---- Git----

Contents

We can use gitbash and git GUI to track the changes in a project folder on our local machine.

To work with gitbash, download it from the link below: https://git-scm.com/downloads

Going with gitbash rather than the GUI as this will be the same in the terminals in cloud IDE's.

You can then PUSH the changes to Github.com where they can be stored, and shared with others. We can also do this cloud IDE’s and most other frameworks.

emer.thornbury@ncirl.ie

---- New Repo ----

There are 2 strands to this process:

1. Setting up a repository (repo, or project) in Github.com

2. Initialising and pushing your project (wherever it is, locally on your machine or in a cloud IDE) to Github.com

1: Setting up in github

Login to www.github.com

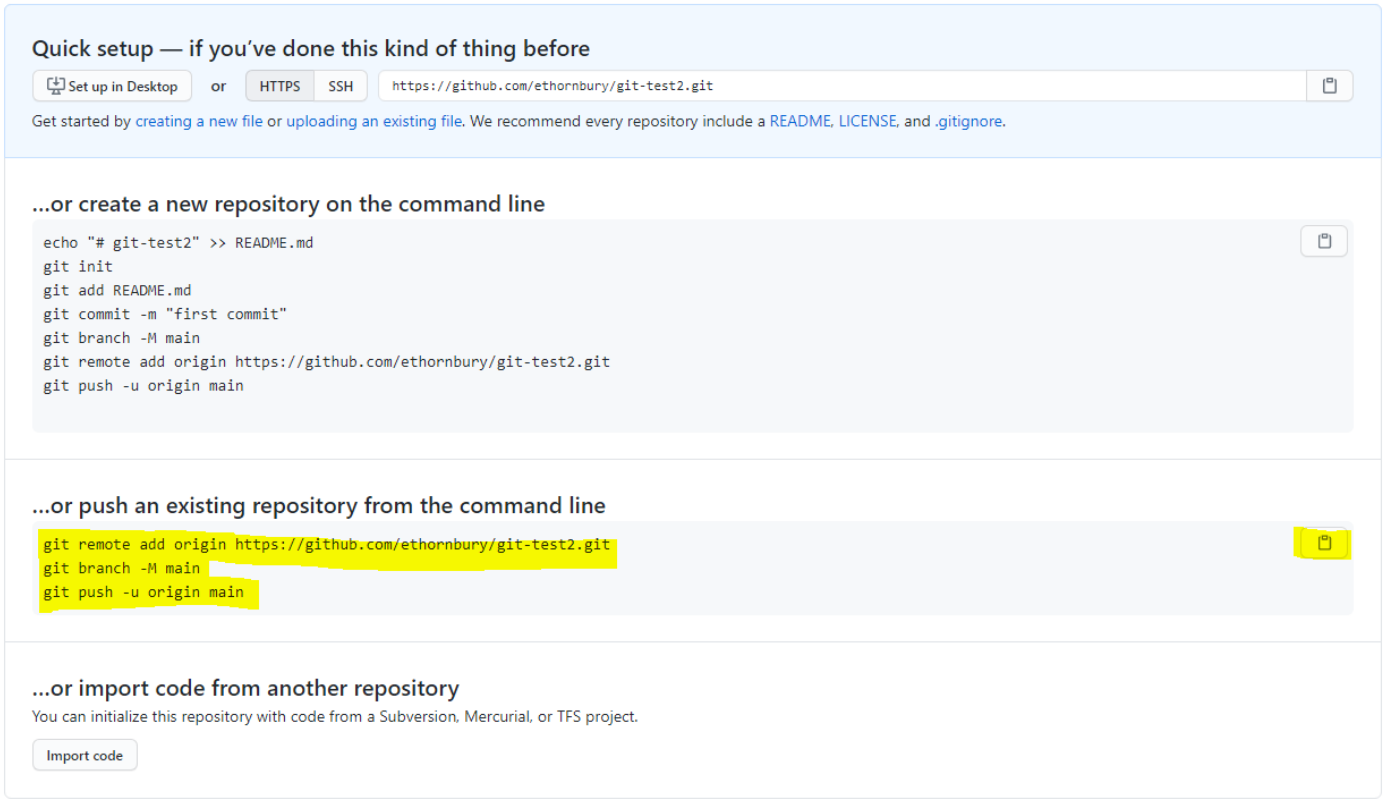
Sign in to Github.com and create a new repository

Name your project and leave all the default settings, click CREATE

You will then see various sets of instructions

We will use the option for an existing project (see screen shot below the table)

git remote add origin	This line connects your project to Github
git push -u origin main	This pushes code to Github (to the main branch)



emer.thornbury@ncirl.ie

2: Initialising and pushing your project

You MUST be inside your project folder.

(If working on your local folder on your machine, right click in folder and choose gitbash option, a terminal will open)

(If working in a cloud IDE, go to a terminal)

In gitbash/terminal type the following commands:

git init	This will initialise a new repo, you only need this once
git add .	This will add the files to be committed and pushed to Github
git commit -m "type your own commit message between quotes"	This will commit them
git remote add origin	This line that links your local project to Github, you only need it once
git branch -M main	Setting the main branch
git push origin main	Pushes code to Github, you may be asked for your Github username and password. In the browser, refresh the screen to see your repo updated in github.com

---- the essential 3 steps for getting code to github.com ----

git add .	to add all files
git commit -m "message"	Commit and send text with the commit
git push origin main	Push your files to the origin (Github) main (main/master branch)

---- Some other essential commands are ----

git status	To see if you have files modified to be committed
git remote -v	to show if the project is linked to github.com and which repo it's linked to
git log	To see your history of commits

---- For branching ----

git checkout -b mybranchname	this line will create and move you to the new branch
git add .	Add files to be commited and pushed
git commit -m "message"	Commit and send text with the commit
git push origin mybranchname	Push your files to the origin (Github) but to mybranchname (branch)
git checkout main	Move to main branch
git merge mybranchname	Will merge your saved branch with the main branch
git push origin main	*** you do need to save the newly merged work!!

emer.thornbury@ncirl.ie

---- Other useful bits for collaborating ----

The person who sets up the project can add collaborators. Go in to the repo on Github, in SETTINGS, click COLLABORATORS and add Github usernames.

An email will be generated and sent. When you accept an invite to collaborate in the email from Github then you will be taken and see the project. There you will see a button with CLONE OR DOWNLOAD ZIP

git clone http....	This command will allow you to take and create the project in codeanywhere or c9.io. You need to be in the main directory and the command will create the project directory for you
git pull	Before you start working when collaborating, you will need to PULL first which means you take down the current main/master branch from Github any new code from collaborators

---- Managing conflicts ----

If you go to push OR pull the file to or from Github and an error reads “merge conflict”.

You need to do the following:

- Open file in your editor.
- To see the beginning of the merge conflict in your file, search the file for the conflict marker <<<<<<<.
- You'll see the changes from the HEAD or base branch after the line <<<<<<< HEAD.
- Next, you'll see =======, which divides your changes from the changes in the other branch, followed by >>>>>>> BRANCH-NAME.
- This means you will see your changes and the conflicting changes.
- Decide what you want to keep and delete the remaining code.
- Delete the conflict markers <<<<<<<, =======, >>>>>>> and make the changes you want in the final merge.
- Save your file and follow the steps below:

git add filename	to add all files
git commit -m "message"	Commit and send text with the commit
git push origin main/branchname	Push your files to the origin (Github) main (main/master branch or branch)

emer.thornbury@ncirl.ie

---- Troubleshooting common issues ----

git remote set-url http://github.com/username/git-repo.git	to reset the url of the project you are connecting to on Github
git config –global user.name “githubUsername”	If you get error that it doesn’t know you do this and the next command
git config –global user.email “email@email.com”	If you get error that it doesn’t know you do this and the above command

---- Useful Links ----

Sam Cogan’s YouTube playlist on getting started on Gitgub

https://www.youtube.com/watch?v=riv6cIpFcC8&list=PLccPlNFtq5uEImaHGI_qA1NLg1mktA1Op

Cheat Sheet

https://services.github.com/kit/downloads/github-git-cheat-sheet.pdf

Nice display of commands

https://githowto.com/history/

emer.thornbury@ncirl.ie