PPOL 768 – Spring 2023 - Week 2

Béatrice Leydier

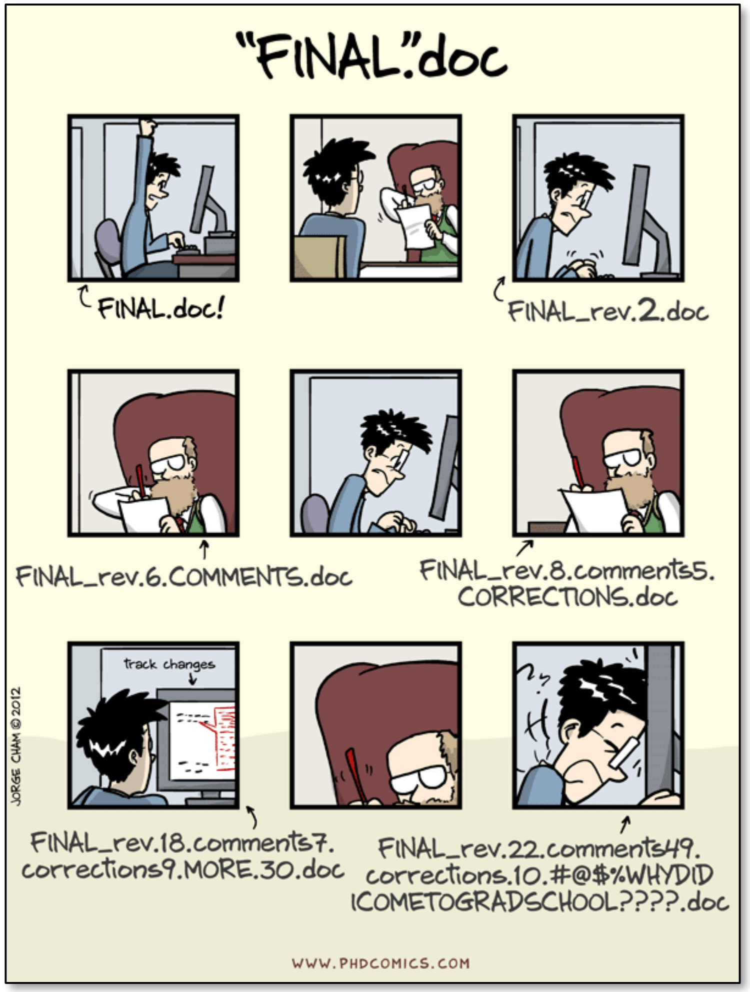
Version Control : Git & Github Workshop

Georgetown University Initiative

guide

on Innovation, Development and Evaluation

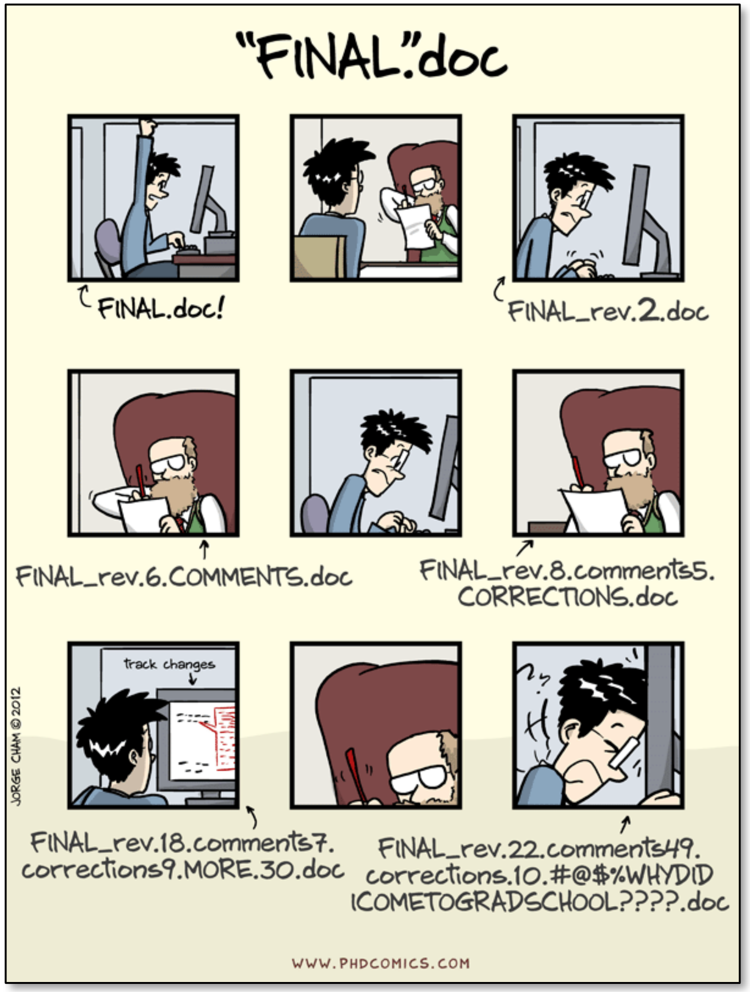
What is Git?


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git?

Git is a **version control protocol.** It keeps track of *versions* of every file in a project in an orderly way.

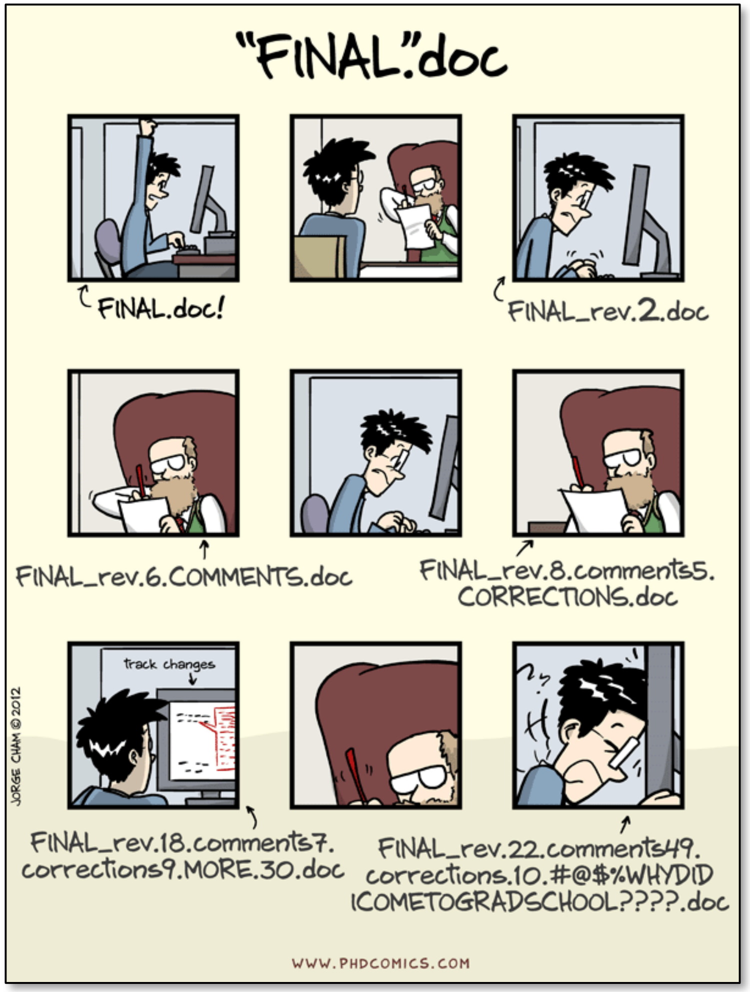
Note: if you want to be fancy, you can say that git is a self-certifying (decentralized) protocol, and some people say it is the kind of stuff that [web3](web3) is made of.


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git?
What is GitHub?

Git is a **version control protocol.** It keeps track of *versions* of every file in a project in an orderly way.

Note: if you want to be fancy, you can say that git is a self-certifying (decentralized) protocol, and some people say it is the kind of stuff that [web3](web3) is made of.

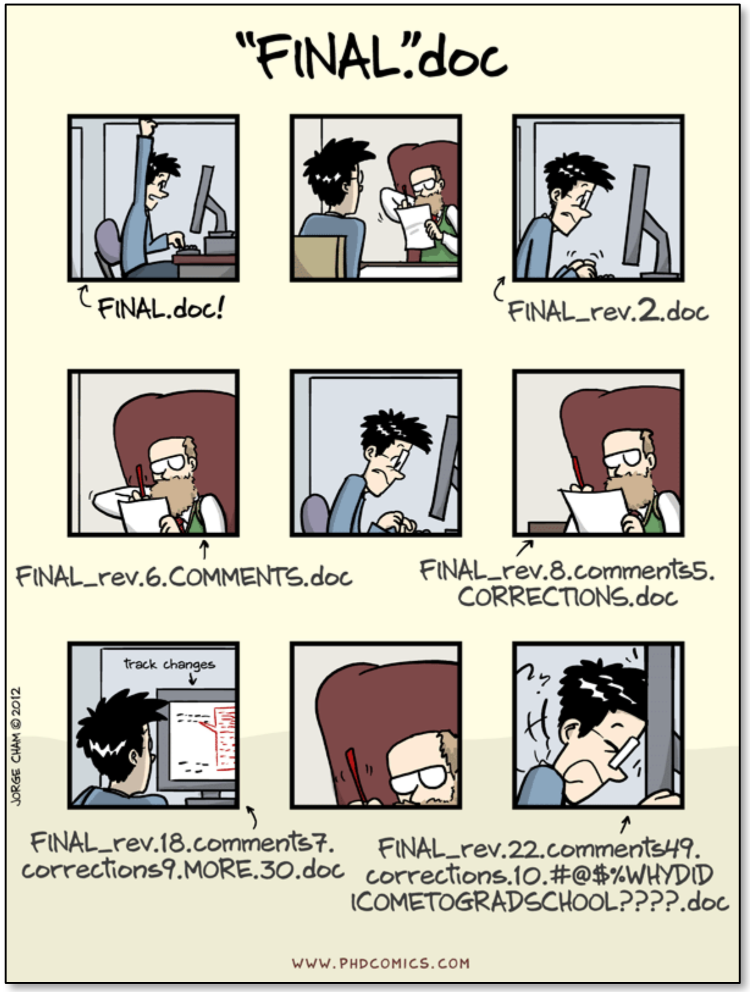

"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git?
What is GitHub?

Git is a **version control protocol.** It keeps track of *versions* of every file in a project in an orderly way.

Note: if you want to be fancy, you can say that git is a self-certifying (decentralized) protocol, and some people say it is the kind of stuff that [web3]() is made of.

GitHub is a **website and server** where people can host the projects they use Git on, in order to track the different *contributions* to the projects in an orderly way.

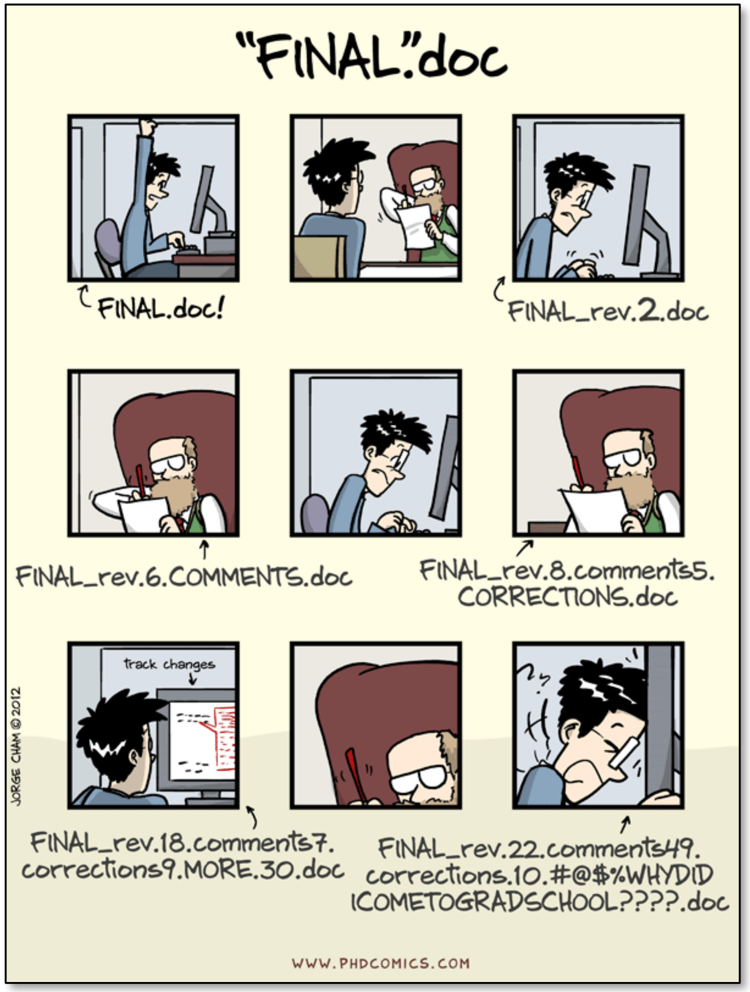

"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git?
What is GitHub?
What is GitKraken?

Git is a **version control protocol.** It keeps track of *versions* of every file in a project in an orderly way.

Note: if you want to be fancy, you can say that git is a self-certifying (decentralized) protocol, and some people say it is the kind of stuff that [web3](web3) is made of.

GitHub is a **website and server** where people can host the projects they use Git on, in order to track the different *contributions* to the projects in an orderly way.


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git?
What is GitHub?
What is GitKraken?

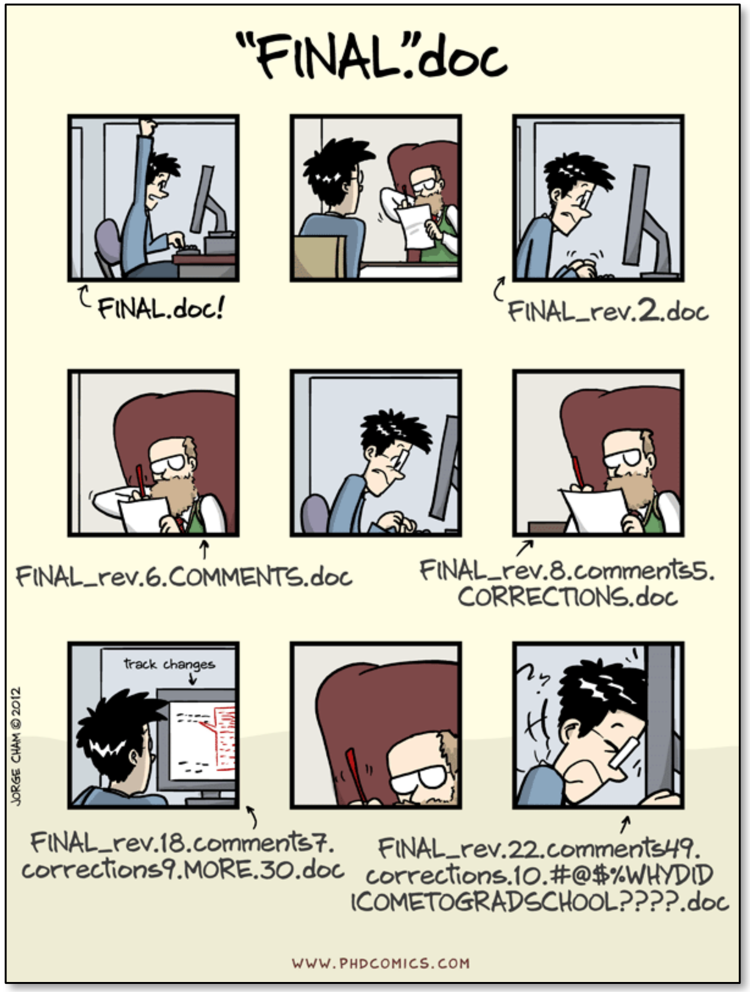
Git is a **version control protocol.** It keeps track of *versions* of every file in a project in an orderly way.

Note: if you want to be fancy, you can say that git is a self-certifying (decentralized) protocol, and some people say it is the kind of stuff that web3 is made of.

GitHub is a **website and server** where people can host the projects they use Git on, in order to track the different *contributions* to the projects in an orderly way.

GitKraken is a **client** (software on your computer) that allows you to use Git and Github in a user-friendly way.

Together, Git, GitHub and GitKraken allow you to avoid the "FINAL".doc problem.



So how does it *work*?

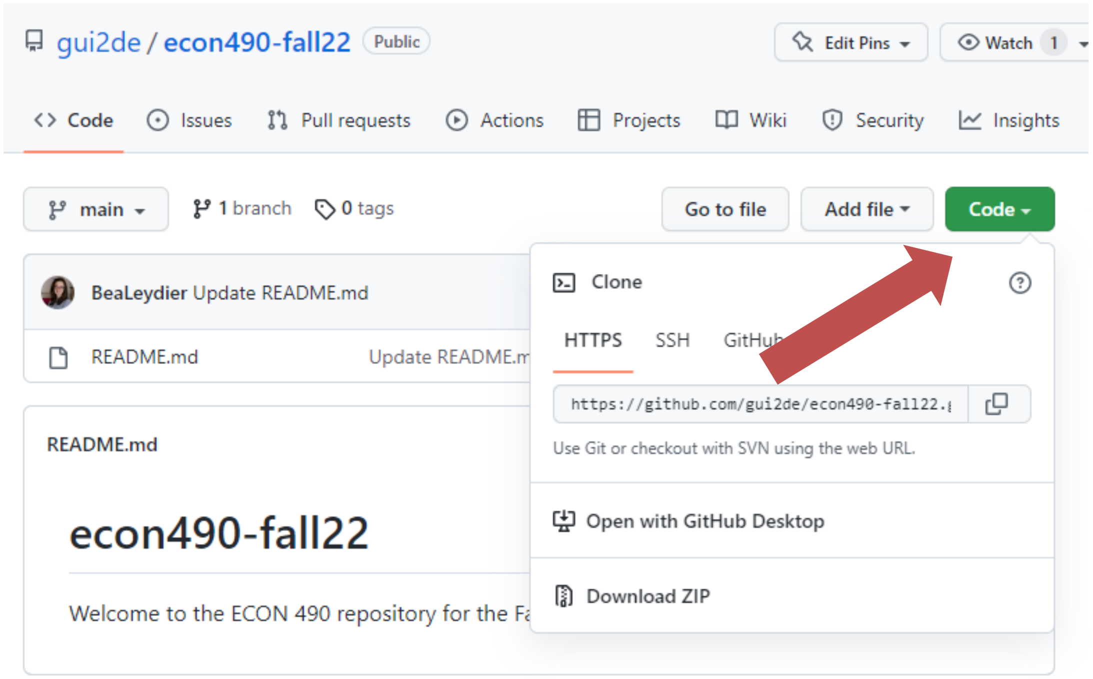
Part 1: Cloning a repository

GEORGETOWN UNIVERSITY

Clone the class repo with your desktop client

Cloning the repository makes a complete copy of it at the new location, including the entire history (remember, the repository is the history).

Cloning from GitHub to your local computer is a good way to start a repository.



BeaLeydier Update README.md

README.md Update README.m

https://github.com/gui2de/econ490-fall22.g

Use Git or checkout with SVN using the web URL.

README.md

econ490-fall22

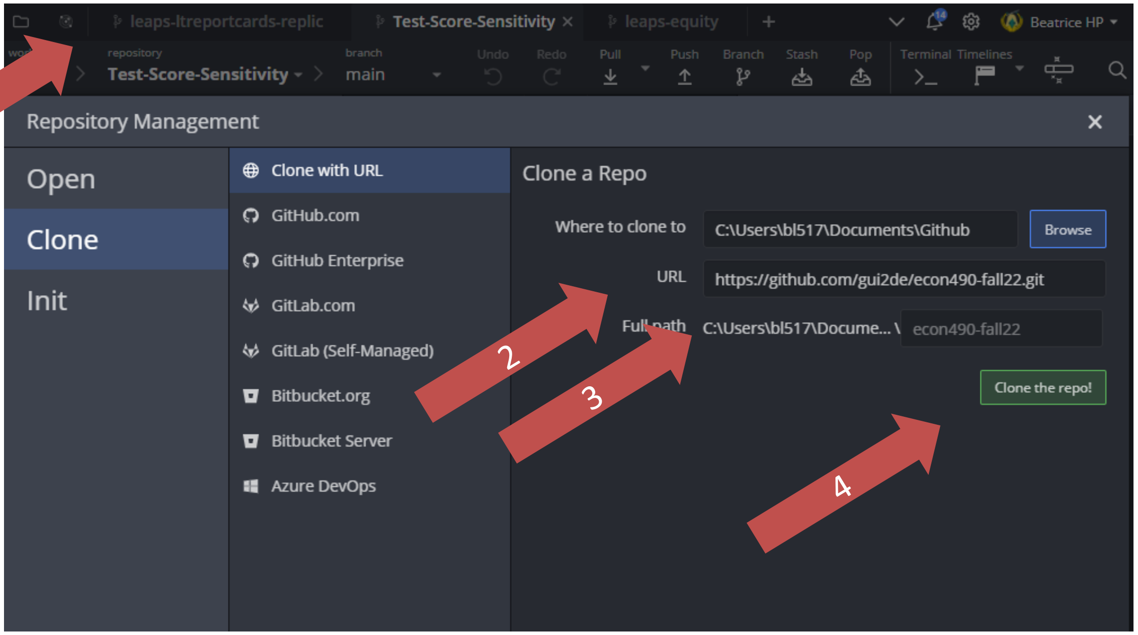
Welcome to the ECON 490 repository for the Fa

Clone the repo with your desktop client

Cloning the repository makes a complete copy of it at the new location, including the entire history (remember, the repository is the history).

Cloning from GitHub to your local computer is a good way to start a repository.

(I know it seems like a lot of steps now, but this becomes second nature. Bear with me, and use this presentation as a guide!)

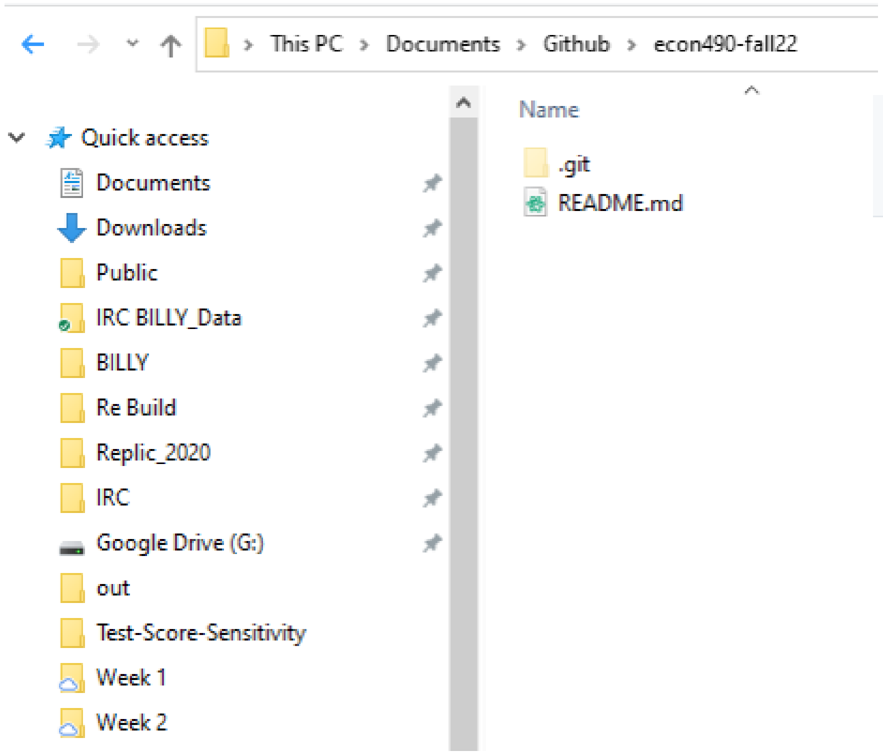


Where to clone to: C:\Users\bl517\Documents\Github

URL: https://github.com/gui2de/econ490-fall22.git

Full path: C:\Users\bl517\Docume...\ econ490-fall22

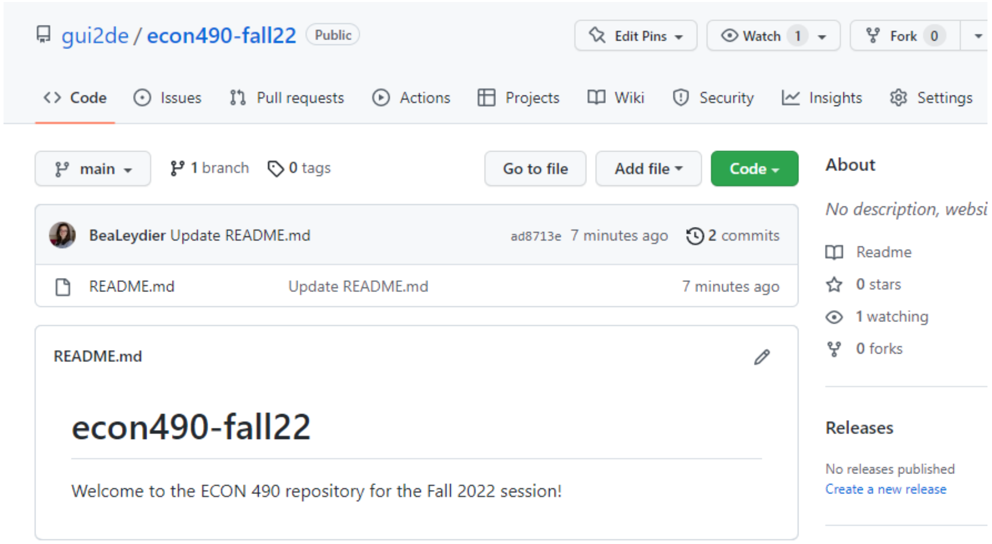
The "local" and "remote" instances are identical



This PC › Documents › Github › econ490-fall22

Quick access
- Documents
- Downloads
- Public
- IRC BILLY_Data
- BILLY
- Re Build
- Replic_2020
- IRC
- Google Drive (G:)
- out
- Test-Score-Sensitivity
- Week 1
- Week 2

Name
- .git
- README.md



gui2de / econ490-fall22 Public

Edit Pins · Watch 1 · Fork 0

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

main · 1 branch · 0 tags · Go to file · Add file · Code

BeaLeydier Update README.md ad8713e 7 minutes ago 2 commits

README.md Update README.md 7 minutes ago

README.md

econ490-fall22

Welcome to the ECON 490 repository for the Fall 2022 session!

About

No description, websi

Readme · 0 stars · 1 watching · 0 forks

Releases

No releases published

Create a new release

So how does it *work*?

Part 2: Creating branches in your local repository and making changes

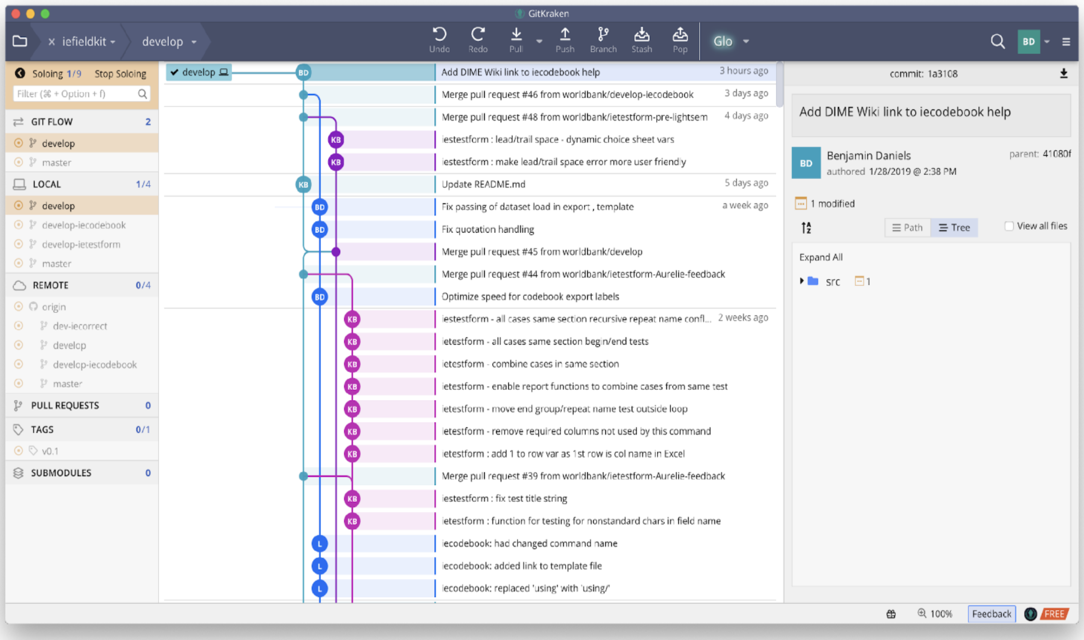
GEORGETOWN UNIVERSITY

Git creates branches

Branches are the “killer feature” of Git. Nearly every advanced Git usage you learn will be about how to manage branches.

Branches enable different people to work on the same thing at the same time, and they enable you to view different versions of your files.

Branching allows you to move forwards or backwards in time; and to move “horizontally” in time through various concurrent versions.



Add DIME Wiki link to iecodebook help	3 hours ago
Merge pull request #46 from worldbank/develop-iecodebook	3 days ago
Merge pull request #48 from worldbank/ietestform-pre-lightsem	4 days ago
iestestform : lead/trail space - dynamic choice sheet vars	
iestestform : make lead/trail space error more user friendly	
Update README.md	5 days ago
Fix passing of dataset load in export , template	a week ago
Fix quotation handling	
Merge pull request #45 from worldbank/develop	
Merge pull request #44 from worldbank/ietestform-Aurelie-feedback	
Optimize speed for codebook export labels	
iestestform - all cases same section recursive repeat name confl...	2 weeks ago
ietestform - all cases same section begin/end tests	
ietestform - combine cases in same section	
ietestform - enable report functions to combine cases from same test	
ietestform - move end group/repeat name test outside loop	
ietestform - remove required columns not used by this command	
ietestform : add 1 to row var as 1st row is col name in Excel	
Merge pull request #39 from worldbank/ietestform-Aurelie-feedback	
iestestform : fix test title string	
ietestform : function for testing for nonstandard chars in field name	
iecodebook: had changed command name	
iecodebook: added link to template file	
iecodebook: replaced 'using' with 'using/'	

commit: 1a3108

Add DIME Wiki link to iecodebook help

Benjamin Daniels
authored 1/28/2019 @ 2:38 PM
parent: 41080f

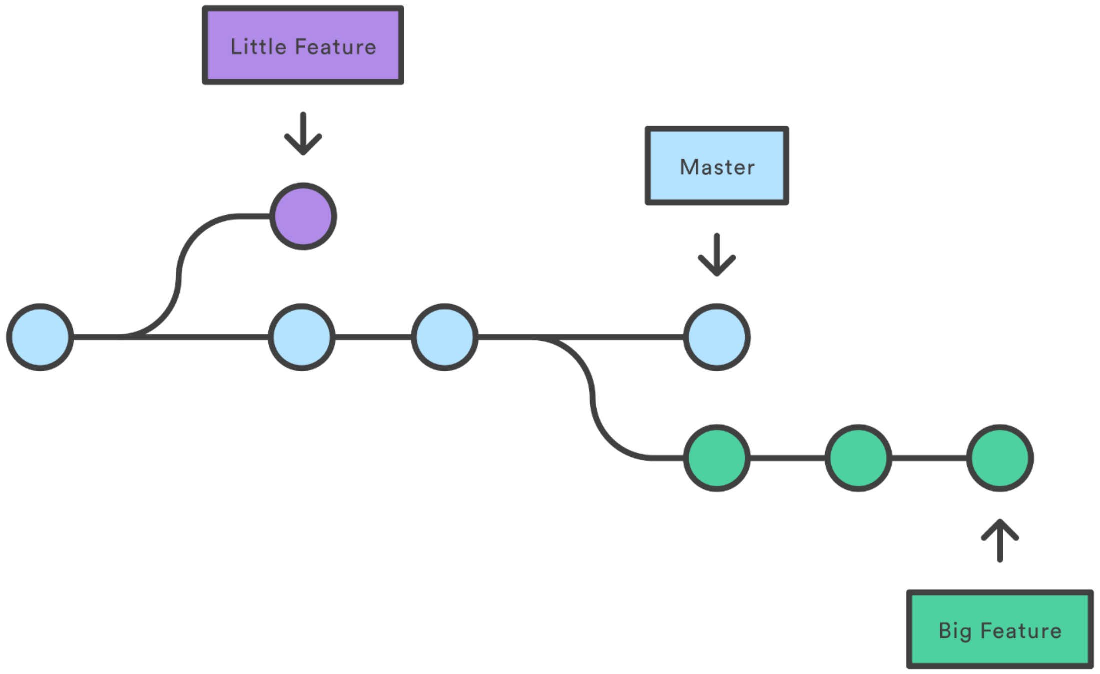
1 modified

Branches name current versions

Each "branch" points to a commit, usually the latest version in some development workflow. You can switch between branches *locally*. When you do, your working directory will look exactly as it did when that commit was made.

Every contributor can be on any branch they like at any time – past or current.

This means various experimental changes can be made simultaneously without affecting the current edition of a product.

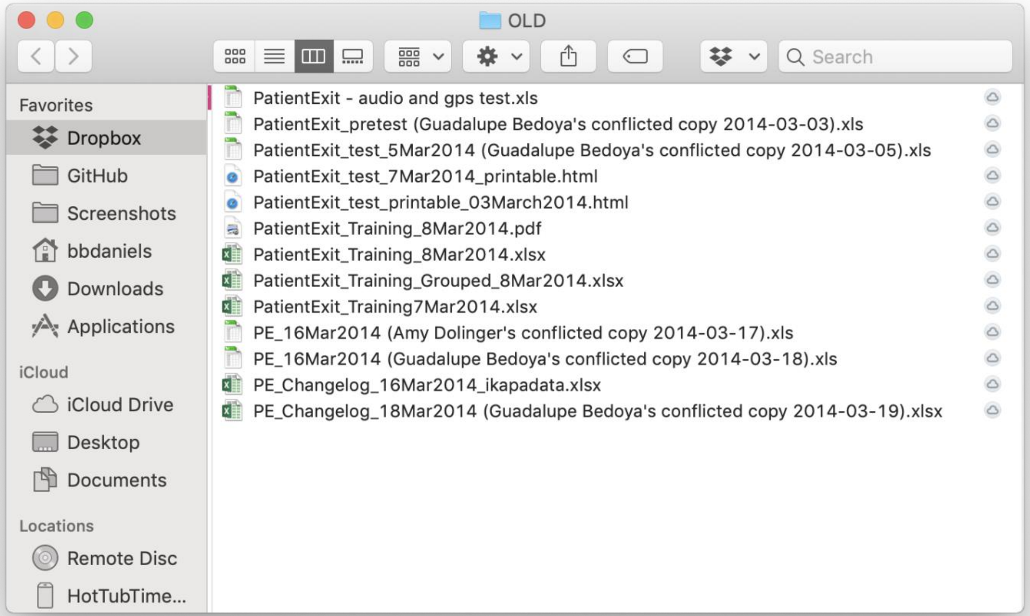


How is that better than Dropbox?

In Dropbox, there can be only *one* living version of a file at any time – otherwise a “conflicted copy” is created. This means nobody can edit the same file at the same time: Dropbox has no concurrency.

Worse, if you “roll back” a file to a previous version to see what it looked like, this affects everyone’s *current* version, even if you don’t want it to.

Finally, Dropbox versions are costly (computationally) – so they delete them often without telling you.



PatientExit - audio and gps test.xls
PatientExit_pretest (Guadalupe Bedoya's conflicted copy 2014-03-03).xls
PatientExit_test_5Mar2014 (Guadalupe Bedoya's conflicted copy 2014-03-05).xls
PatientExit_test_7Mar2014_printable.html
PatientExit_test_printable_03March2014.html
PatientExit_Training_8Mar2014.pdf
PatientExit_Training_8Mar2014.xlsx
PatientExit_Training_Grouped_8Mar2014.xlsx
PatientExit_Training7Mar2014.xlsx
PE_16Mar2014 (Amy Dolinger's conflicted copy 2014-03-17).xls
PE_16Mar2014 (Guadalupe Bedoya's conflicted copy 2014-03-18).xls
PE_Changelog_16Mar2014_ikapadata.xlsx
PE_Changelog_18Mar2014 (Guadalupe Bedoya's conflicted copy 2014-03-19).xlsx

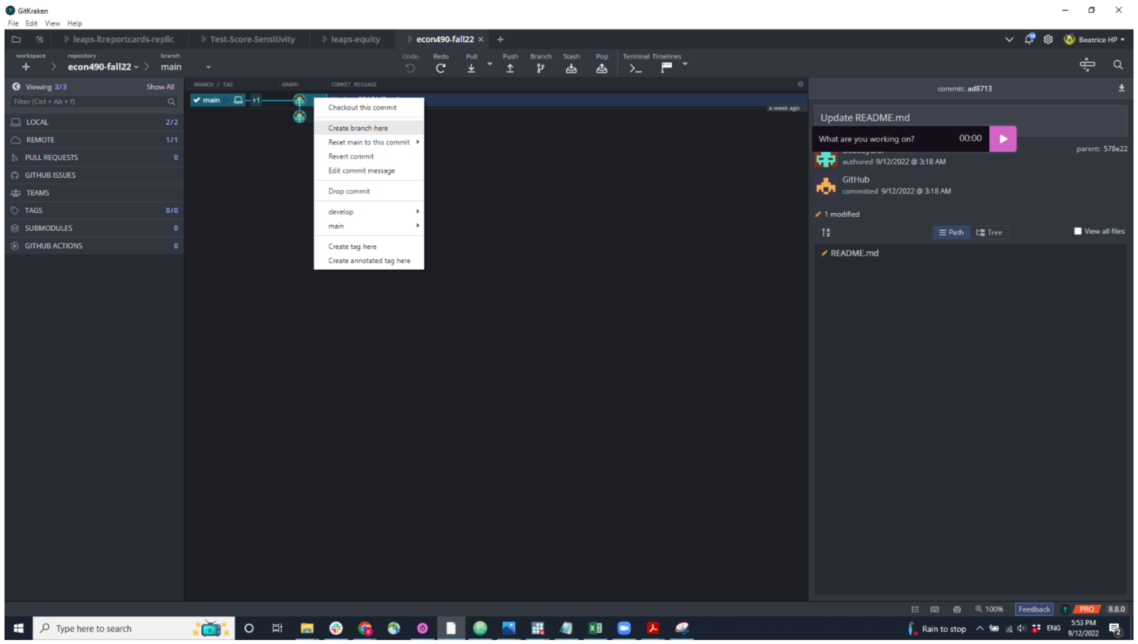
Create a branch with your initials and week number in GitKraken

Using your initials is common to identify specific branches you are working on.

In one popular workflow model, *main* always holds a released product, *develop* is the workstream for the next version, and specific features and other versions are in specific branches.

For the class repo, we only have a *main* branch and your weekly feature branches.

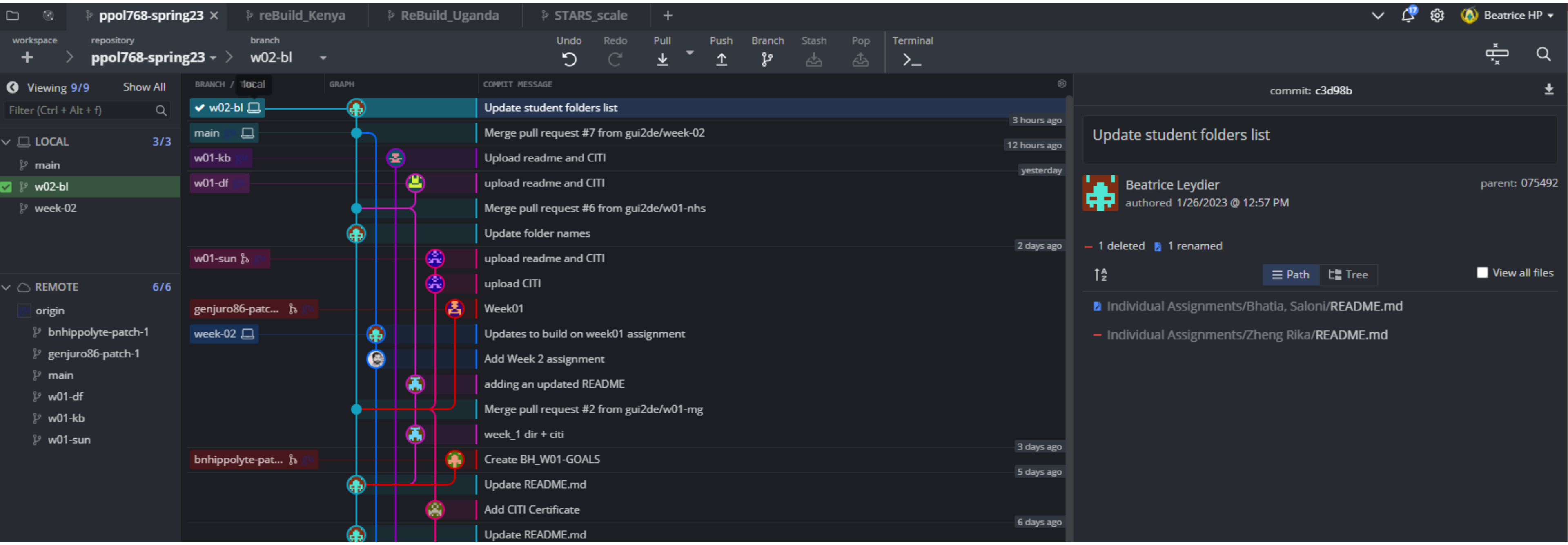
Right-click on the commit named “Initial commit”, and select “Create branch here”.



Make changes to the repo on your own branch

Create your weekly subfolder, add your bio and certificate files

Stage and commit the changes



w02-bl	Update student folders list
main	Merge pull request #7 from gui2de/week-02
w01-kb	Upload readme and CITI
w01-df	upload readme and CITI
	Merge pull request #6 from gui2de/w01-nhs
	Update folder names
w01-sun	upload readme and CITI
	upload CITI
genjuro86-patc...	Week01
week-02	Updates to build on week01 assignment
	Add Week 2 assignment
	adding an updated README
	Merge pull request #2 from gui2de/w01-mg
	week_1 dir + citi
bnhippolyte-pat...	Create BH_W01-GOALS
	Update README.md
	Add CITI Certificate
	Update README.md

3 hours ago
12 hours ago
yesterday
2 days ago
3 days ago
5 days ago
6 days ago

commit: c3d98b

Update student folders list

Beatrice Leydier
authored 1/26/2023 @ 12:57 PM
parent: 075492

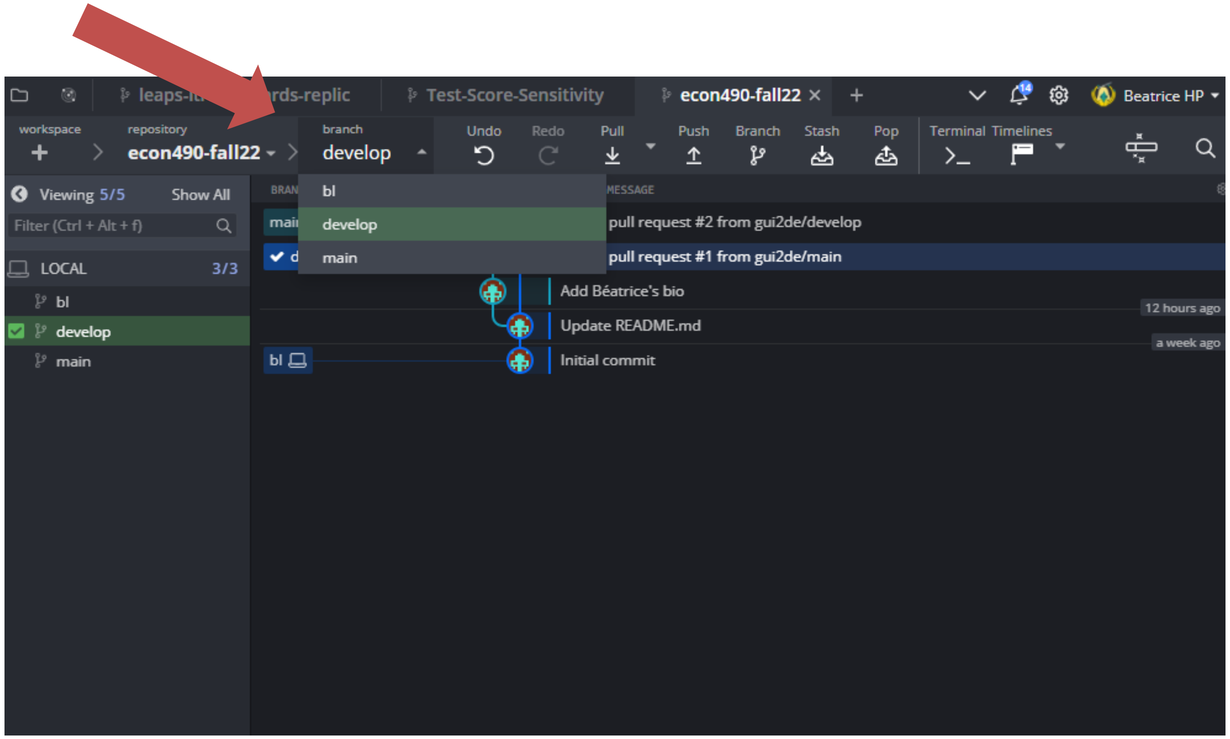
1 deleted 1 renamed

Path Tree View all files

Individual Assignments/Bhatia, Saloni/README.md
Individual Assignments/Zheng Rika/README.md

Check out the *main* branch

Go back and forth between branches and see how the files disappear from your local repo.



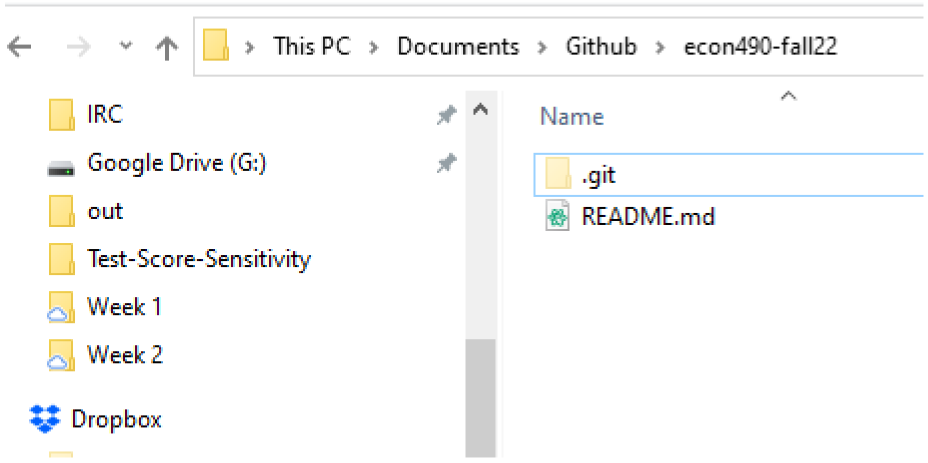
pull request #2 from gui2de/develop
pull request #1 from gui2de/main
Add Béatrice's bio
Update README.md
Initial commit

Where's my file?

If you look in the working directory associated with the class, you will see that it is exactly in the state it was when you initialized it.

Don't ask where your file "is". You really don't need to know.

What you do need to know is that by "checking out" your *feature* branch, it will be in the working directory again. Try it!



So how does it *work*?

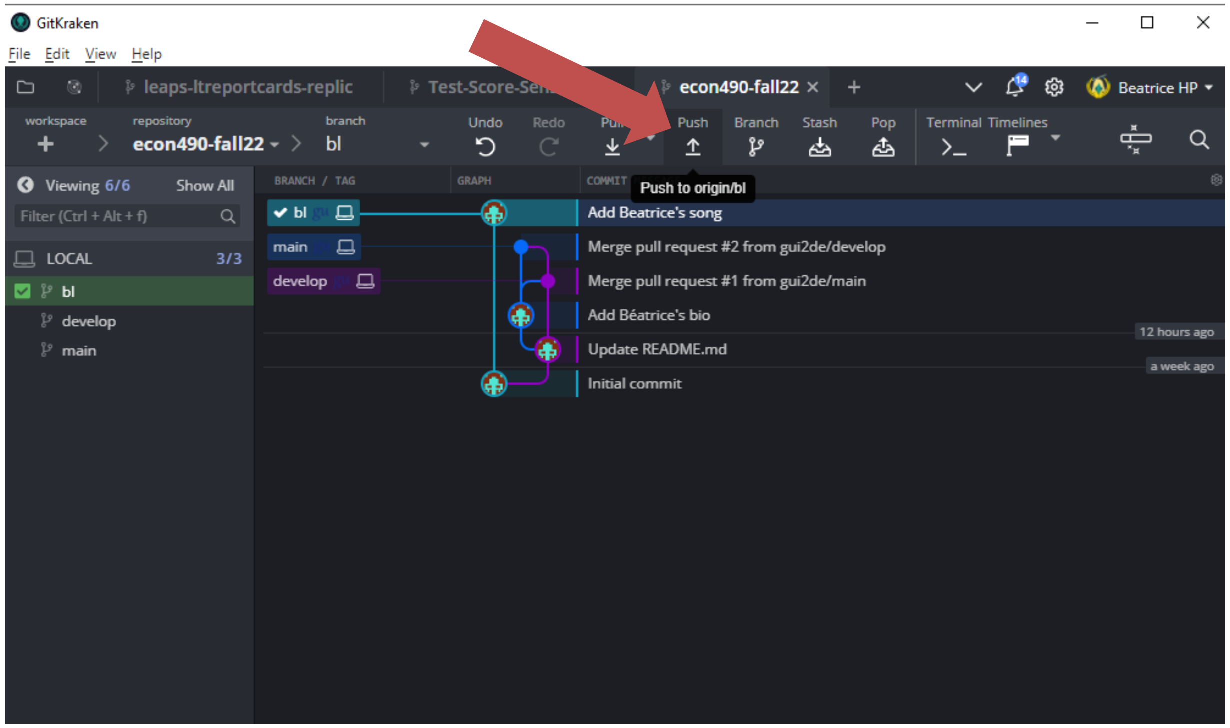
Part 3: Syncing your local repo to the remote repo (Push/Pull)

GEORGETOWN UNIVERSITY

Push your own branch branch to GitHub

Navigate back to the client and make sure your own branch is checked out.

“Push” this branch to GitHub (the “origin”). You will get a notification that the branch does not yet exist there, and ask you to name it. The default is the same as your local branch name; stick with that.



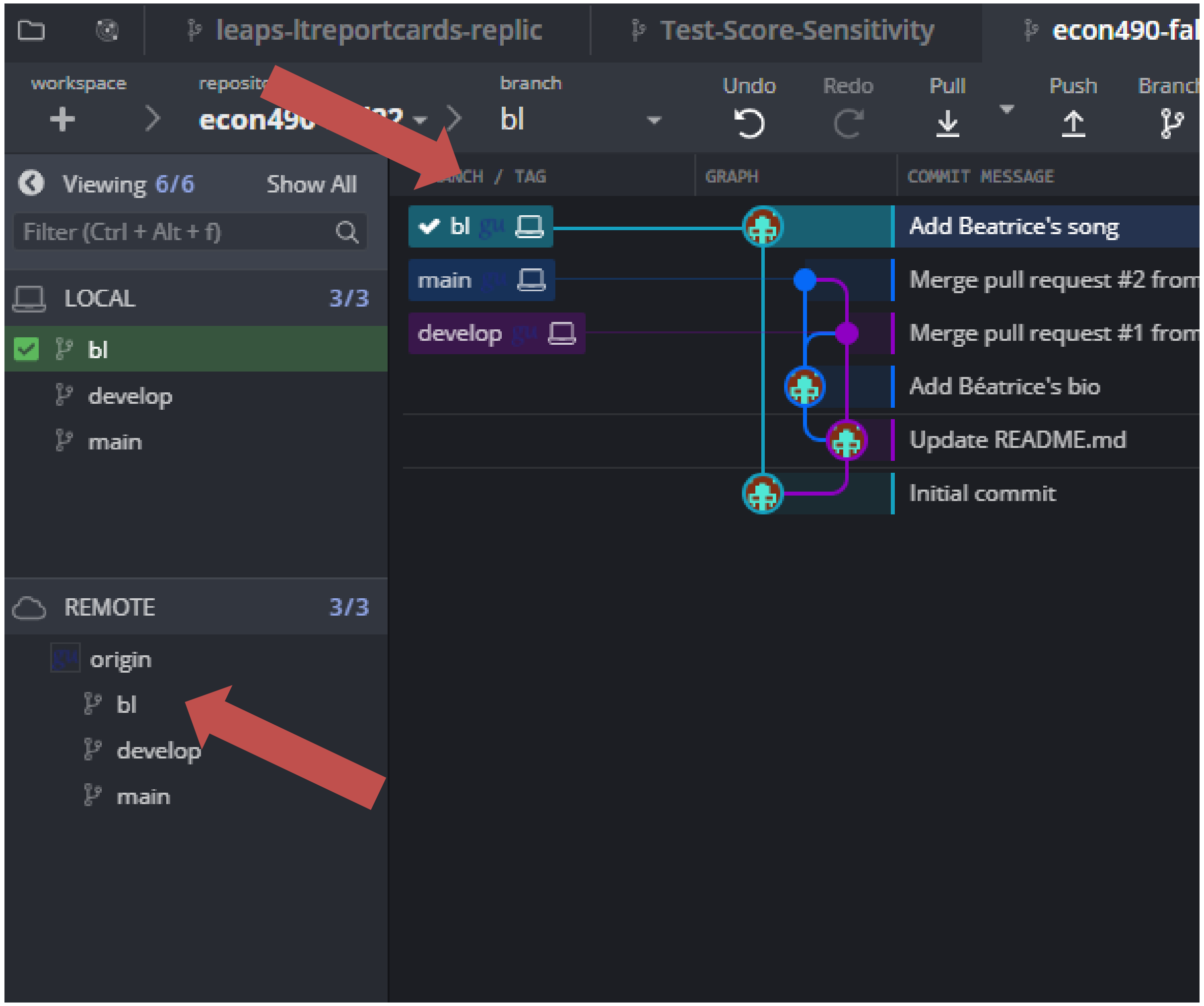
Add Beatrice's song
Merge pull request #2 from gui2de/develop
Merge pull request #1 from gui2de/main
Add Béatrice's bio
Update README.md
Initial commit
12 hours ago
a week ago

Push your own branch to GitHub

Navigate back to the client and make sure your own branch is checked out.

"Push" this branch to GitHub (the "origin"). You will get a notification that the branch does not yet exist there, and ask you to name it. The default is the same as your local branch name; stick with that.

You'll know it's worked when the client reflects the existence and location of your own branch on *origin*.



Add Beatrice's song
Merge pull request #2 from
Merge pull request #1 from
Add Béatrice's bio
Update README.md
Initial commit

So how does it *work*?

Part 4: Merging (Pull Request) branches using GitHub

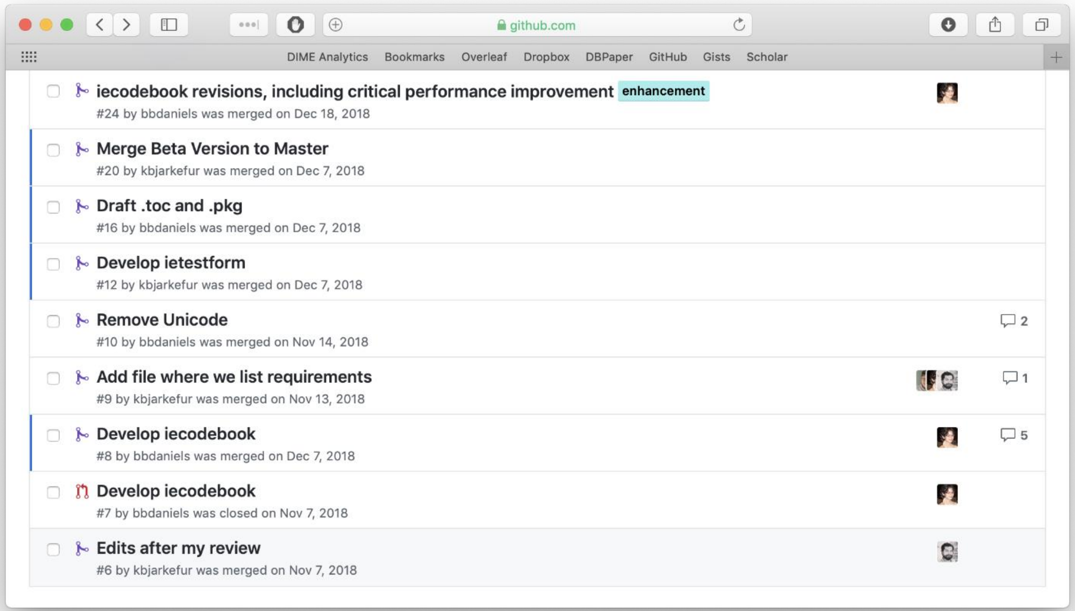
GEORGETOWN UNIVERSITY

GitHub manages contributions

GitHub provides interfaces for assigning tasks, submitting updates, and approving and accepting contributions into a project.

Like Git, everything you ever do is saved in an orderly way, so you can always look back and see when and why you did (or undid) something.

It is designed to make project record-keeping and task-management as part of your workflow.


github.com
DIME Analytics Bookmarks Overleaf Dropbox DBPaper GitHub Gists Scholar

iecodebook revisions, including critical performance improvement enhancement
#24 by bbdaniels was merged on Dec 18, 2018

Merge Beta Version to Master
#20 by kbjarkefur was merged on Dec 7, 2018

Draft .toc and .pkg
#16 by bbdaniels was merged on Dec 7, 2018

Develop ietestform
#12 by kbjarkefur was merged on Dec 7, 2018

Remove Unicode 2
#10 by bbdaniels was merged on Nov 14, 2018

Add file where we list requirements 1
#9 by kbjarkefur was merged on Nov 13, 2018

Develop iecodebook 5
#8 by bbdaniels was merged on Dec 7, 2018

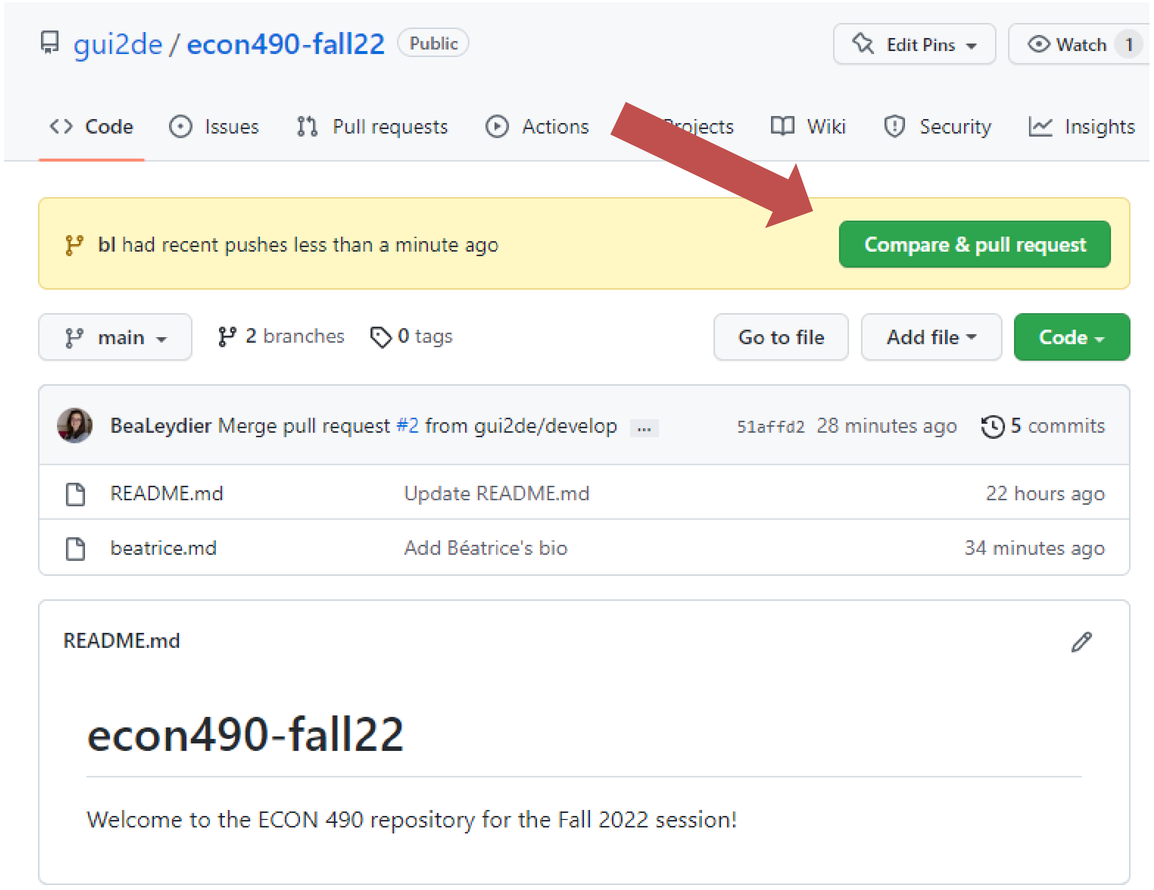
Develop iecodebook
#7 by bbdaniels was closed on Nov 7, 2018

Edits after my review
#6 by kbjarkefur was merged on Nov 7, 2018

Go to GitHub and open a pull request

When there are recent changes, GitHub will notify you and ask you if you want to merge the changes.

This is not the most common way to start a pull request, but it will do for now.



gui2de / econ490-fall22 Public

bl had recent pushes less than a minute ago

2 branches 0 tags

BeaLeydier Merge pull request #2 from gui2de/develop ... 51affd2 28 minutes ago 5 commits

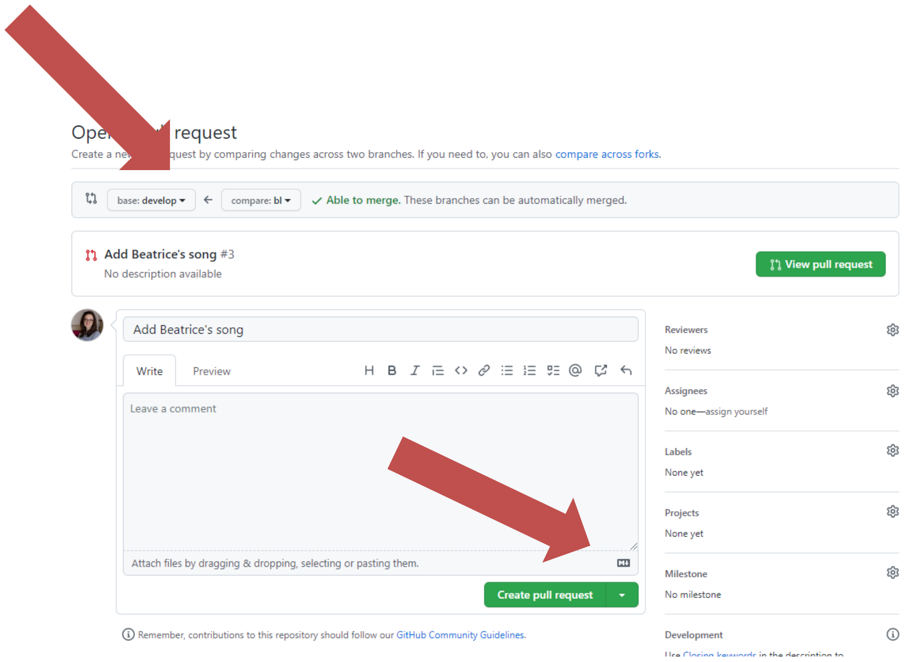
README.md	Update README.md	22 hours ago
beatrice.md	Add Béatrice's bio	34 minutes ago

README.md

econ490-fall22

Welcome to the ECON 490 repository for the Fall 2022 session!

Create and merge the pull request on the main branch



Ope[illegible] request

Create a ne[illegible]uest by comparing changes across two branches. If you need to, you can also compare across forks.

base: develop ← compare: bl ✓ Able to merge. These branches can be automatically merged.

Add Beatrice's song #3
No description available

View pull request

Add Beatrice's song

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

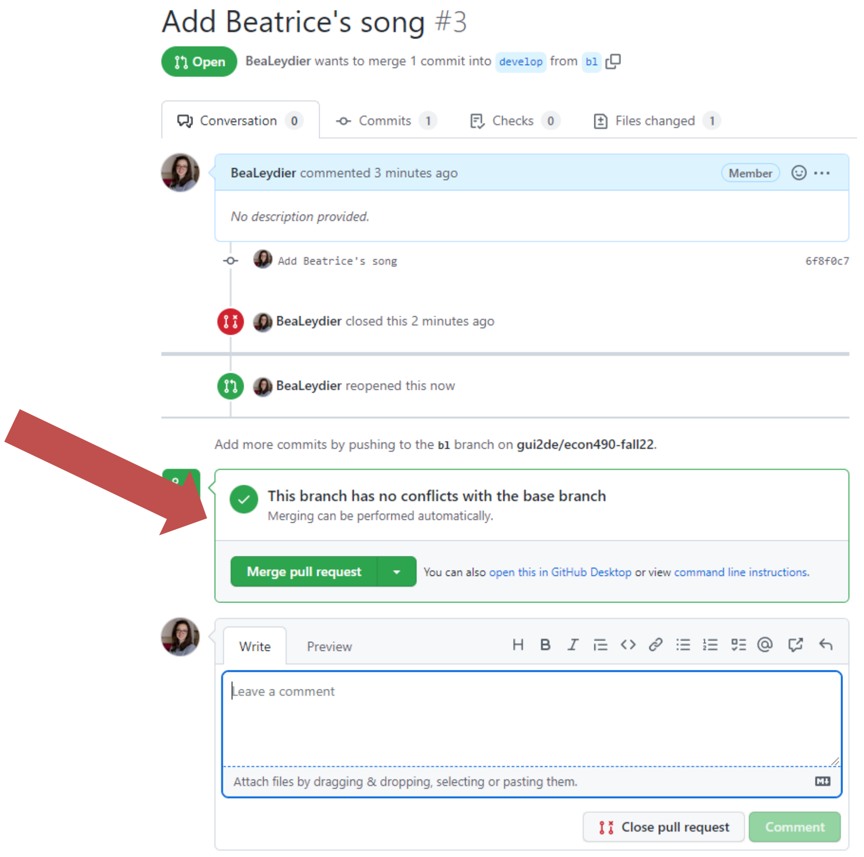
Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development



Add Beatrice's song #3

Open BeaLeydier wants to merge 1 commit into `develop` from `bl`

Conversation 0 Commits 1 Checks 0 Files changed 1

BeaLeydier commented 3 minutes ago Member

No description provided.

Add Beatrice's song 6f8f0c7

BeaLeydier closed this 2 minutes ago

BeaLeydier reopened this now

Add more commits by pushing to the **b1** branch on **gui2de/econ490-fall22**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

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

Write Preview

Leave a comment

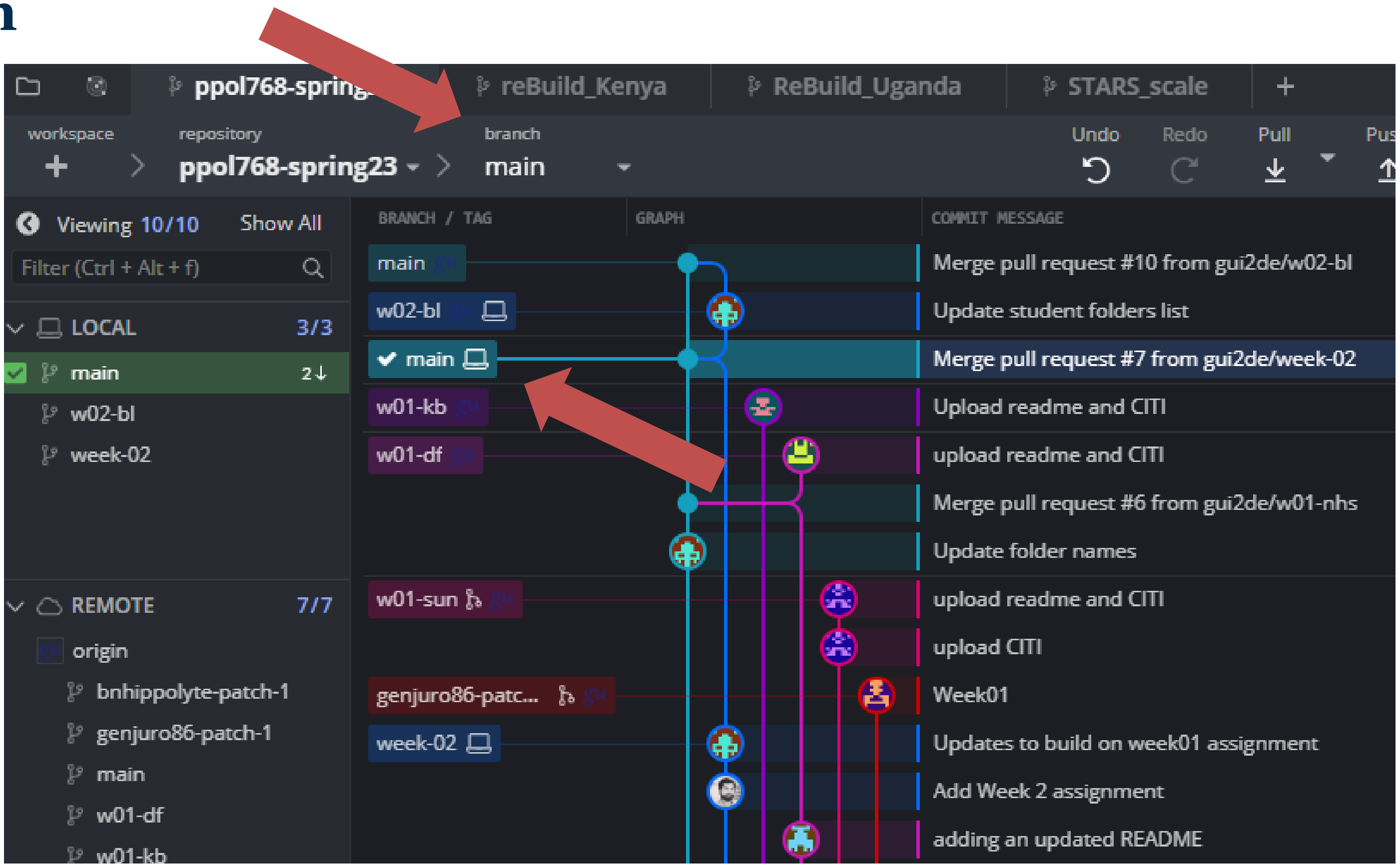
Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

Check out and pull the *main* branch

In your client, you will at first see that the "origin" copy of the *develop* branch is ahead of the local copy.

That's because you merged it on GitHub, and all operations in Git and GitHub are manual.



BRANCH / TAG	COMMIT MESSAGE
main	Merge pull request #10 from gui2de/w02-bl
w02-bl	Update student folders list
✓ main	Merge pull request #7 from gui2de/week-02
w01-kb	Upload readme and CITI
w01-df	upload readme and CITI
	Merge pull request #6 from gui2de/w01-nhs
	Update folder names
w01-sun	upload readme and CITI
	upload CITI
genjuro86-patc...	Week01
week-02	Updates to build on week01 assignment
	Add Week 2 assignment
	adding an updated README

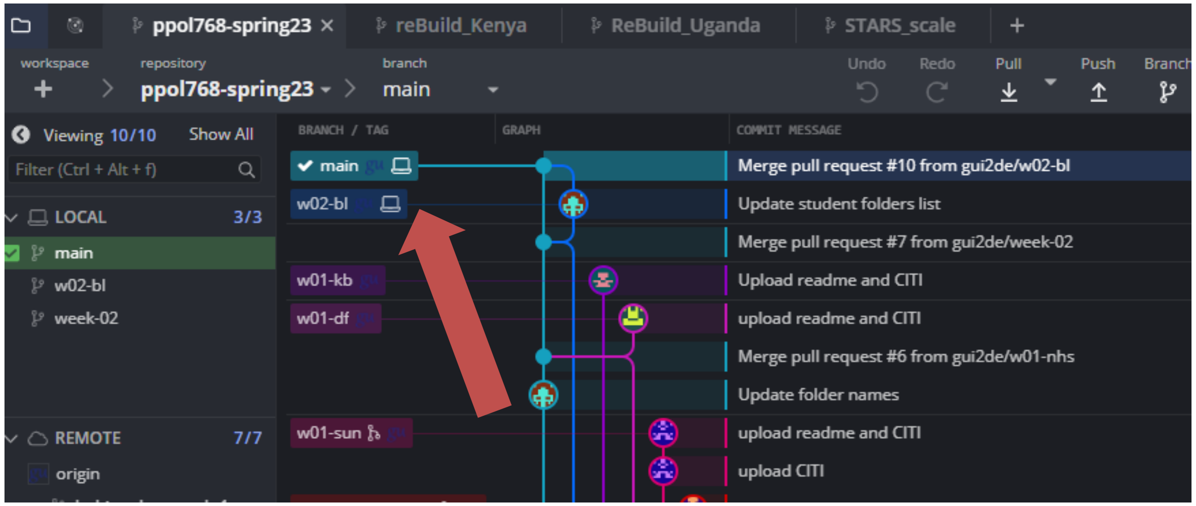
Check out and pull the *main* branch

In your client, you will at first see that the "origin" copy of the *develop* branch is ahead of the local copy.

That's because you merged it on GitHub, and all operations in Git and GitHub are manual.

When you "pull" the branch, you update your local copy to reflect the origin copy, and you will see the local pointer for the branch move forward.

What did this do?



Merge pull request #10 from gui2de/w02-bl
Update student folders list
Merge pull request #7 from gui2de/week-02
Upload readme and CITI
upload readme and CITI
Merge pull request #6 from gui2de/w01-nhs
Update folder names
upload readme and CITI
upload CITI

What is the difference between push/pull and pull request

- **Push/pull is a syncing operation between your local repository and the remote one**
  - There are two different repositories: the local one (the one in your working directory in your Documents) and the remote one (the one on Github)
  - Syncing between the two is what allows you to share your work with others
  - You can use the "version control" (track changes and version history) of Git locally on your computer, without ever pushing to a remote repository (if you are working on your own, for example)
  - The two repositories (local and remote/origin) do not communicate unless you tell them to with a push (from local to remote) or pull (from remote to local)

- **Pull request is a merge between two branches of the repository**
  - You typically do pull requests on your remote (github) repository because it is a process for obtaining permission before merging, but you could do merges between branches locally as well (without any collaboration)
  - Branches are ways to organize your commits in your repository in order to conveniently move between different versions of the same repository (for example, the version with/without a new feature)
  - Pull requests are ways to merge a branch into another one, to combine two versions of the repo (for example, accept the changes in your core repo after reviewing them)

I have deleted my branch after a pull request but I still see it on GitKraken

- **Branches** can exist independently in the local and/or remote version of the repository
- Branches serve the purpose of **pointing towards one or multiple commits to identify them**
  - We recommend **creating a branch for new features** (any new set of changes you make to the repo) on the repository to avoid merge conflicts when we work collaboratively
  - We recommend **deleting a feature branch after a successful pull request** that merges the branch into the develop branch, because the branch in that situation only serves the purpose of isolating your changes while you are still testing them
  - Github automatically suggests to delete a branch after it has been successfully merged into the develop branch
  - **Deleting a branch on the github repo doesn't automatically delete it in your local repository**
  - We recommend also manually deleting that feature branch in your local repo, to avoid confusion between your remote and local repo

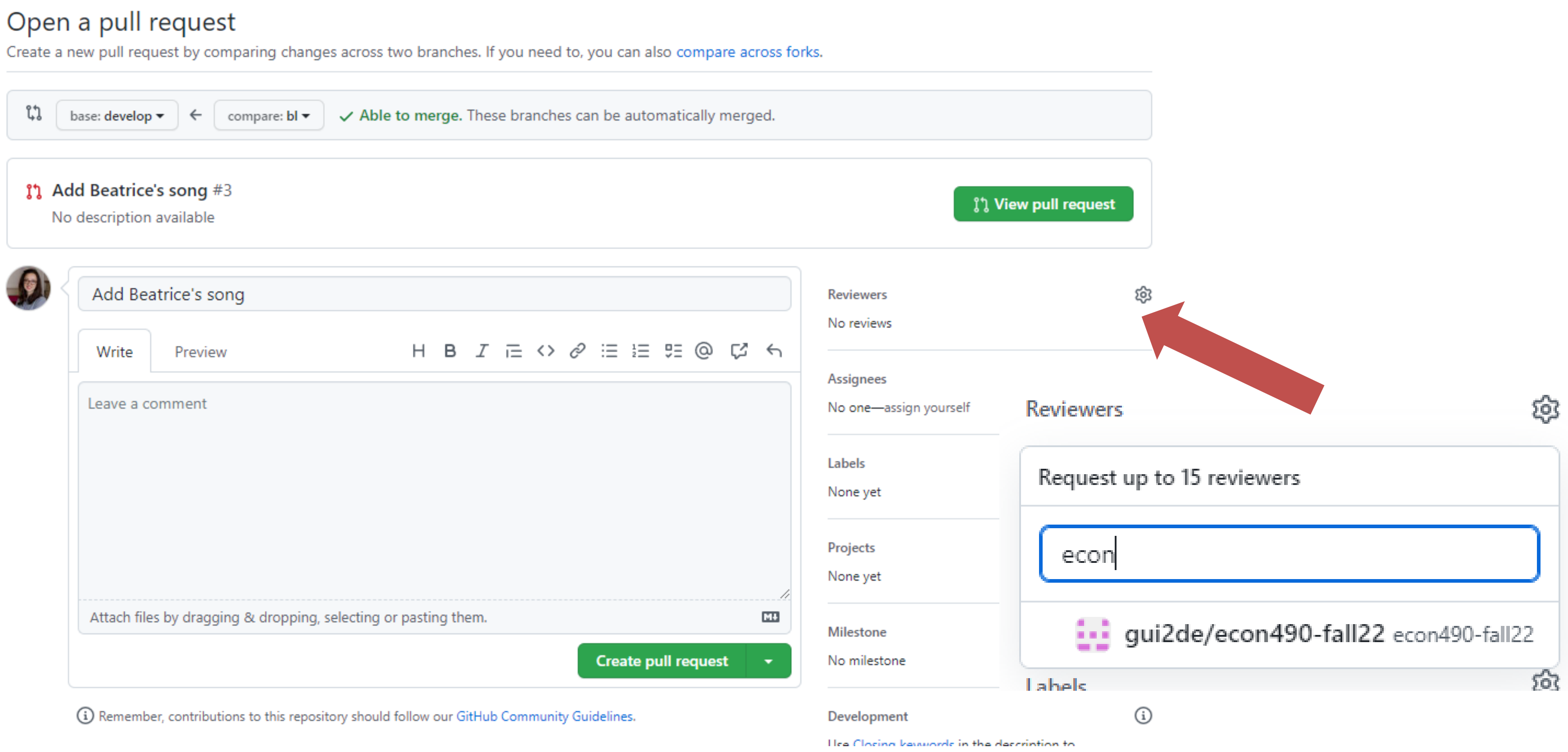
So how does it *work*?

Part 5: Reviewing and commenting Pull Requests

GEORGETOWN UNIVERSITY

Post Class Assignment

Go to the remote repo on github.com to create a Pull Request (PR) of your branch into the develop branch, and assign a random reviewer to it by tagging ppol768-spring23 as a reviewer *(it will randomly select another student of the class as a reviewer)*


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: develop ← compare: bl ✓ Able to merge. These branches can be automatically merged.

Add Beatrice's song #3

No description available

View pull request

Add Beatrice's song

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

Development

Reviewers

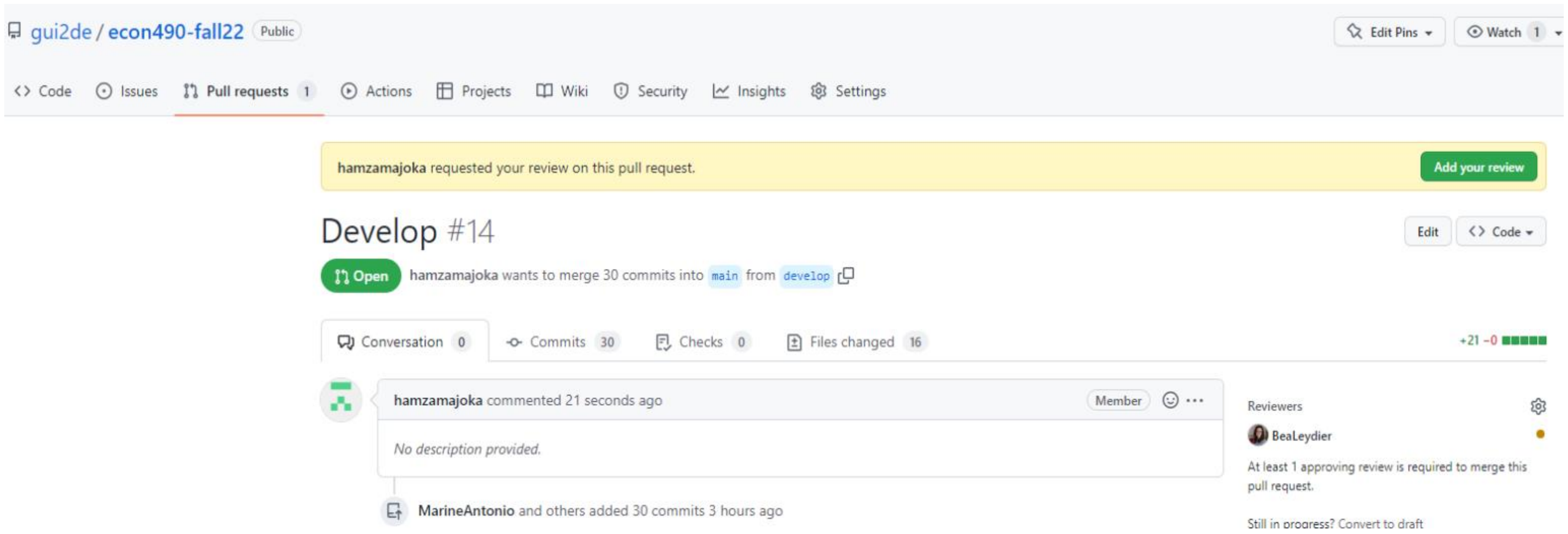
Request up to 15 reviewers

econ

gui2de/econ490-fall22 econ490-fall22

Reviewer : review someone else's work

1. Github should automatically assign you someone else's PR (Pull Request) to review. You can find it under the Pull Requests tab of the repo on github.com


gui2de / econ490-fall22 Public

Edit Pins · Watch 1

Code · Issues · Pull requests 1 · Actions · Projects · Wiki · Security · Insights · Settings

hamzamajoka requested your review on this pull request. Add your review

Develop #14

Edit · Code

Open hamzamajoka wants to merge 30 commits into main from develop

Conversation 0 · Commits 30 · Checks 0 · Files changed 16 · +21 −0

hamzamajoka commented 21 seconds ago · Member

No description provided.

MarineAntonio and others added 30 commits 3 hours ago

Reviewers

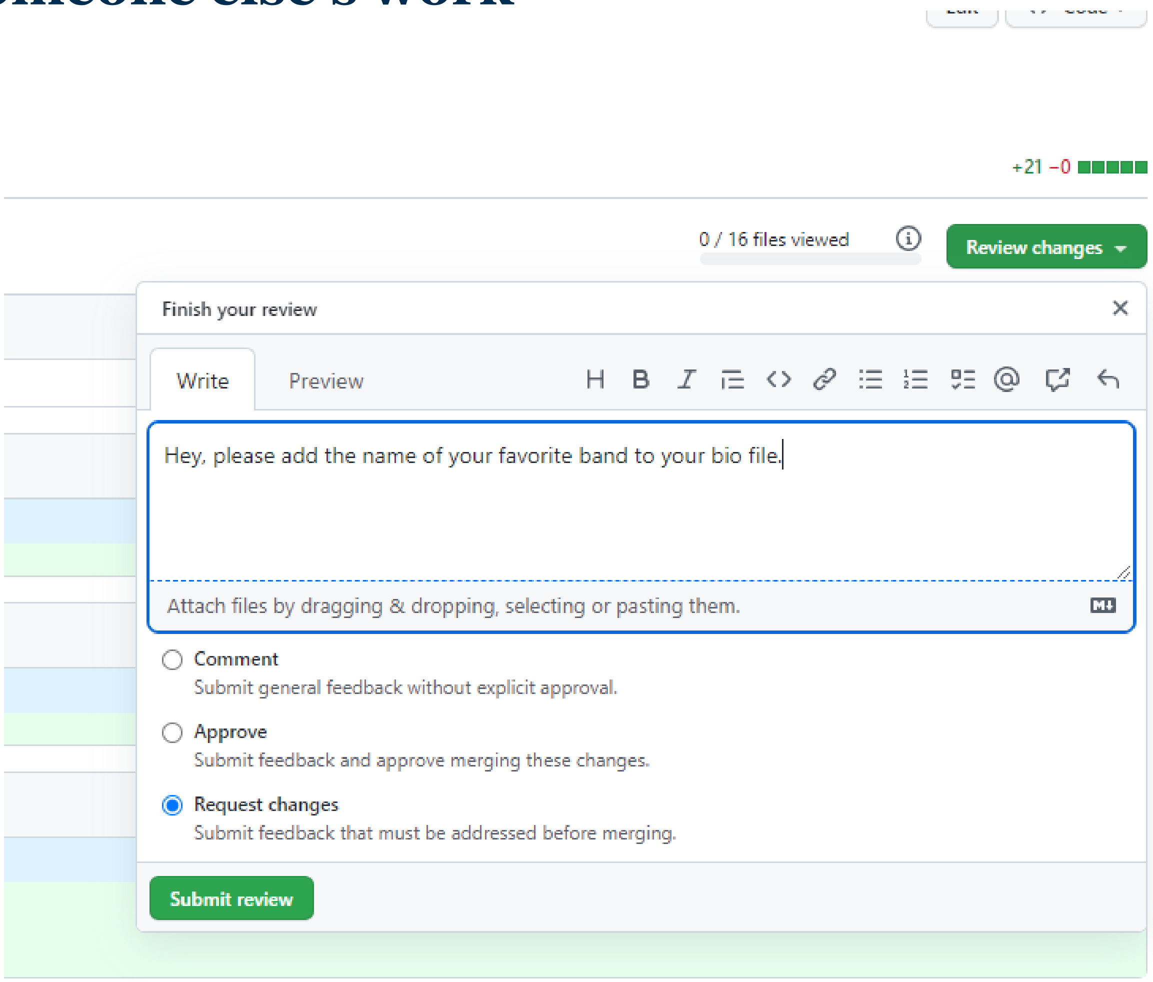
BeaLeydier

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

Reviewer : review someone else’s work

2. Click on Add your Review, review the files and include a comment **requesting a change** to the files. You can simply ask another piece of information to be added, for example.



+21 −0

0 / 16 files viewed

Finish your review

Hey, please add the name of your favorite band to your bio file.

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

○ Comment
Submit general feedback without explicit approval.

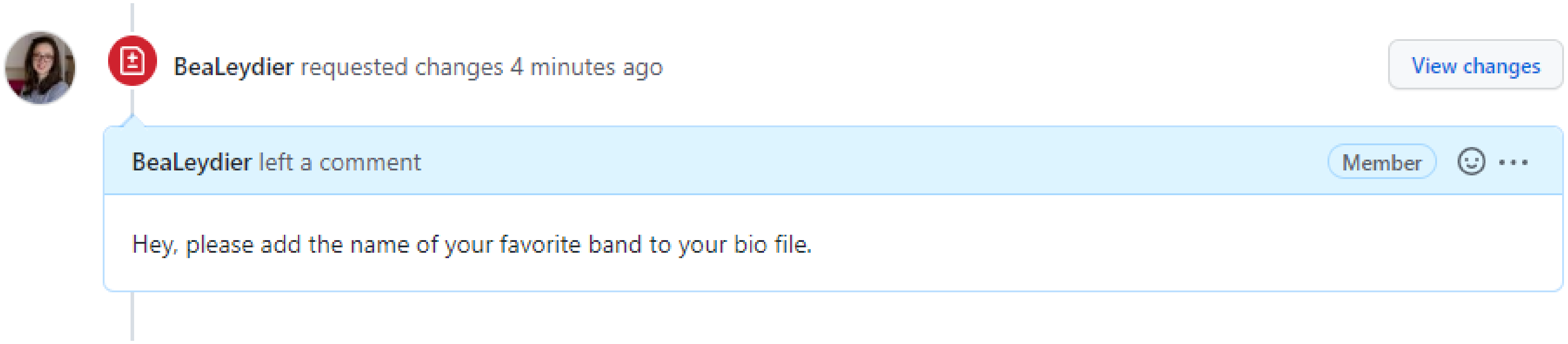
○ Approve
Submit feedback and approve merging these changes.

● Request changes
Submit feedback that must be addressed before merging.

Reviewee : make the requested changes

You will see in the history of the Pull Request the requested changes as a comment.

Address the changes them by making changes on your local repo on the same branch and pushing them to the remote repo, then notifying your reviewer that you have addressed the changes.

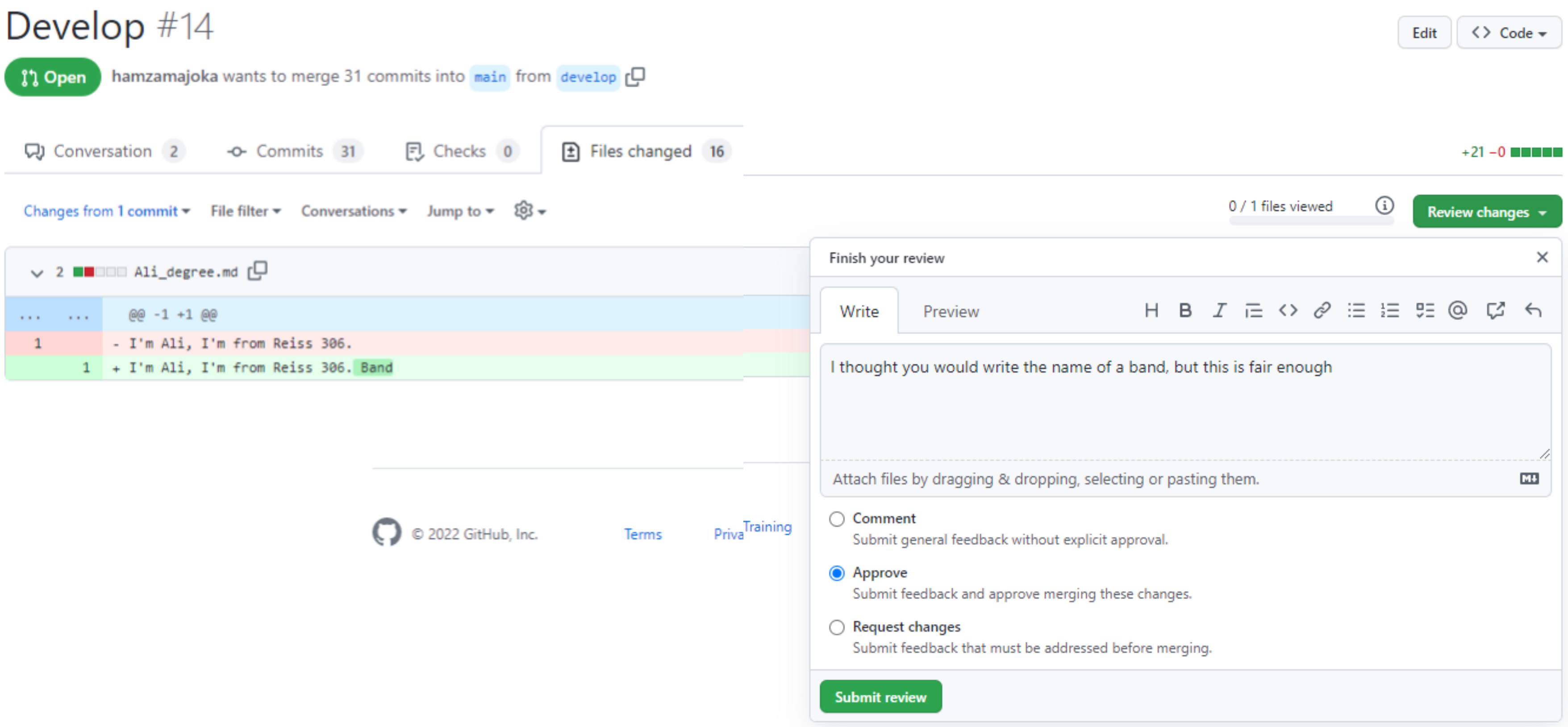

BeaLeydier requested changes 4 minutes ago

BeaLeydier left a comment

Hey, please add the name of your favorite band to your bio file.

Reviewer : review the response to your comments, approve

You will see in the history of the Pull Request that new changes have been pushed to the branch. You can **review the changes, make a final comment and approve them** for the final merge to be completed (if they are satisfactory, though they should be).



hamzamajoka wants to merge 31 commits into `main` from `develop`

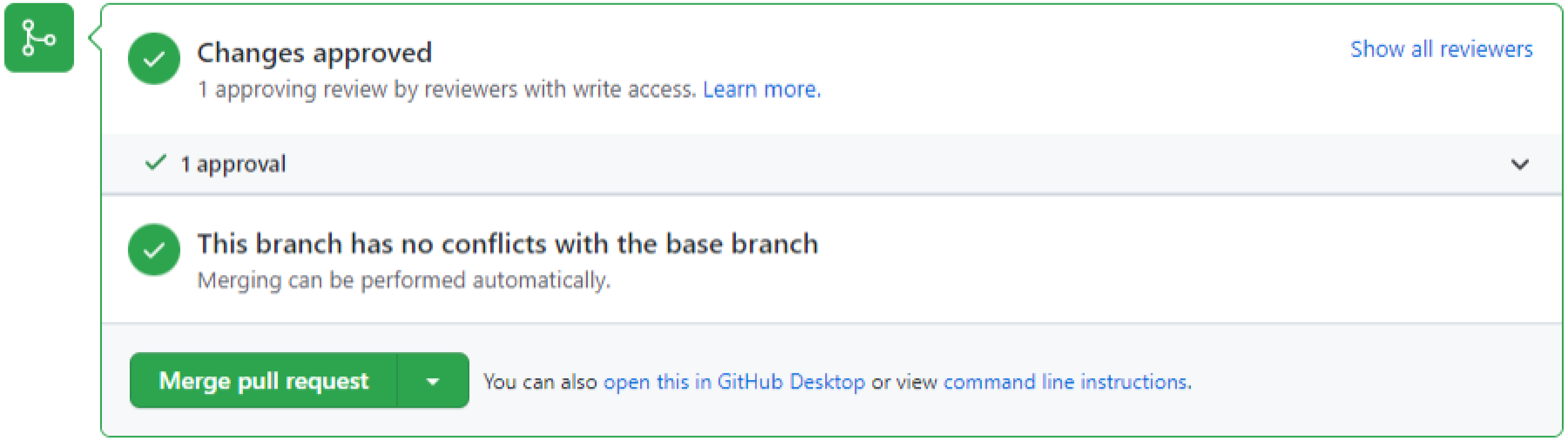
```
2 Ali_degree.md
@@ -1 +1 @@
1   - I'm Ali, I'm from Reiss 306.
  1 + I'm Ali, I'm from Reiss 306. Band
```

I thought you would write the name of a band, but this is fair enough

- ( ) Comment
  Submit general feedback without explicit approval.
- (●) Approve
  Submit feedback and approve merging these changes.
- ( ) Request changes
  Submit feedback that must be addressed before merging.

Reviewee : finally merge your branch into develop

Once your reviewer has approved your edits, you will be able to **merge the pull request** (go to the Pull Request tab on the repo on github.com) onto the develop branch. Complete the merge and delete your branch on the remote repo. Back to the local repo (on GitKraken), check out the develop branch and pull to update it. You should see your changes reflected. You can delete the branch you had created on your local repo (git won't do it automatically even if it is deleted on the remote repo).



Changes approved
1 approving review by reviewers with write access. Learn more.

Show all reviewers

✓ 1 approval

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.

I am stuck and nothing works

- **Burn everything and start over**
  - Delete your local repo
    - Save any uncommitted changes (ie the files you made changes to) in a separate location on your computer
  - Reclone your repo locally from the remote/origin repo
  - Create a new branch and add your changes again
- As long as you don’t delete the remote repo, and don’t discard uncommitted changes, **it’s hard to really mess anything up** with github

Georgetown University Initiative

gui²de

on Innovation, Development and Evaluation

www.gui2de.org