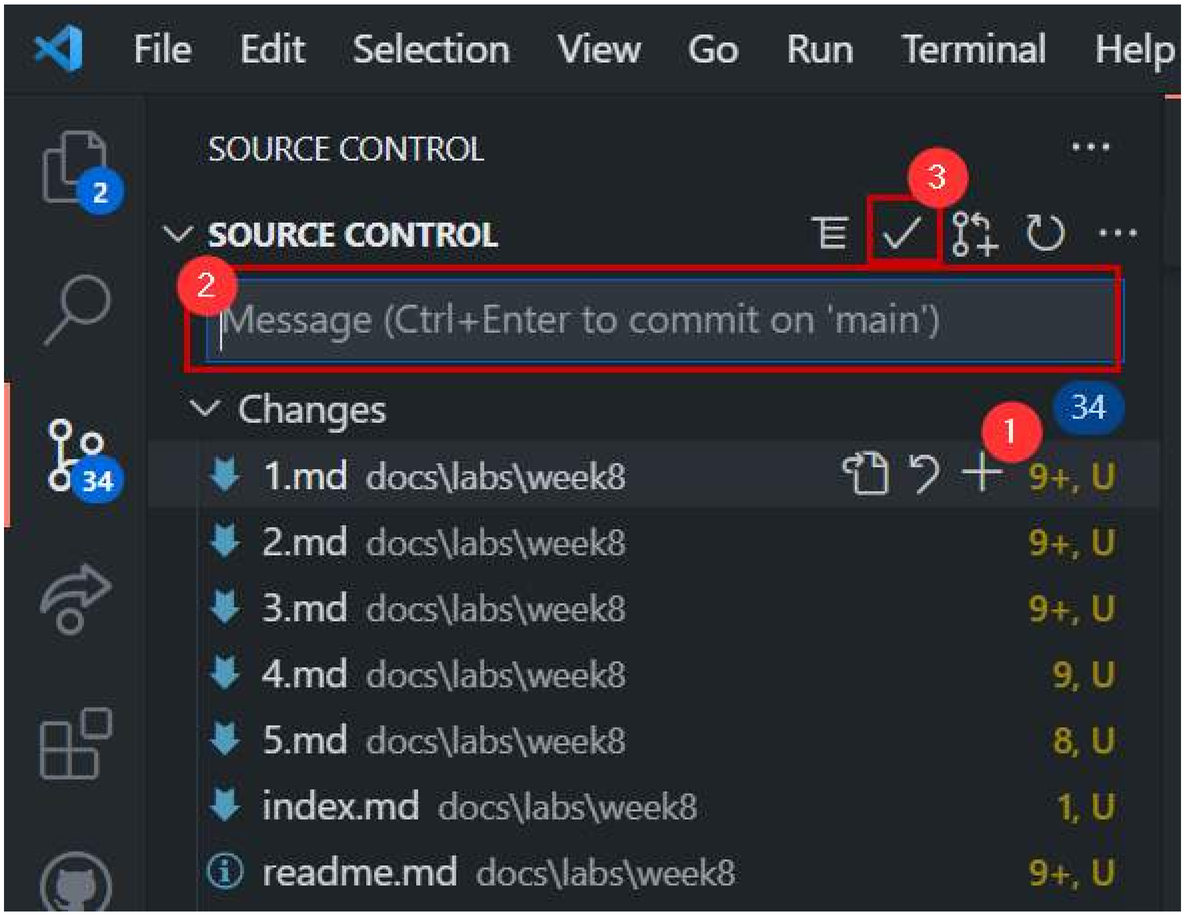
Navigating the Terminal/Command Prompt

- To move up a directory use `cd ..`
- To see what directory and files type in Mac/Linux: `ls` or Windows: `dir`
- To make a directory use `mkdir aNewFolderName`
- To move into a directory use `cd aNewFolderName`

Here are the basic git commands for adding new changes:

```
git add .
git commit -am "message"
git push
```

These commands are identical to what we do in the source control tab in VS Code:


File
Edit
Selection
View
Go
Run
Terminal
Help
SOURCE CONTROL
SOURCE CONTROL
Message (Ctrl+Enter to commit on 'main')
Changes
34
1.md docs\labs\week8 9+, U
2.md docs\labs\week8 9+, U
3.md docs\labs\week8 9+, U
4.md docs\labs\week8 9, U
5.md docs\labs\week8 8, U
index.md docs\labs\week8 1, U
readme.md docs\labs\week8 9+, U
1
2
3

1. Is the `git add .`
2. Is the `git commit` with a `message`
3. Is the `git push`

Last update: 2023-05-25

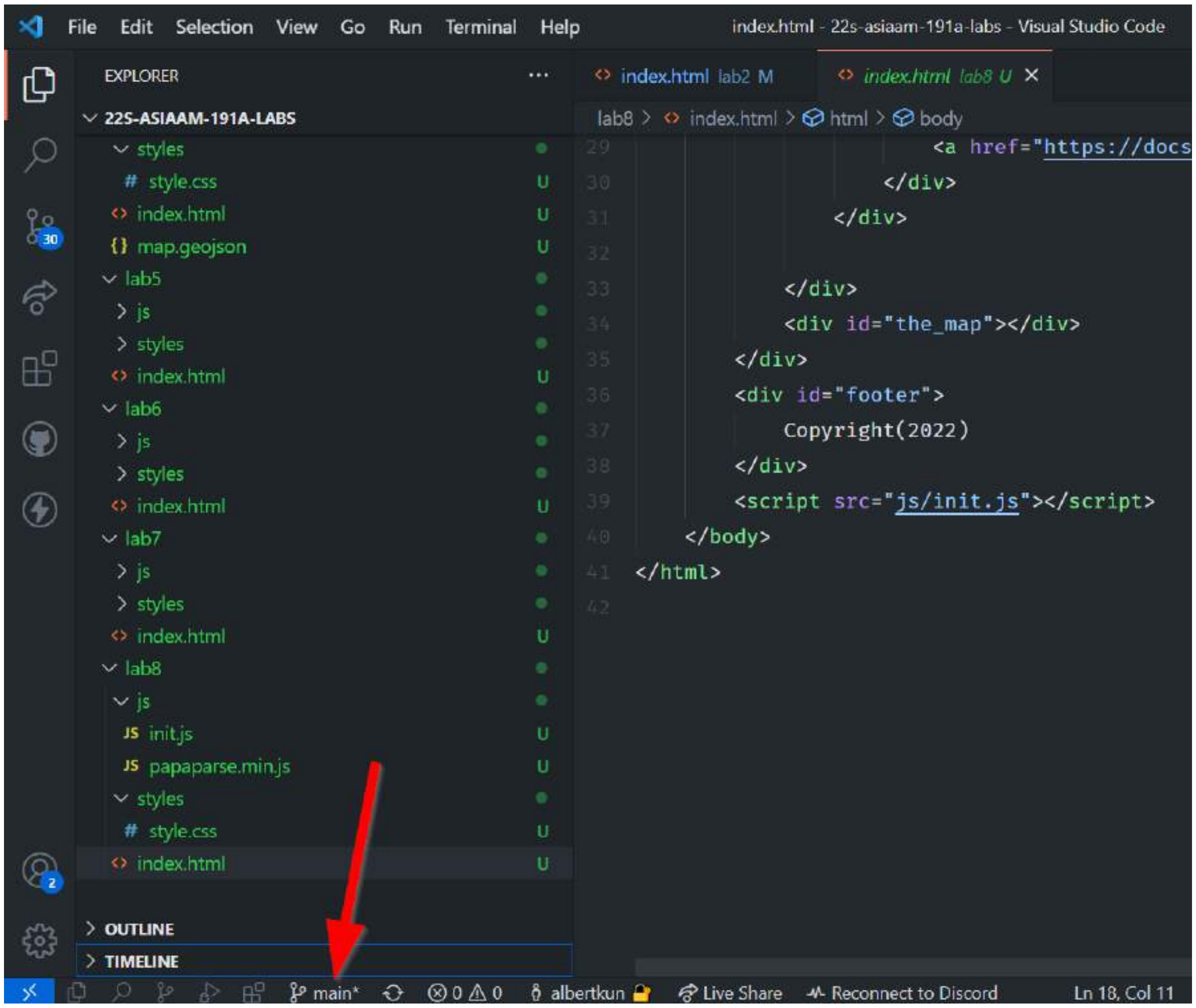
Working with `branches`

Making a new branch

```
git checkout -b helloNewBranch
```

This creates a branch called `helloNewBranch` and switches to it!

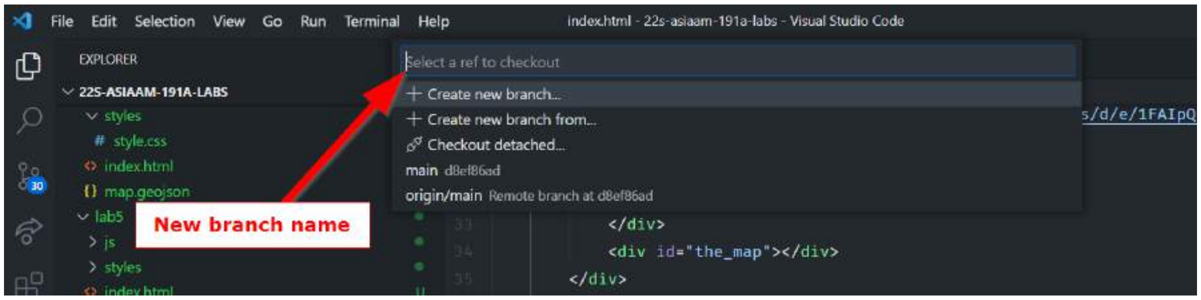
You can also make a new branch in VS Code by clicking this button:



```
                        <a href="https://docs
                    </div>
                </div>

            </div>
            <div id="the_map"></div>
        </div>
        <div id="footer">
            Copyright(2022)
        </div>
        <script src="js/init.js"></script>
    </body>
</html>
```

You can then create a name for it


File Edit Selection View Go Run Terminal Help
index.html - 22s-asiaam-191a-labs - Visual Studio Code
EXPLORER
22S-ASIAAM-191A-LABS
Select a ref to checkout
+ Create new branch...
+ Create new branch from...
Checkout detached...
main d8ef86ad
origin/main Remote branch at d8ef86ad
New branch name

`git add .` your changes to the new branch:

Make some changes and add them to the branch:

```
git add .
```

Add a message to your commit

```
git commit -am "message"
```

Push your changes to your new branch

This code creates a new branch called `helloNewBranch` on GitHub to push to:

```
git push --set-upstream origin helloNewBranch
```

You only need to run it when the branch DOES NOT exist on GitHub!!! After the branch is on GitHub, use only need to use `git push`:

```
git push
```

Updating your branch

Sometimes you want to make sure your branch is up to date, so you can use the following command:

```
git merge <branch_you_want_to_merge>
```

For example this command will `merge` content from `main` to the branch I am currently on:

```
git merge main
```

However!!!

What happens when a `git push` affects in a file that was changed locally but someone else edited on GitHub?

Refer to this medium post to learn more about git merges

Merge Conflicts!!!

A `merge conflict` occurs when one file was changed in two places. For example, Person A edits line 1 of `readme.md` and `Person B` also edits line 1 of `readme.md`. A `git` doesn't know which changes to keep, so a person needs to take a look and manually `merge` them.

First, do a `git pull` which will check if you are behind a commit:

```
git pull
```

When your commit is behind, you may receive this message:

```
error: Your local changes to the following files would be overwritten by merge:
        **SOME FILE(S)**
Please commit your changes or stash them before you merge.
Aborting
Updating 6ac38e2..4dbc13c
```

Do a git commit:

```
git add .
git commit -am "message"
git push
```

After you try to push, this message should pop-up:

```
error: failed to push some refs to 'https://github.com/albertkun/23S-ASIAAM-191A-
Git-Practicing.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.```
```

Run another `git pull`

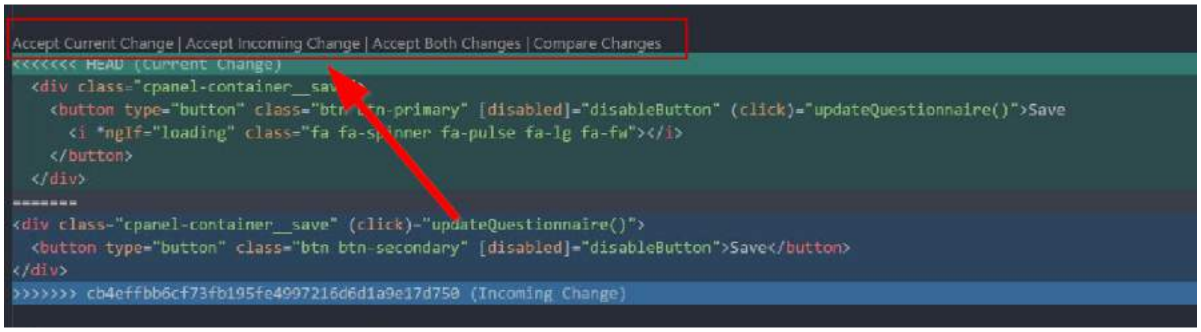
```
git pull
```

If files didn't change at the same time, then auto-merging could take place.

Then proceed to push as normal:

```
git push
```

If files did change at the same time, you have to choose which version to keep:


Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
<<<<<<< HEAD (Current Change)
  <div class="cpanel-container__save">
    <button type="button" class="btn btn-primary" [disabled]="disableButton" (click)="updateQuestionnaire()">Save
      <i *ngIf="loading" class="fa fa-spinner fa-pulse fa-lg fa-fw"></i>
    </button>
  </div>
=======
<div class="cpanel-container__save" (click)="updateQuestionnaire()">
  <button type="button" class="btn btn-secondary" [disabled]="disableButton">Save</button>
</div>
>>>>>>> cb4effbb6cf73fb195fe4997216d6d1a9e17d750 (Incoming Change)
```

After choosing an option, you can can push as normal:

```
git push
```

With a better understanding of `branches` and `merge conflicts`, now we can go ahead and test some new features without worrying about blowing up our repository!

Last update: 2023-05-25

Adding a new Leaflet JavaScript Plugin

The optional part of the previous lab was to change the basemap.

Taking that one step further we can add brand-new functionality to our maps.

While the Leaflet provider is a plug-in of basemaps for Leaflet, there are many plugins that we can use to add extra functionality to our mapplication in JavaScript as well. Here are some examples:

Visualizations

- Turf.js
- Charts

Functions

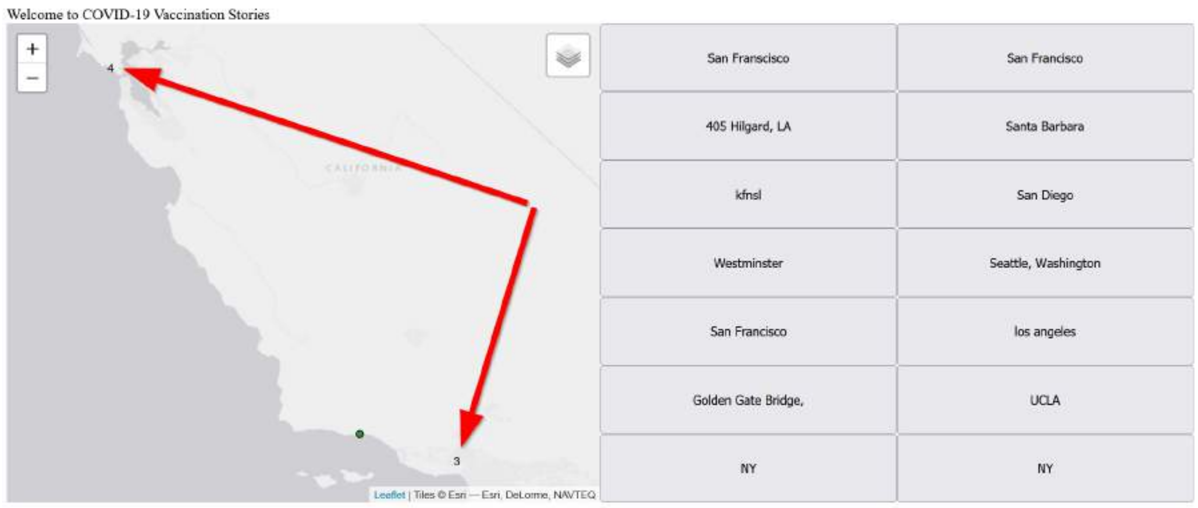
- Scrollama

Leaflet Related

- Leaflet Plugins
- Leaflet Cluster Markers

To keep things simple, we will add a cluster marker functionality to our Leaflet map. Clustering makes it easier to see when multiple points are in the same area.

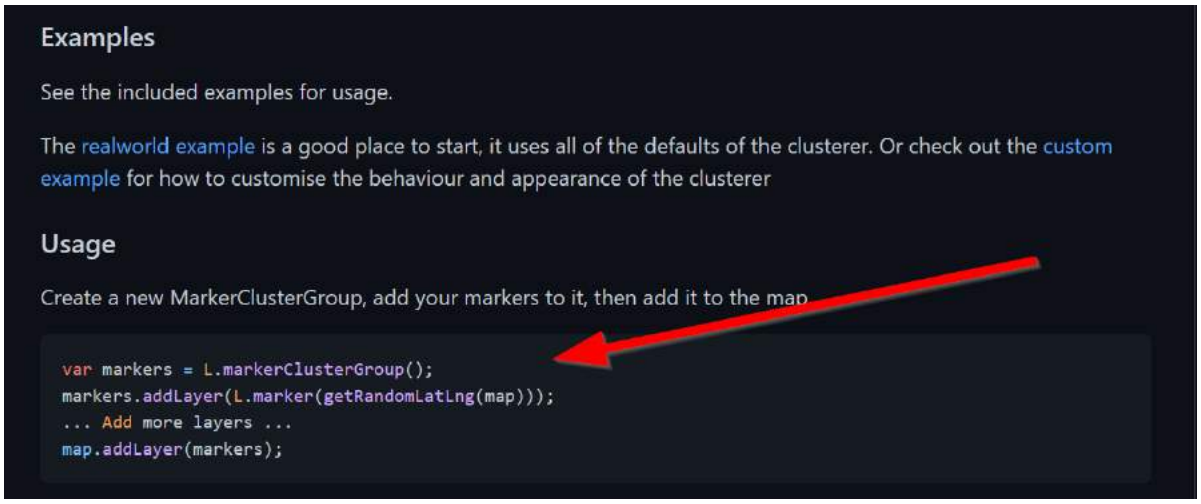
With just a few changes our map will look as follows:



As with when we first used Leaflet we need to include the library, so in our html add the following lines:

> index.html
>
> ```
>         <!-- Cluster Marker's CSS -->
>         <link rel="stylesheet"
>   href="https://unpkg.com/leaflet.markercluster@1.4.1/dist/MarkerCluster.css" />
>         <!-- Cluster Marker's JavaScript -->
>         <script
>   src="https://unpkg.com/leaflet.markercluster@1.4.1/dist/leaflet.markercluster.js">
>   </script>
> ```

Next, let's read the documentation on how to use the `cluster maker`:



Examples

See the included examples for usage.

The realworld example is a good place to start, it uses all of the defaults of the clusterer. Or check out the custom example for how to customise the behaviour and appearance of the clusterer

Usage

Create a new MarkerClusterGroup, add your markers to it, then add it to the map

```
var markers = L.markerClusterGroup();
markers.addLayer(L.marker(getRandomLatLng(map)));
... Add more layers ...
map.addLayer(markers);
```

Judging from this code, we might be able to simply change our group layers for the markers!

Head over to our `init.js` file and find the following lines for our group layers:

js/init.js

```
let vaccinated = L.featureGroup();
let nonVaccinated = L.featureGroup();
```

Change it to:

js/init.js

```
let vaccinated = L.markerClusterGroup();
let nonVaccinated = L.markerClusterGroup();
```

And… wow that's it!

This flexibility is what makes opensource tools and plugins so great! However, be warned that not all plugins will be as simple to plug and play.

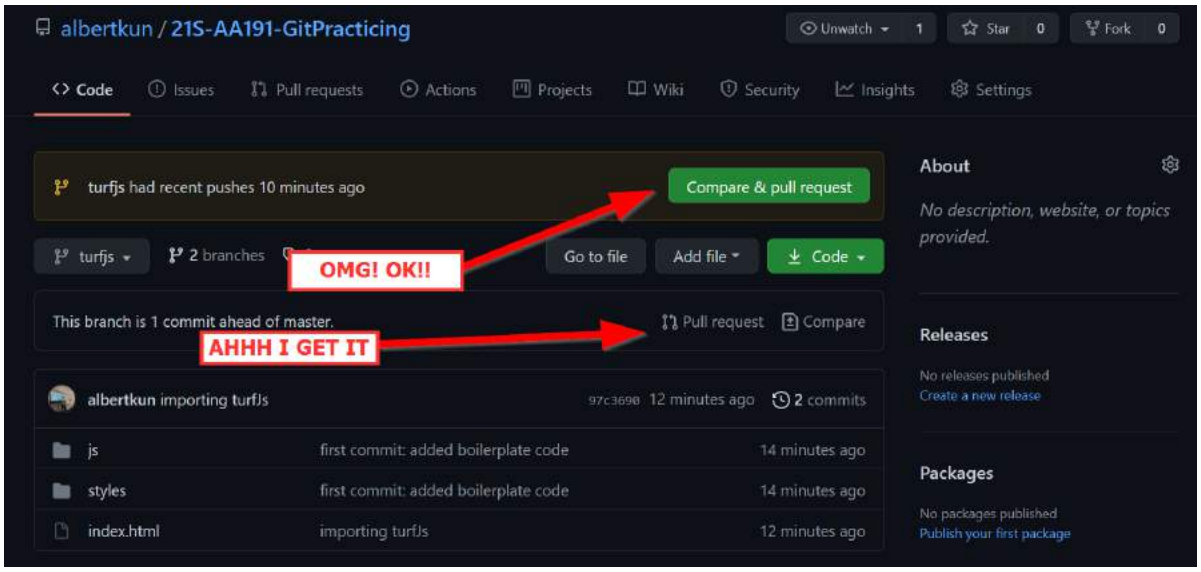
Congrats!

After you've made this change the time has come to make a pull request!

Last update: 2023-05-25

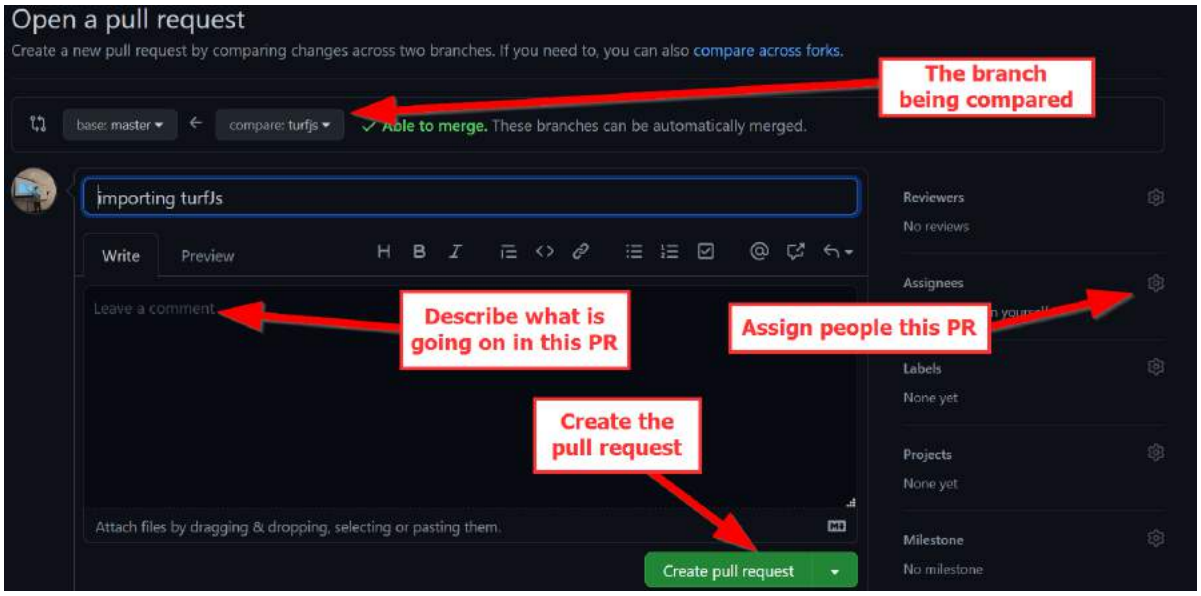
Pull Requests on GitHub

On GitHub you may have seen this nagging icon a few times by now:

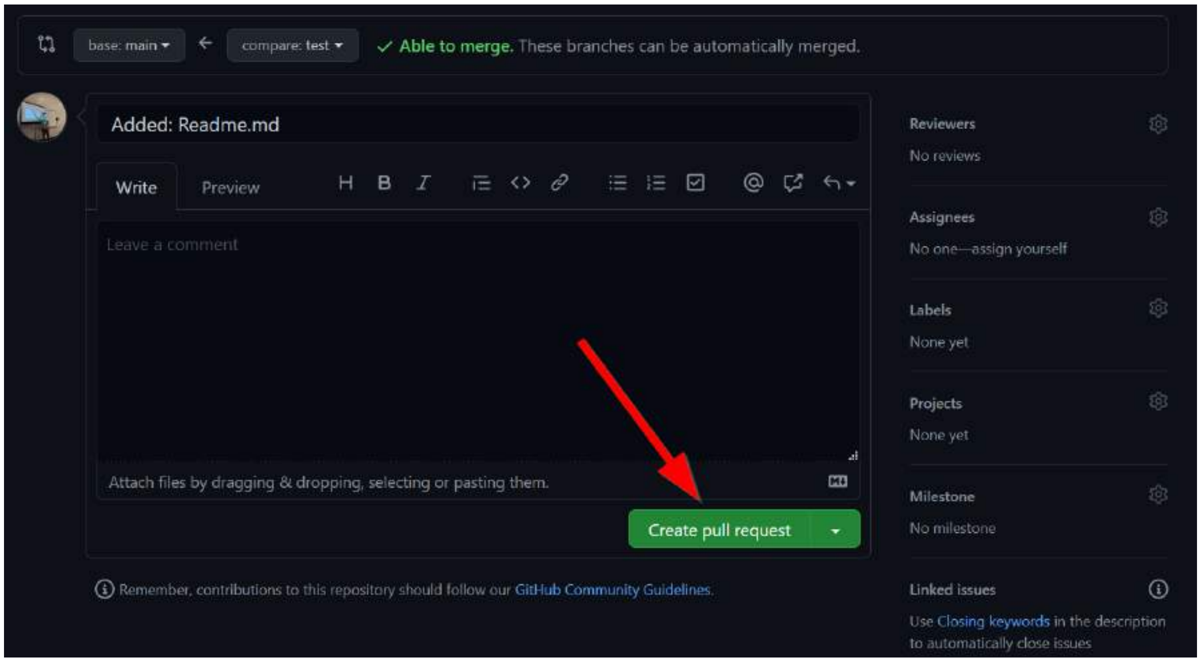


albertkun importing turfJs	97c3698 12 minutes ago	2 commits
js	first commit: added boilerplate code	14 minutes ago
styles	first commit: added boilerplate code	14 minutes ago
index.html	importing turfJs	12 minutes ago

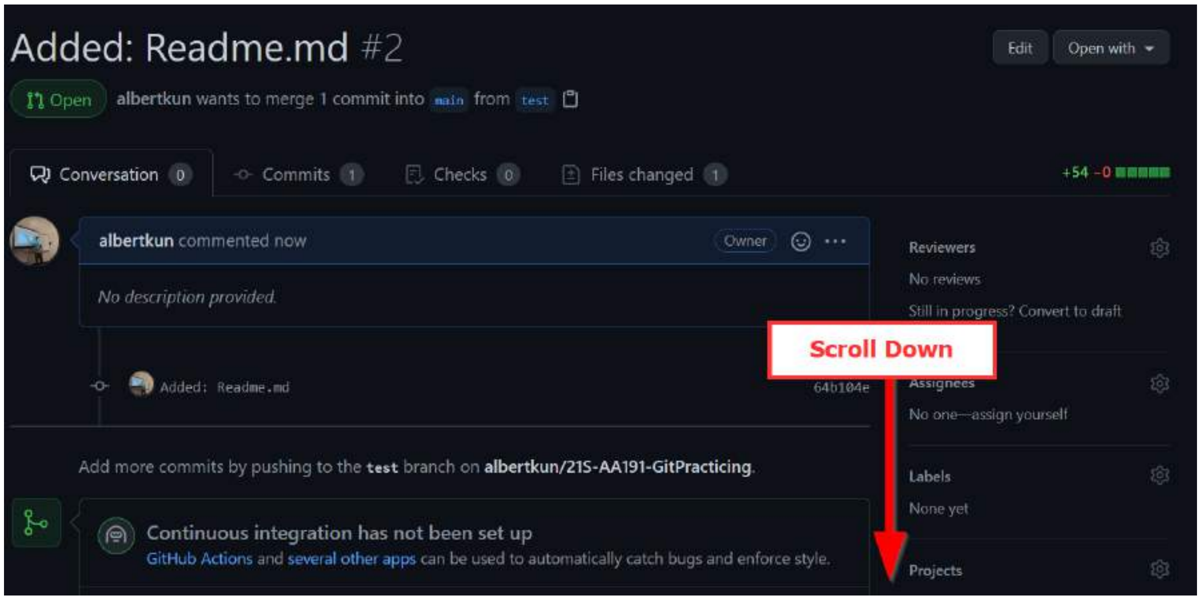
You will then be greeted by a new page where you can title, assign, comment, etc. about the pull request (or PR):


Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.
base: master
compare: turfjs
Able to merge. These branches can be automatically merged.
The branch being compared
importing turfJs
Write
Preview
Leave a comment
Describe what is going on in this PR
Assign people this PR
Create the pull request
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Reviewers
No reviews
Assignees
Labels
None yet
Projects
None yet
Milestone
No milestone

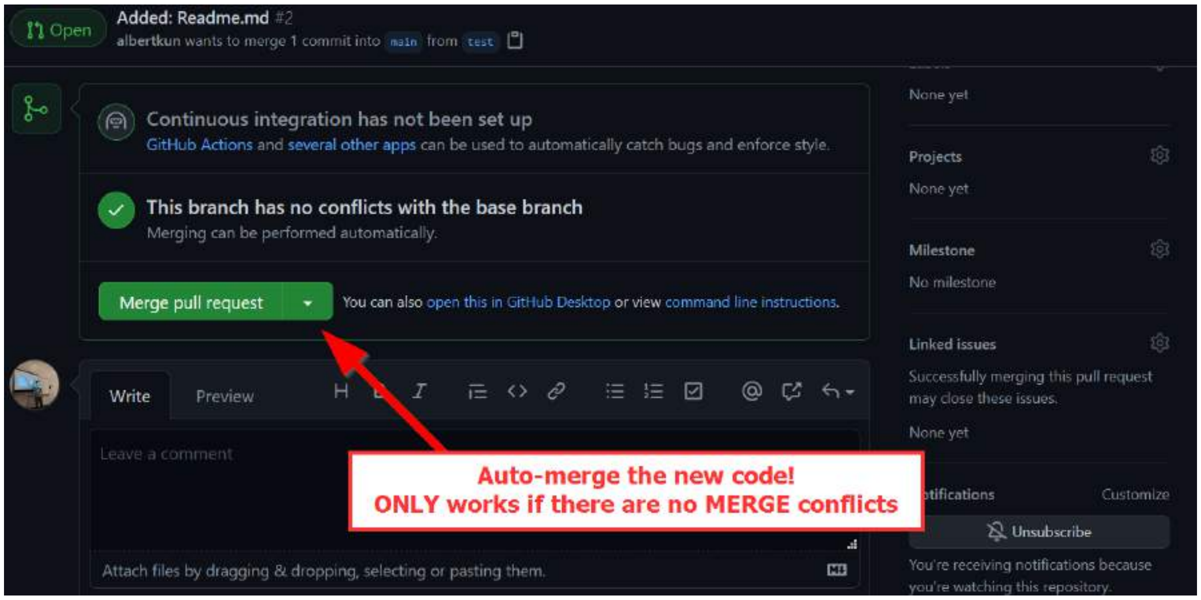
and most importantly create a `pull request`:


base: main
compare: test
✓ Able to merge. These branches can be automatically merged.
Added: Readme.md
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributions to this repository should follow our GitHub Community Guidelines.
Reviewers
No reviews
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Linked issues
Use Closing keywords in the description to automatically close issues

After clicking the button to `create a new pull request` scroll down to the resulting page:


Added: Readme.md #2
Edit
Open with
Open
albertkun wants to merge 1 commit into main from test
Conversation 0
Commits 1
Checks 0
Files changed 1
+54 -0
albertkun commented now
Owner
No description provided.
Added: Readme.md
64b104e
Scroll Down
Add more commits by pushing to the test branch on albertkun/21S-AA191-GitPracticing.
Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.
Reviewers
No reviews
Still in progress? Convert to draft
Assignees
No one—assign yourself
Labels
None yet
Projects

You should be able to click on `merge` if your `pull request` has no `merge conflicts`:



Added: Readme.md #2

albertkun wants to merge 1 commit into `main` from `test`

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

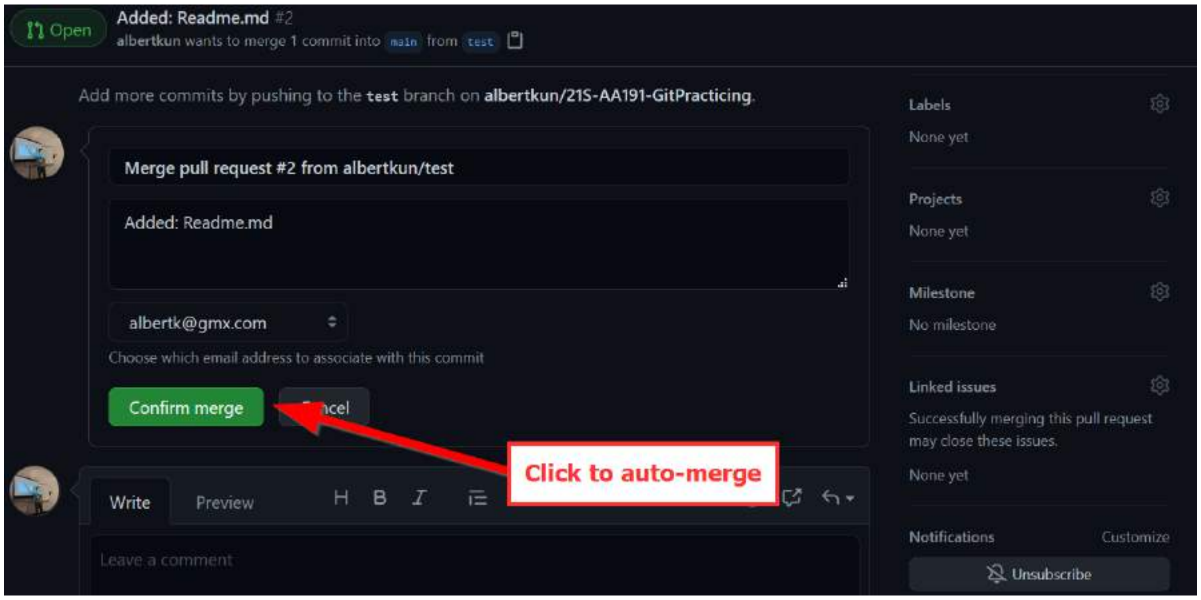
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Successfully merging this pull request may close these issues.

You're receiving notifications because you're watching this repository.

Click to `confirm` the auto merge:



Added: Readme.md #2

albertkun wants to merge 1 commit into `main` from `test`

Add more commits by pushing to the **test** branch on **albertkun/21S-AA191-GitPracticing**.

Merge pull request #2 from albertkun/test

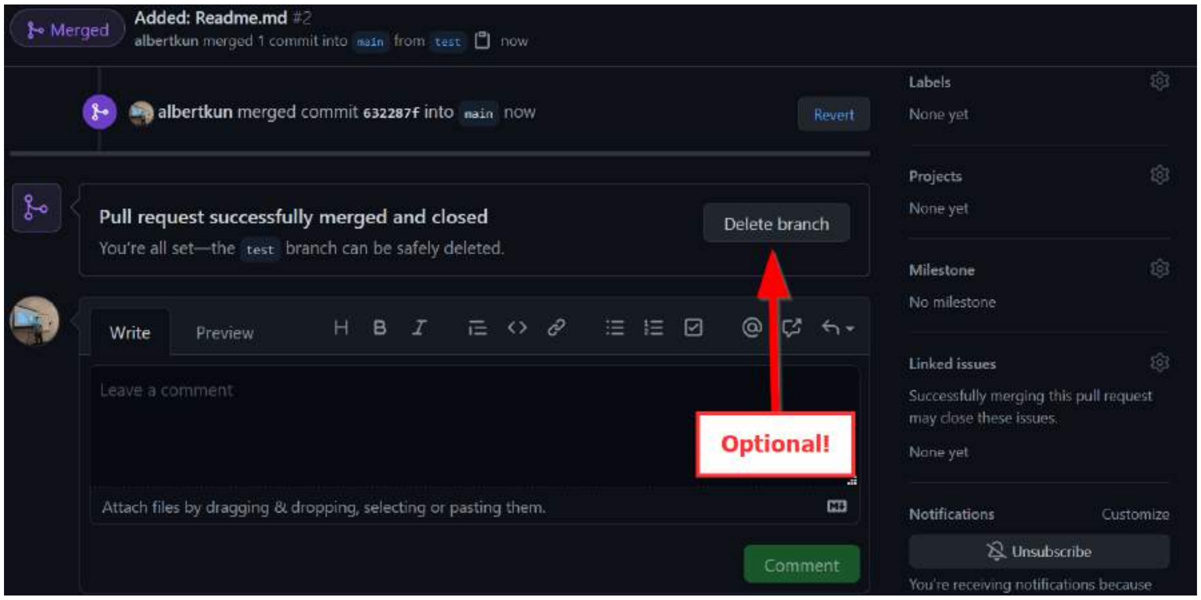
Added: Readme.md

albertk@gmx.com

Choose which email address to associate with this commit

Successfully merging this pull request may close these issues.

And now you can delete the branch:


Merged
Added: Readme.md #2
albertkun merged 1 commit into main from test now
albertkun merged commit 632287f into main now
Revert
Pull request successfully merged and closed
You're all set—the test branch can be safely deleted.
Delete branch
Optional!
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment
Labels
None yet
Projects
None yet
Milestone
No milestone
Linked issues
Successfully merging this pull request may close these issues.
None yet
Notifications
Customize
Unsubscribe
You're receiving notifications because

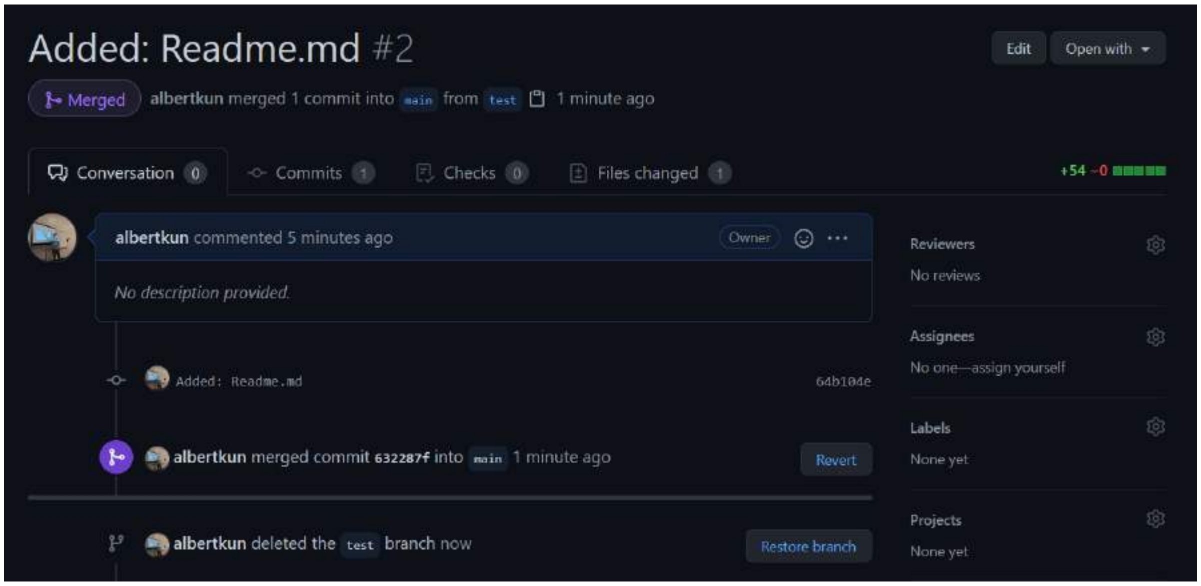
Warning: Merge Conflicts do not allow you to auto-merge a Pull Request!

Warning! Merge Conflicts pvrevent auto-merges of a Pull Request!

You will be unable to `auto-merge` if there is a merge conflict, so refer to the `merge-conflict` steps in order to finish the `pull request`.

Completed Pull Request

Your completed pull request should look like the following:



Added: Readme.md #2

Merged albertkun merged 1 commit into `main` from `test` 1 minute ago

Conversation 0 · Commits 1 · Checks 0 · Files changed 1 · +54 −0

albertkun commented 5 minutes ago (Owner)

No description provided.

Added: Readme.md 64b104e

albertkun merged commit 632287f into `main` 1 minute ago

albertkun deleted the `test` branch now

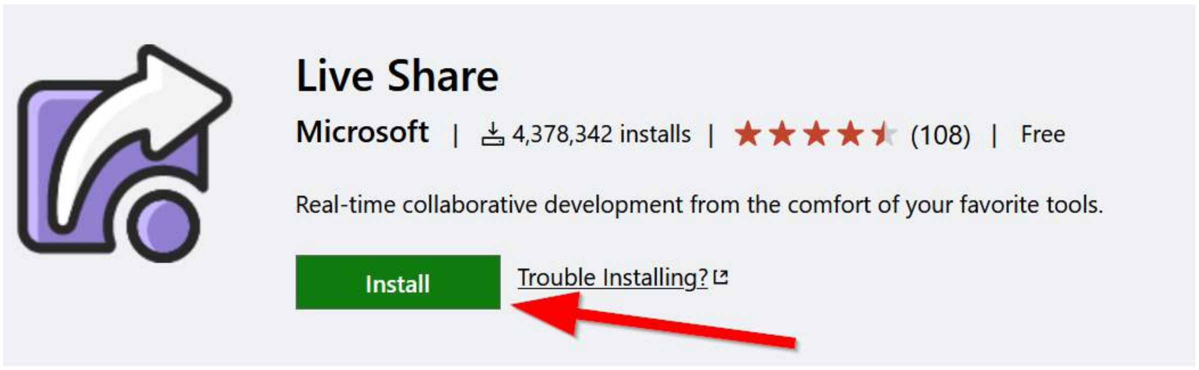
Reviewers: No reviews
Assignees: No one—assign yourself
Labels: None yet
Projects: None yet

Last update: 2023-05-25

`Live Share` The VS Code Plugin for Collaboration

Make sure you have installed the Live Share extension by going to the following link and clicking on "Install":

> https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare


Live Share
Microsoft | 4,378,342 installs | ★★★★½ (108) | Free
Real-time collaborative development from the comfort of your favorite tools.
Install
Trouble Installing?

You can read the documentation to learn more about Live Share and what it does too:

> https://docs.microsoft.com/en-us/visualstudio/liveshare/

After installing Live Share, you can join a session with these steps:

Joining a live share session

1. Click on the Join url

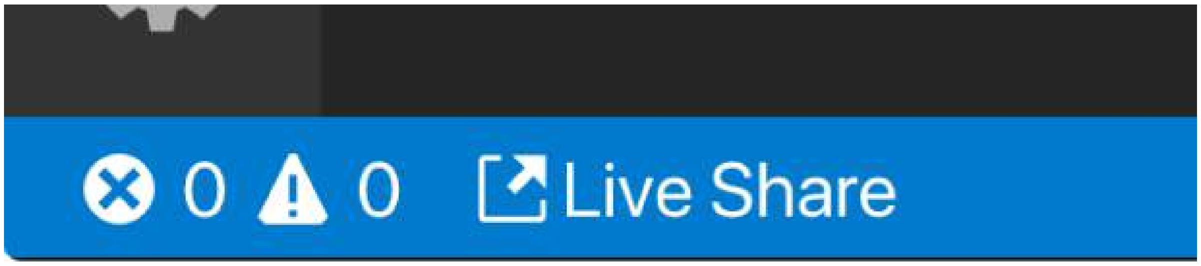
Click the session URL the **host** sent you, which will open it up in a browser. When prompted, allow your browser to launch VS Code

2. Sign in to GitHub

Tip

This will only need be done once.

Click on the `Live Share` status bar item **or** press `Ctrl+Shift+P` / `Cmd+Shift+P` and select the `Live Share: Sign In With Browser` command.

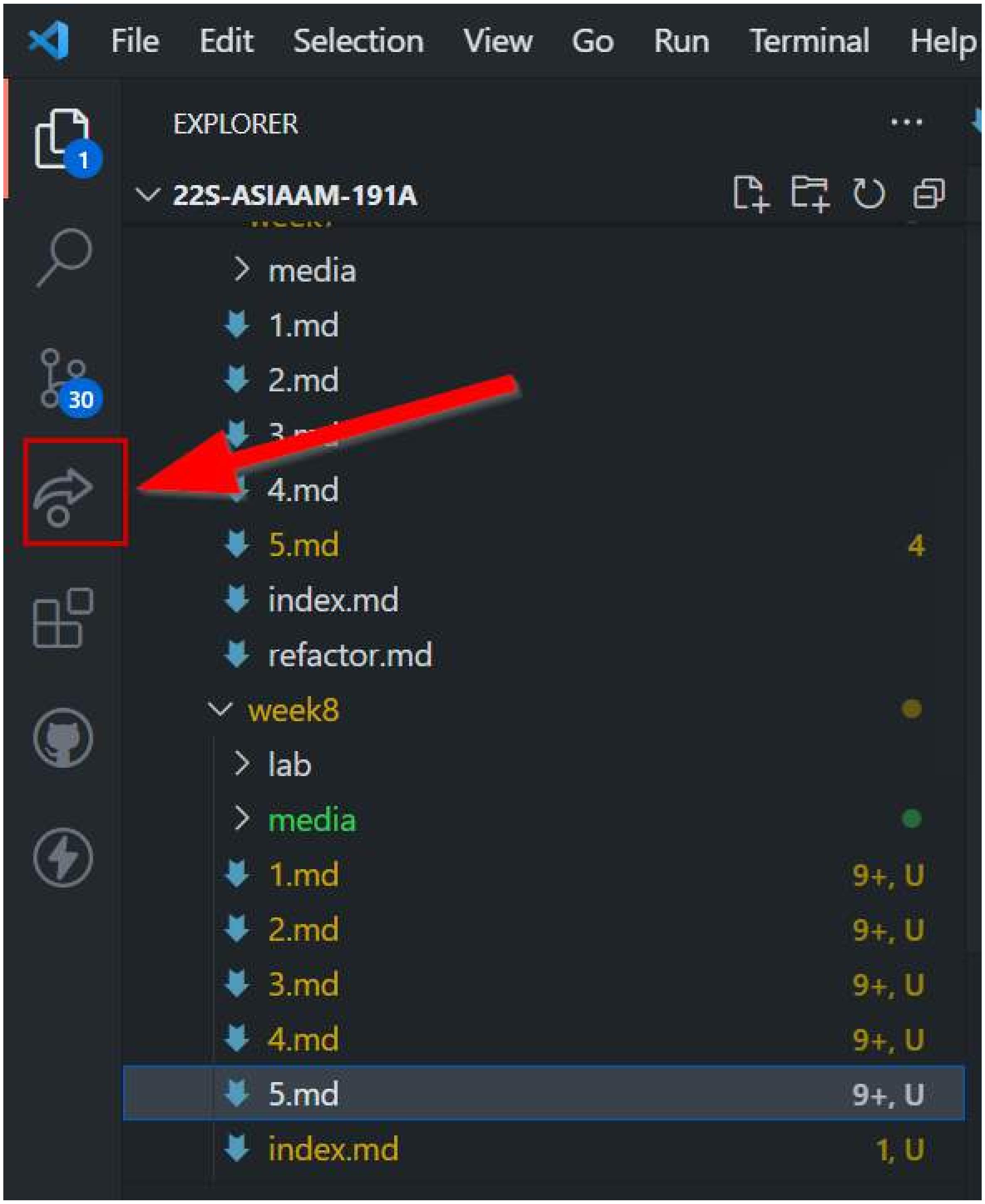


3. Working the session

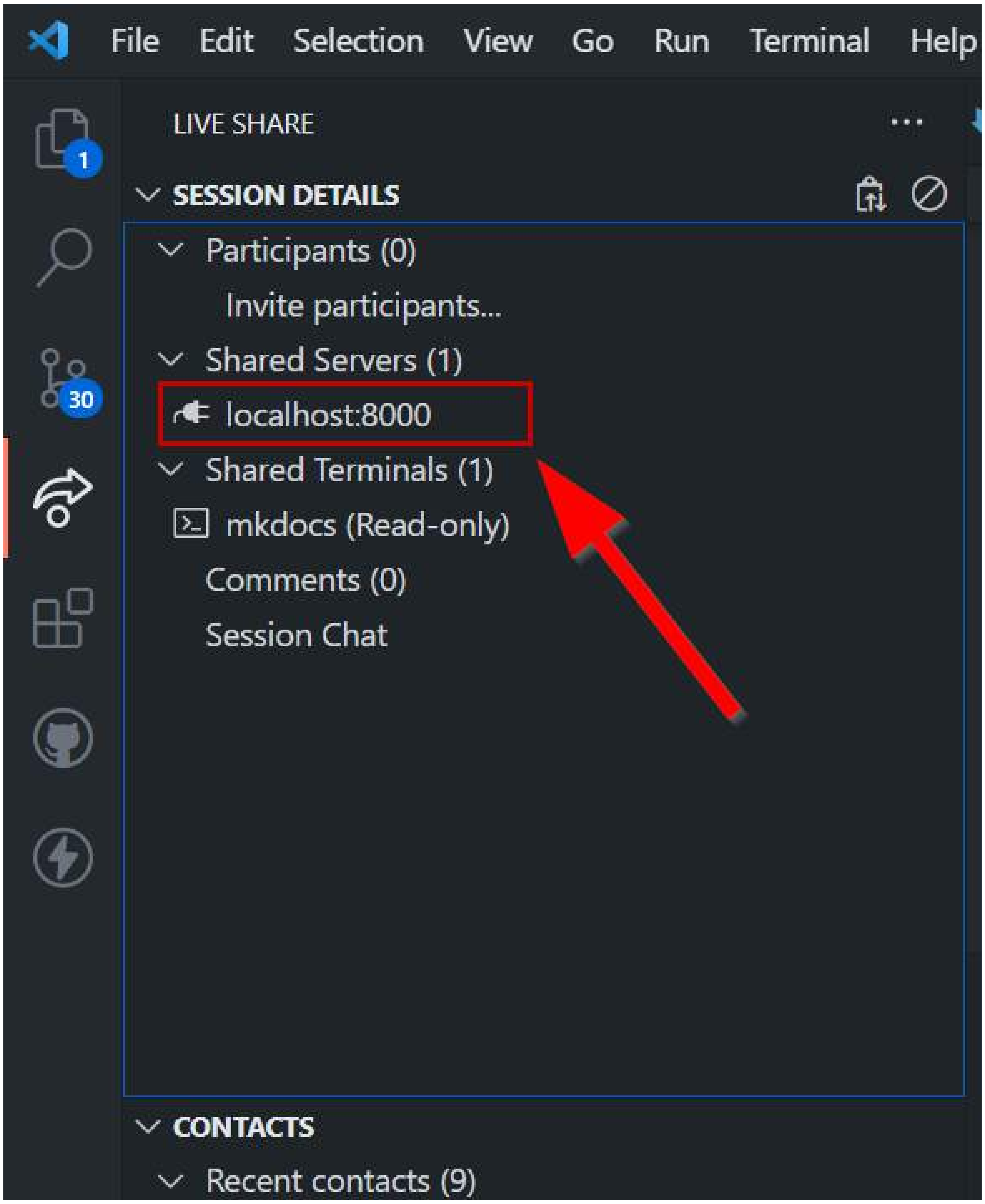
After you join, you'll be immediately presented with the file that the "host" has open, and can see their cursor and any edits they make.

4. Viewing a live server

If the **host** is sharing a live server of their website, you can view it on your local machine by clicking on the live share button:



Then you can click on any server under `Shared Servers`, such as `localhost:8000` to open it:

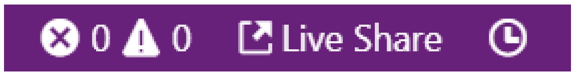


Hosting a live share session

Live Server and Live Share

To make previewing content easier, always remember to start a `Live Server` before starting a `Live Server` session!

Click the “live share” button to immediately start sharing your coding session.

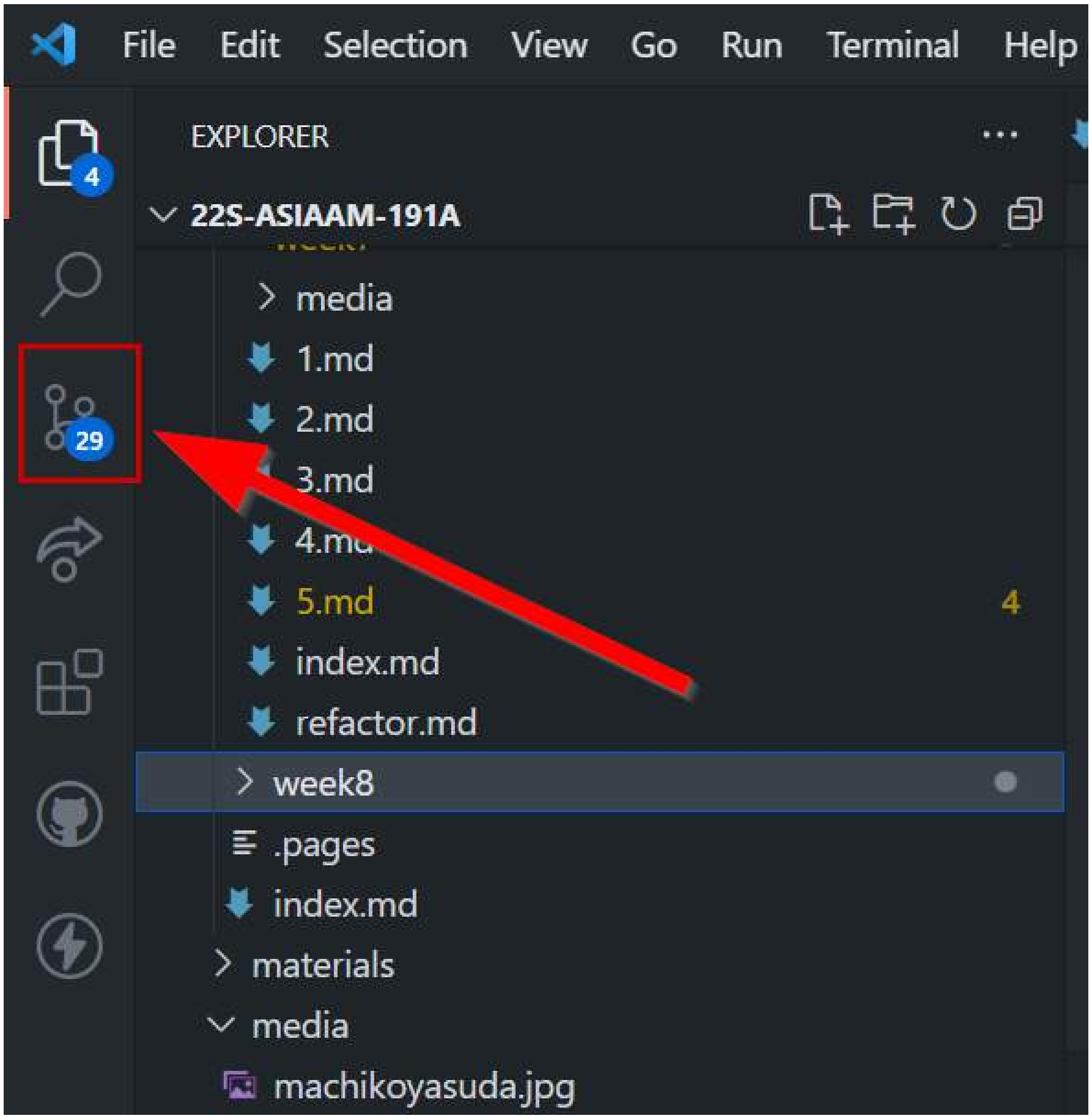


An invitation link will automatically be copied to your clip board, you can invite anyone to join your session by sharing it.

You can join your own collaboration session by clicking the link yourself. open it in any browser to join your session.

Saving Changes

The host can **commit** changes by clicking on the source control tab in VS Code when they are done with the session.



🏁Final Checkpoint

1. Make sure you have live share and live server installed to make collaboration easier.
2. Be sure to have made a pull request to the GitPracticing repo!

Congrats on finishing the lab! There is no assignment, so you can focus on the Group Assignment #5 due next week.