GIS4x07 GitHub and exercise folder setup

# Intro

This document contains the instructions for creating local and remote GitHub repositories for GIS4x07 exercises. This example uses gis4207-week02. You will replace this with the appropriate course/day (e.g. gis4107-week04, gis4207-week02, etc.) as specified in the exercise documentation.

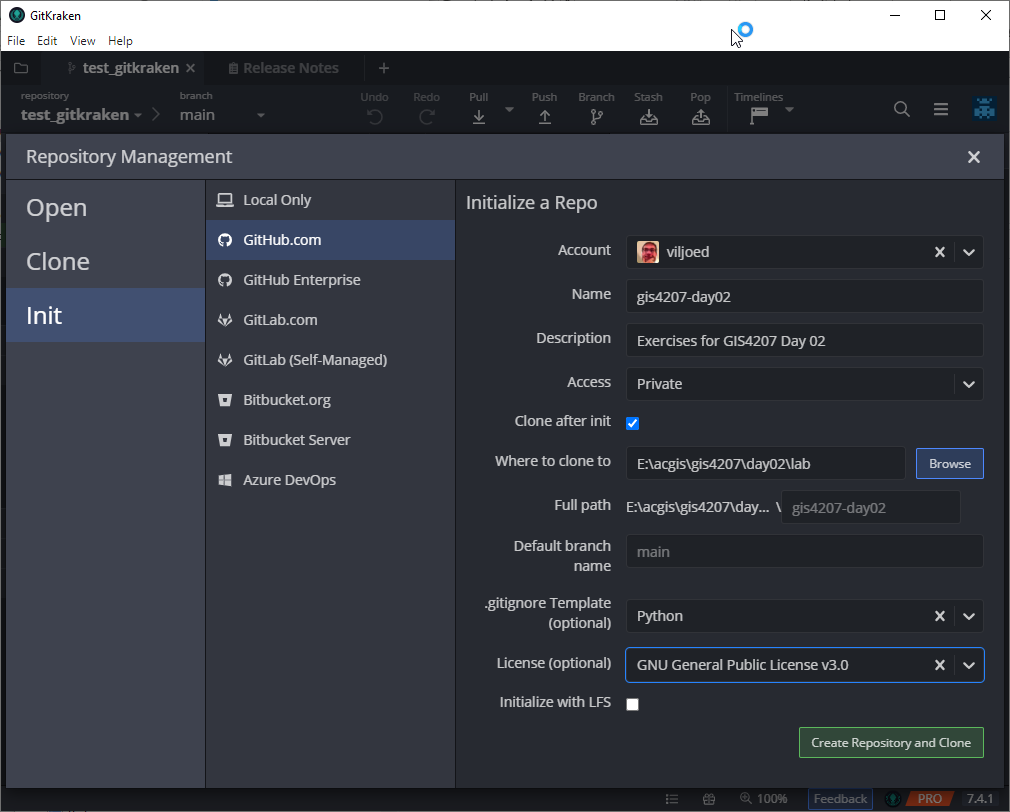
# Initial repository setup

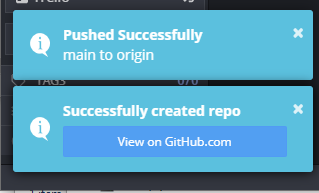
The repository owner specified by the instructor will do the initial repository setup. Their partner will standby/observe until this process is complete.

The owner will:

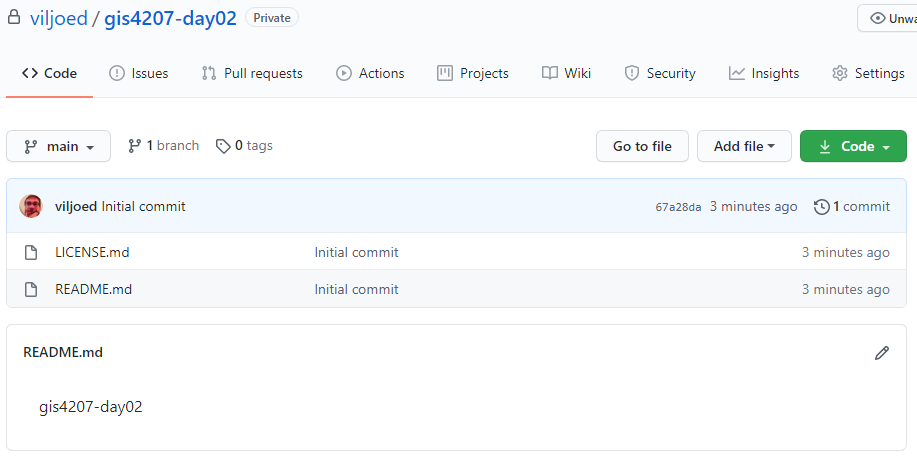
1. Open GitKraken
2. Press **Ctrl+I** to initialize a repository (or Init Repo under the File menu). This opens the “Repository Management” page with Init selected.
3. Select GitHub.com as the remote host.
4. In the “Initialize a Repo” panel:

* Select your Account (e.g. acgis-abcd1234)
* Specify a name for the local and remote repository exactly as specified in the exercise documentation. In the example shown below, gis4207-day02.
* Description is optional
* Access is Private. Public repositories are visible to all GitHub users.
* “Clone after init” should be checked by default. Leave it checked.
* “Where to clone to” should be the “lab” folder specified in the exercise documentation. In the example shown below, “E:\acgis\gis4207\day02\lab”
* Leave “Full Path” and “Default branch name” as their default values
* For .gitignore Template, choose Python. This will ensure folders like \_\_pycache\_\_ and files like \*.pyc are not included in the repository. There is no value in having these under version control.
* Leave License unselected or pick one if you are interested. GNU General Public License v3.0 is popular for open source projects.
* Leave Initialize with LFS unchecked. No large files will be under version control in these exercises so there is no need to initialize the repository with support for “Large File Storage”
* Click “Create Repository and Clone”





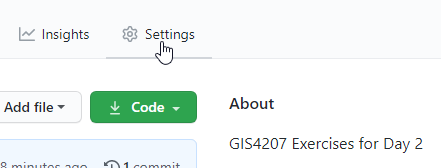
In GitHub you will see the new repository as shown below.



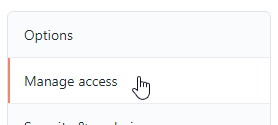
Note: viljoed will be replaced with the GitHub username of the repository owner (e.g. abcd1234)

You have now successfully created a Private repository.  Add viljoed and your partner as collaborators as follows:

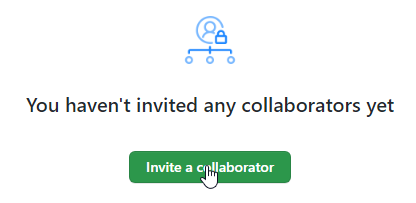
Select the Settings tab for the repository …



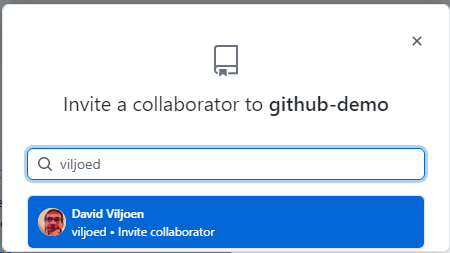
Select “Manage access” from left menu …



Click “Invite a collaborator”

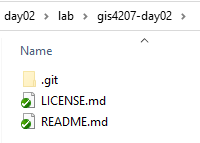


Search / select /add collaborator …



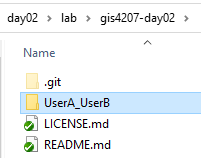
Search / select / add your partner as a collaborator.

On your local computer you will find the local repository in the folder you specified with GitKraken, e.g. E:\acgis\gis4207\day02\lab\gis4207-day02

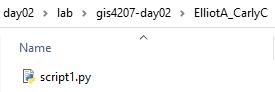


If you do not see the .git folder, refer to “[Show hidden files, folders, and drives](#_Show_hidden_files,)” at the end of this document.

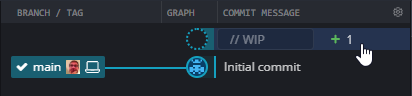
Create a folder in this working directory (tree) for you and your partner, e.g.



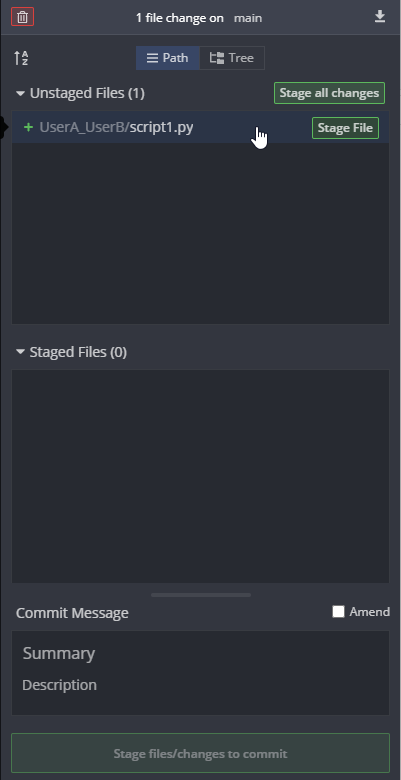
Where UserA\_UserB will be replaced with the first name and last initial of you and your partner (e.g. ElliotA\_CarlyC). This folder will need a file in that folder before you stage, commit, and push to GitHub. Create an empty file with the name of the first script in the exercise. If it was “script1.py”, then your folder would now look like:



In GitKraken, above the Initial commit, you will see a +1 signifying that one new file has been added:



If you click the “+1”, the side panel show below will be displayed:

The trash can in the upper left allows you to discard all changes since the last commit. You do not want to do that in this case.

If you mouse over the new file, you have the option to stage that file. If you had added/modified/deleted more than one file, you could stage them individually.

Recall, staging is specifying that the added/modified/deleted file will be included in the next commit.

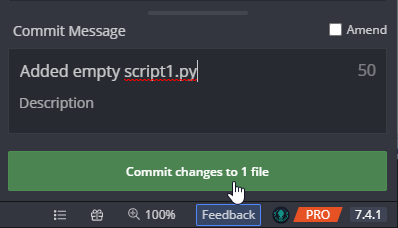
In this case, “Stage all changes” or “Stage File” will have the same result since there has only been one change (adding script1.py).

After staging, you do have the option to “Unstage all changes” or “Unstage file”.

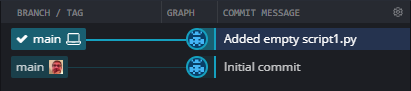
Next step is to commit the change to the local repository. Clicking on “Summary” in the “Commit Message” pane shows the number of remaining characters in the Summary (max 72). As soon as you start typing, the word “Summary” disappears.

For such a simple commit, a summary is all that is required. A more detailed description is optional.

You commit message might look like the following:

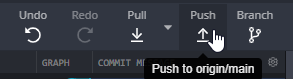


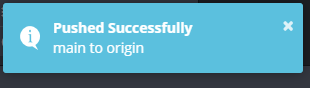
Click “Commit changes to 1 file” button to commit the change to the local repository. Now you will see the first commit in the main branch:



You will also notice to the left, that the main branch in local is “1 ahead” (i.e. 1 commit ahead) of the remote repository. This indicates you need to push your changes to the remote.

|  |  |
| --- | --- |
| Before Push: | After Push: |
|  |  |

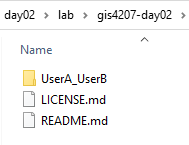
  
To push the changes to the remote (i.e. origin/main)

After the push is complete, an information message box will appear in the lower left corner

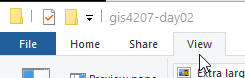
Collaborators can now fork or clone this repository. More on forking/cloning in the exercise documentation.

# Show hidden files, folders, and drives

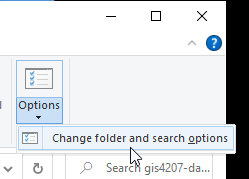
Local git repositories are stored in a .git sub-folder. If you do not see it in File Explorer, e.g.



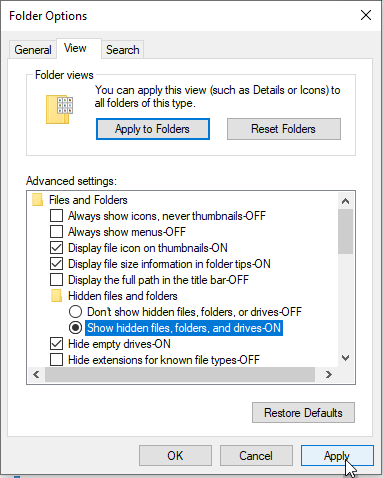
1. Select the View Tab in File Explorer



1. Select “Change folder and search options”



1. Select the View tab and “Show hidden files, folders, and drives” as shown below



1. Click Apply to see the change and click Ok to close the Folder Options dialog. The hidden .git folder will display with a slightly faded icon

