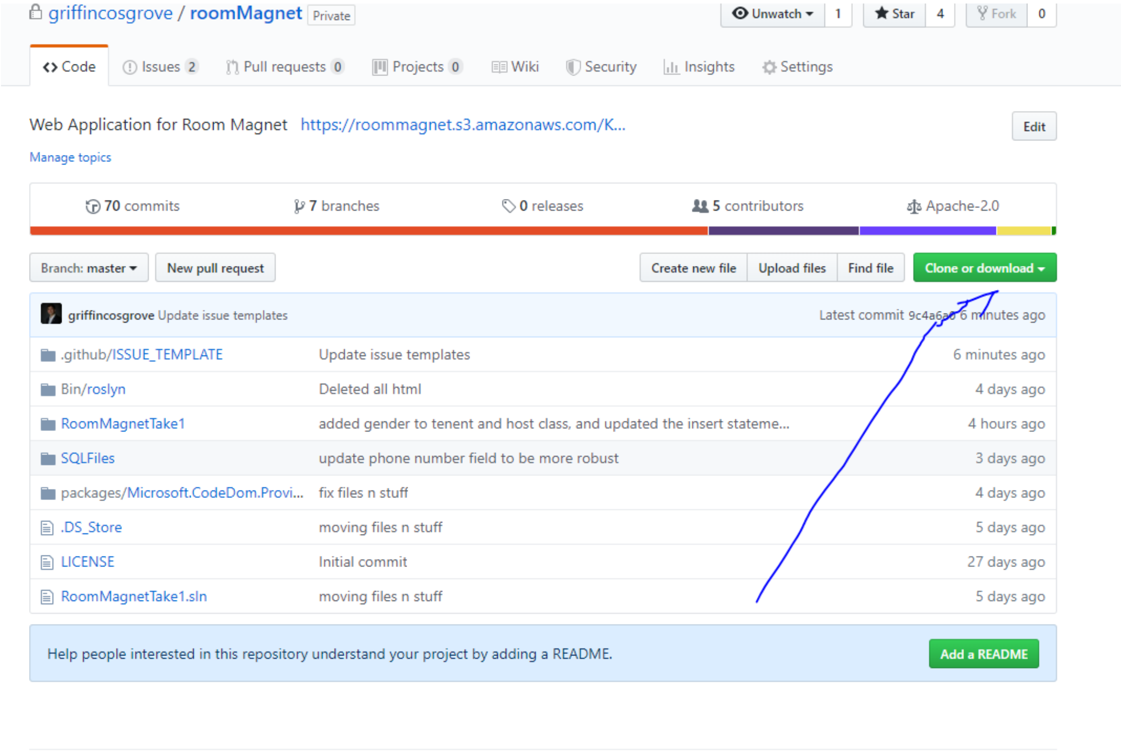
1. Download git
2. make sure you include git bash on the installation, this is default.
3. we will do most of the work from the command line. because that is what i know how to do :)
4. click https://github.com/griffincosgrove/roomMagnet
5. 

griffincosgrove / **roomMagnet** Private

Web Application for Room Magnet https://roommagnet.s3.amazonaws.com/K...

Manage topics

70 commits **7** branches **0** releases **5** contributors Apache-2.0

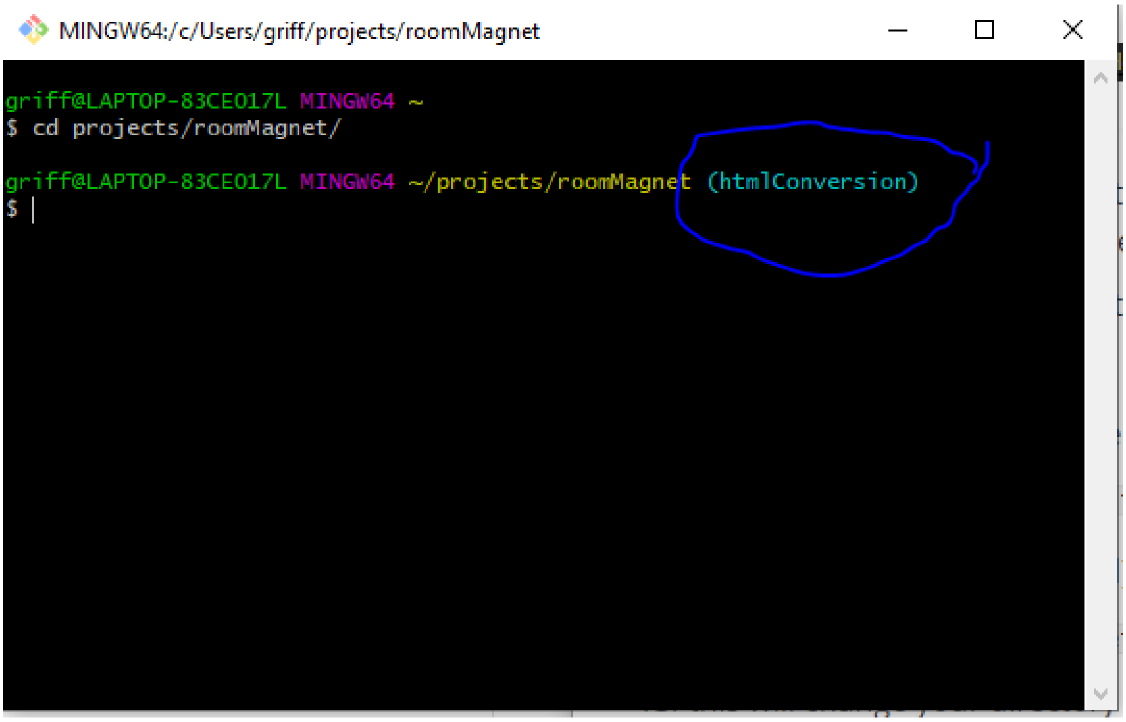
griffincosgrove Update issue templates Latest commit 9c4a6a0 6 minutes ago

.github/ISSUE_TEMPLATE	Update issue templates	6 minutes ago
Bin/roslyn	Deleted all html	4 days ago
RoomMagnetTake1	added gender to tenent and host class, and updated the insert stateme...	4 hours ago
SQLFiles	update phone number field to be more robust	3 days ago
packages/Microsoft.CodeDom.Provi...	fix files n stuff	4 days ago
.DS_Store	moving files n stuff	5 days ago
LICENSE	Initial commit	27 days ago
RoomMagnetTake1.sln	moving files n stuff	5 days ago

Help people interested in this repository understand your project by adding a README.

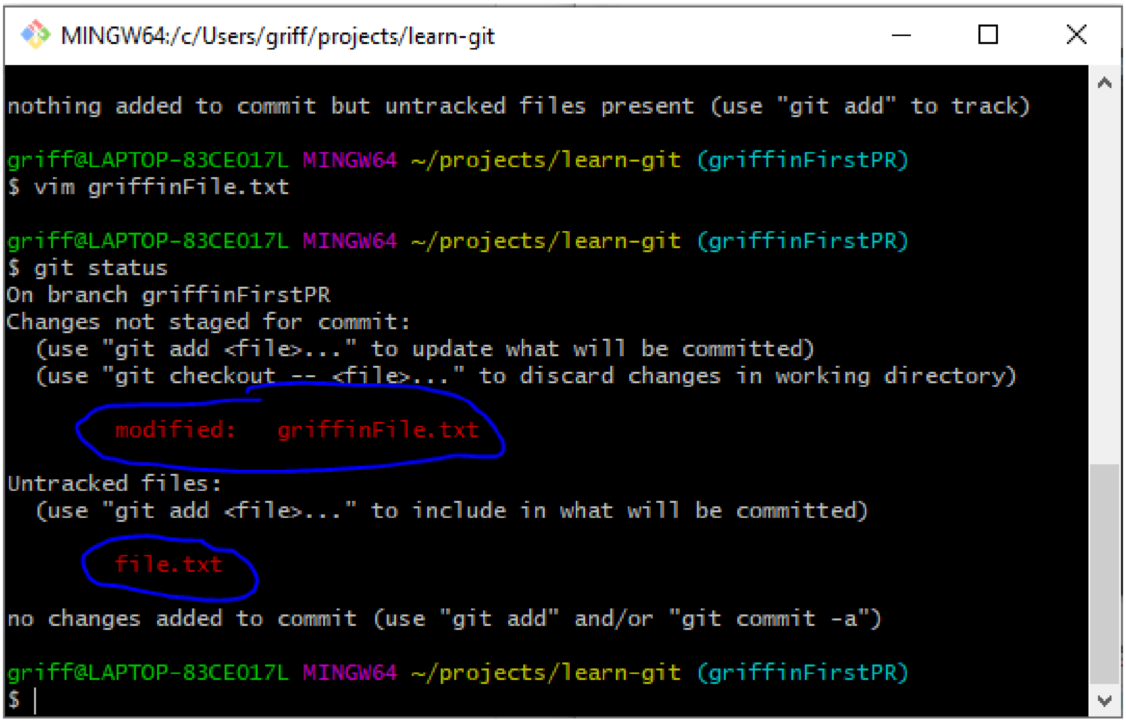
6. click clone, then copy and paste the link it should be: git@github.com:griffincosgrove/roomMagnet.git
7. Once the download is complete launch git bash
8. type `mkdir capstone`
9. that will make a directory called capstone
10. in git bash type `git clone git@github.com:griffincosgrove/roomMagnet.git`
11. this will clone (think download) the repository (think folder) locally.
12. in git bash type `cd roomMagnet`
13. this will change your directory to the repo (short for repository) that you just cloned
14. in git bash type `ls` it should display all the files the same as the repo on github.
15. the next step is to create a branch, it is very bad to develop on the master branch, **we must protect the master branch** from potentially faulty code.
16. in git bash type `git checkout -b  yournameDev` so mine would be griffinDev
17. now it should display the name of your branch instead of (master) like this:

18.



```
griff@LAPTOP-83CE017L MINGW64 ~
$ cd projects/roomMagnet/

griff@LAPTOP-83CE017L MINGW64 ~/projects/roomMagnet (htmlConversion)
$ |
```

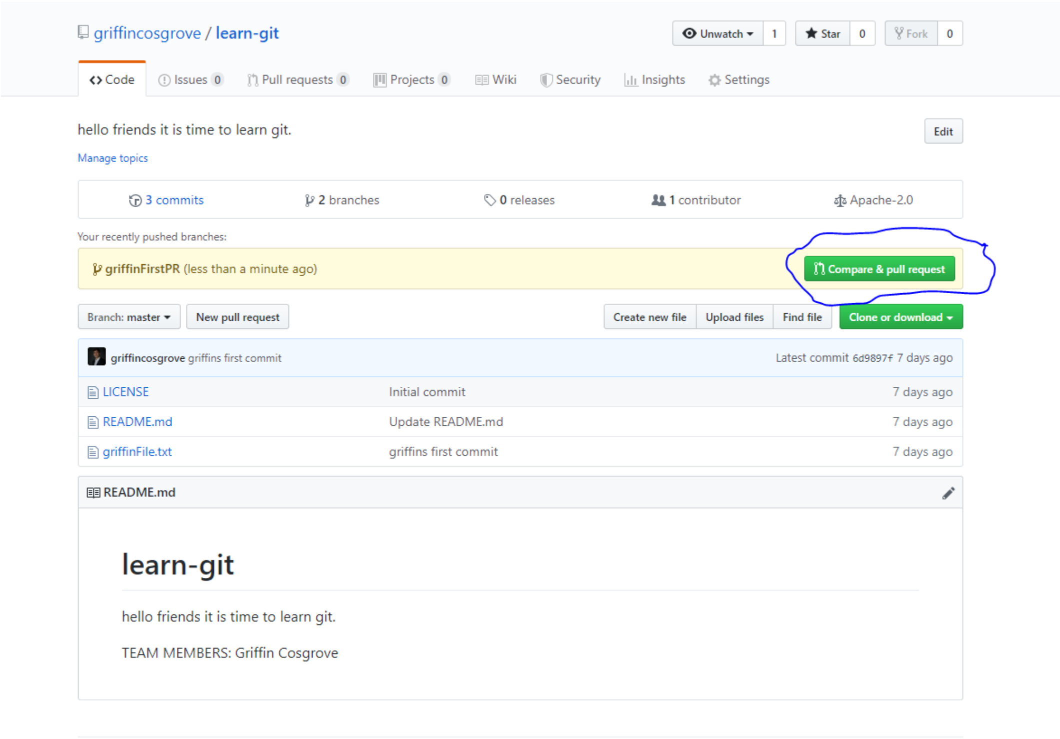
19. my branch is called htmlconversion if it says (master) it means u need to create/switch to a different branch. **this is very important** to double check you are not on the master.
20. now you can edit the code how you like, **BUT YOU MUST EDIT CODE IN THIS FOLDER/REPO**
21. so now youve changed something and have some changes you want to send to the central repo on github.
22. type `git status`
23. if it says you have untracked files looks like this:



```
nothing added to commit but untracked files present (use "git add" to track)

griff@LAPTOP-83CE017L MINGW64 ~/projects/learn-git (griffinFirstPR)
$ vim griffinFile.txt

griff@LAPTOP-83CE017L MINGW64 ~/projects/learn-git (griffinFirstPR)
$ git status
On branch griffinFirstPR
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   griffinFile.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.txt

no changes added to commit (use "git add" and/or "git commit -a")

griff@LAPTOP-83CE017L MINGW64 ~/projects/learn-git (griffinFirstPR)
$ |
```

24. there is two things to notice. one there are modified files, obviously :)
25. two there is an untracked file
26. modified files is ok, just know all modified files the changes will be reflected in the github central repo after you send.
27. you must add untracked files
28. in git bash type `git add --all` or `git add <filename>`

29. now type `git status` and there should be 0 untracked files, unless you want them to be untracked
30. now you are ready to commit
31. in git bash type `git commit -am "<some message about the commit describing the changes you made, but briefly"`
32. now you are ready to push, double check you are **NOT** on the master branch again.
33. pushing is the act of sending the central repo on github your changes.
34. in git bash type `git push --set-upstream origin <yourbranchname>`
35. this will push the changes you made to the central repo for review.
36. open up the repo on github
37.



griffincosgrove / learn-git

hello friends it is time to learn git.

3 commits | 2 branches | 0 releases | 1 contributor | Apache-2.0

Your recently pushed branches:

griffinFirstPR (less than a minute ago)

griffincosgrove griffins first commit		Latest commit 6d9897f 7 days ago
LICENSE	Initial commit	7 days ago
README.md	Update README.md	7 days ago
griffinFile.txt	griffins first commit	7 days ago

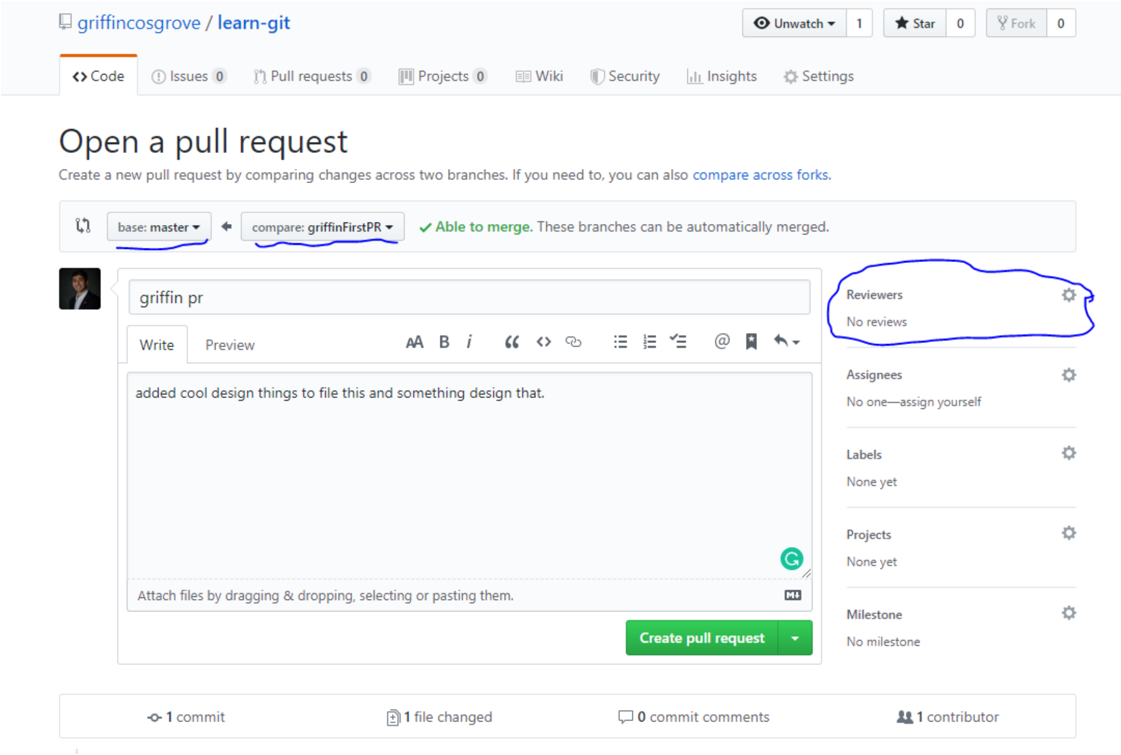
README.md

learn-git

hello friends it is time to learn git.

TEAM MEMBERS: Griffin Cosgrove

38. click that button
39. screen look like this



griffincosgrove / learn-git

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: griffinFirstPR ✓ Able to merge. These branches can be automatically merged.

griffin pr

added cool design things to file this and something design that.

Attach files by dragging & dropping, selecting or pasting them.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

1 commit | 1 file changed | 0 commit comments | 1 contributor

40. **then click reviewers and add griffincosgrove and ayman** to the reviews. then click create pull request.
41. **DO NOT CLICK MERGE PULL REQUEST**
42. then ayman or I will approve your PR if it is safe, or deny the request go from there
43. this is just to make sure the changes you make will not implode the program on a functionality side.
44. you are done and cooler because you can *git*
45. thank you for trying to learn git. if we take care of this we will win.

    -git master griff