

CSCE 120: Learning To Code

Organizing Code II Hacktivity 13.2

Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

Make sure you and your partner each have a copy of the completed code for the Album App from the previous hacktivity. If you've lost it, you can use the code from the initial project available at <https://github.com/cbourne/AlbumsApp>.

In this hacktivity we will use GitHub to share and collaborate on a project. Be sure to have registered with GitHub and downloaded and installed the GitHub Desktop Client. For the following activities, we will identify each of you as either partner *A* or partner *B*.

1 Sharing Your Project

The following should be done by partner *A*. In this activity you will publish your Album App to GitHub.

1. Open your GitHub client. Once it is open you can drag and drop your project folder into the GitHub client app. Click the "Create and Add" button.

2. Upon success, the Git client should appear similar to Figure 1; you can now make an initial commit by filling in the commit message and description and clicking “Commit to master”

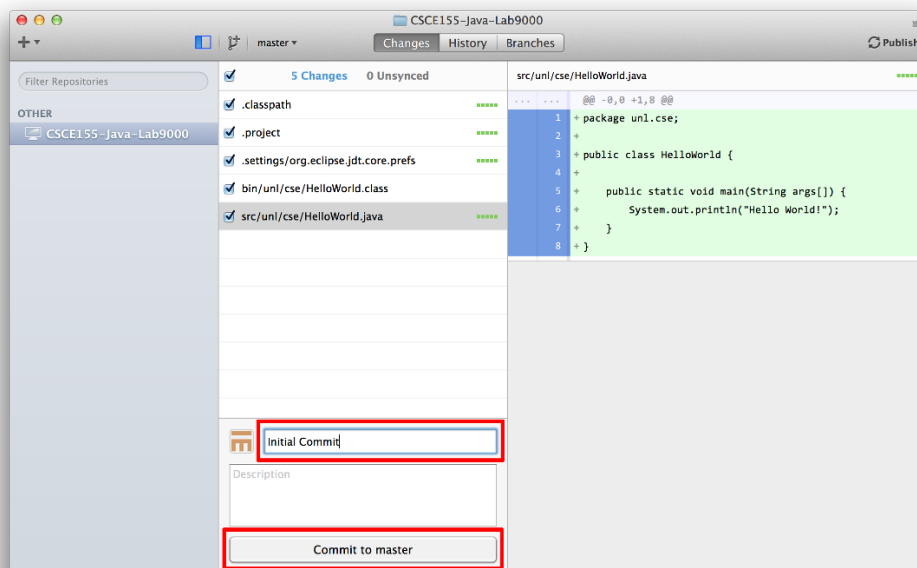
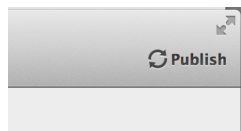
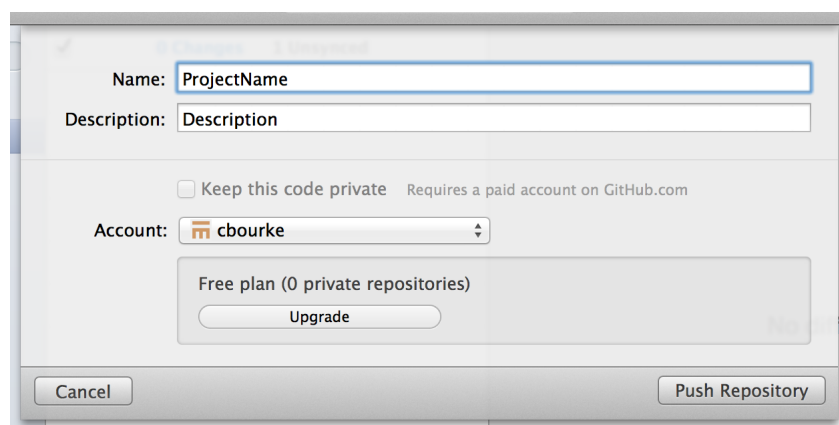


Figure 1: Committing in the GitHub Client

3. You can now “publish” your repository to GitHub by clicking the “Publish” icon in the top right of the Git client



4. This opens a new dialog where you can specify the name (use AlbumsApp) and description of the project as it will appear on GitHub.

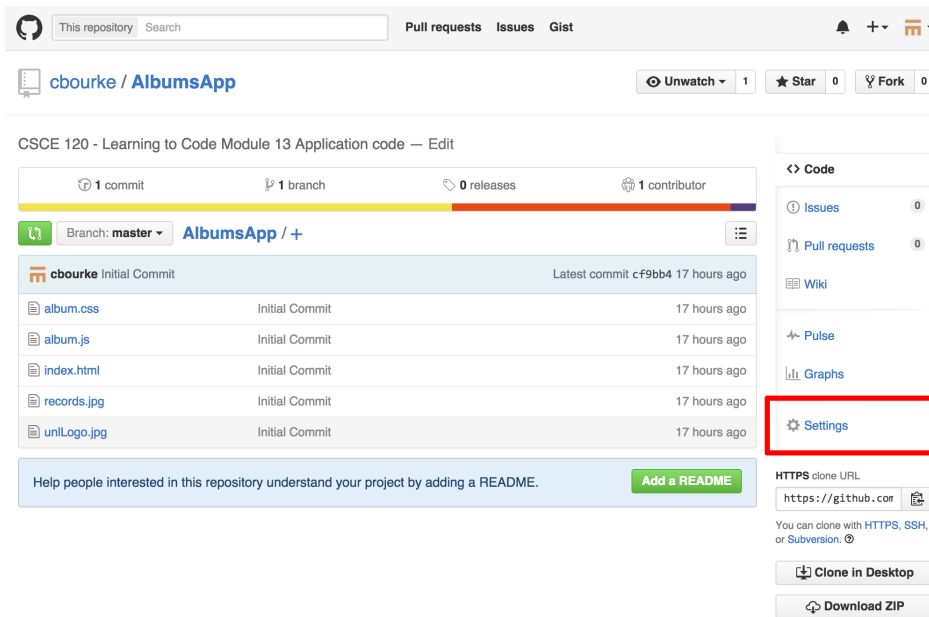


To finish up, click the “Push Repository” button and observe your new project on GitHub

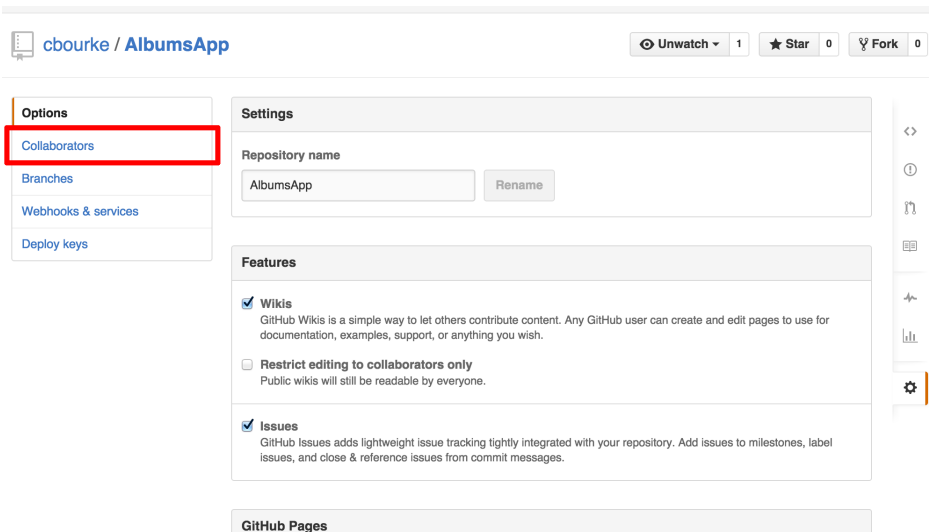
2 Granting Access to Collaborate

Your project is *public* on GitHub: everyone, even if they are not registered on GitHub can find and download your code. However, no one can modify the code on your repository unless you allow them to. We will now grant read *and* write access to your partner.

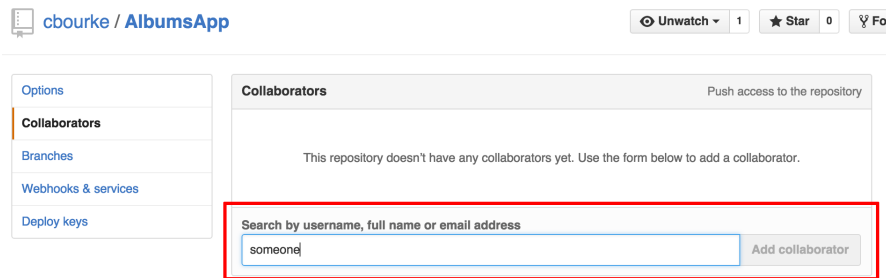
1. Go to your project’s GitHub page and click the “Settings” area:



2. Click the “Collaborators” area:



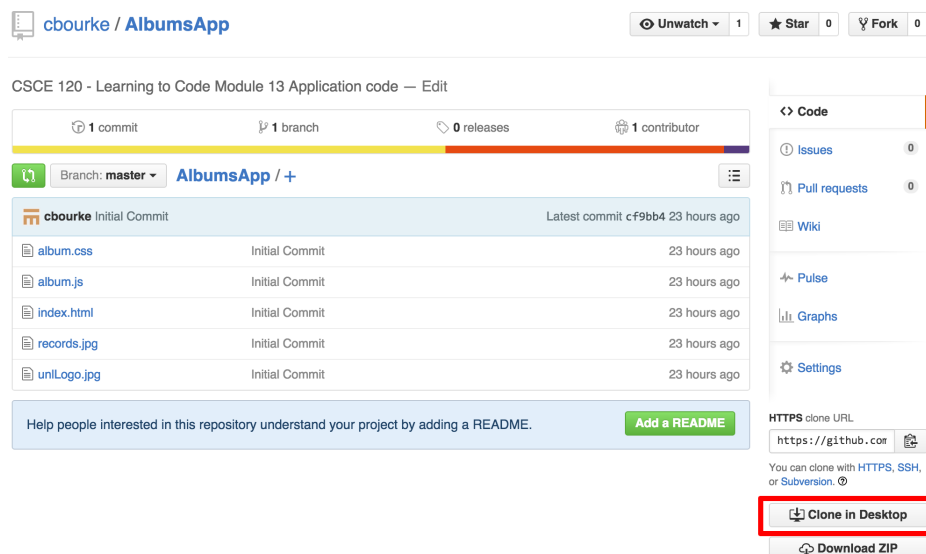
3. Enter your partner's GitHub user name or name (it will search for you) and add them as a collaborator:



3 Checking Out And Making Changes

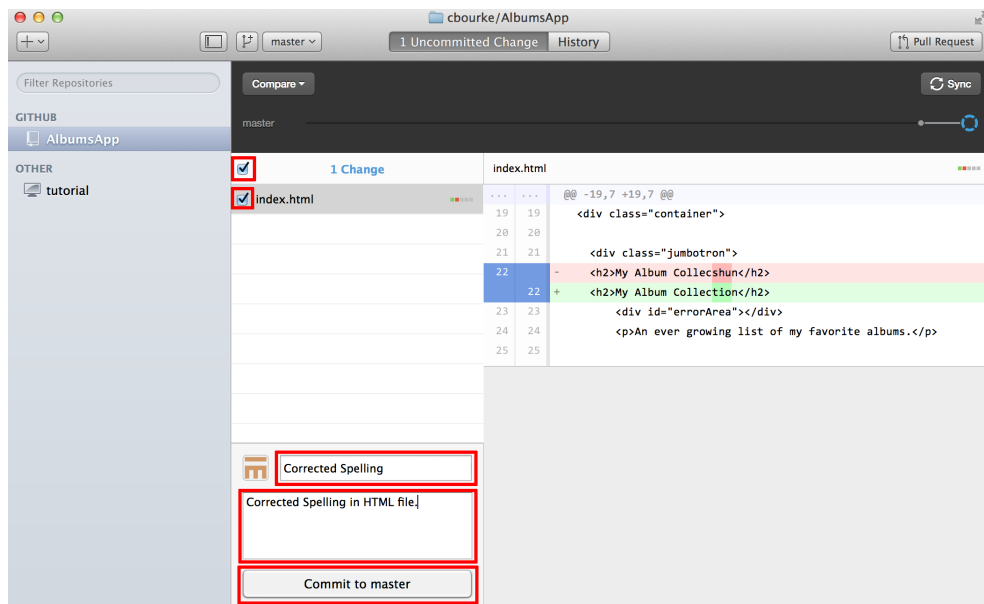
Now partner *B* will *clone* the project from GitHub so that they can make contributions and changes, “pushing” those changes back to the repository on GitHub.

1. Go to partner *A*'s GitHub project page (likely something like <https://github.com/login/AlbumsApp>) and click on the “Clone in Desktop” link. This will launch your GitHub Desktop client if it is not already open.



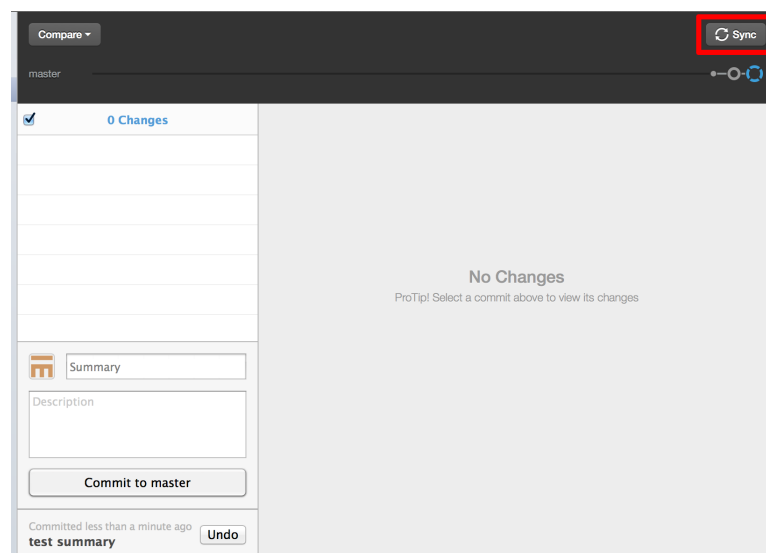
You will need to specify a location for the project folder to be saved to.

2. Once the files have been downloaded you can start working with them. Open the project in Light Table and make a change. In particular, correct the misspelling in the HTML file (“Collecshun” → “Collection”).
3. Navigate back to your GitHub Desktop Client, it will have the changes marked and look something like the following.



Commit and push the changes to the repository on GitHub:

- a) Make sure all changed files' checkboxes are marked.
 - b) Fill in a summary of the changes
 - c) Fill in a more detailed description of the changes
 - d) Click the “Commit to Master” button.
4. After you commit all changes, your GitHub client should look something like the following.



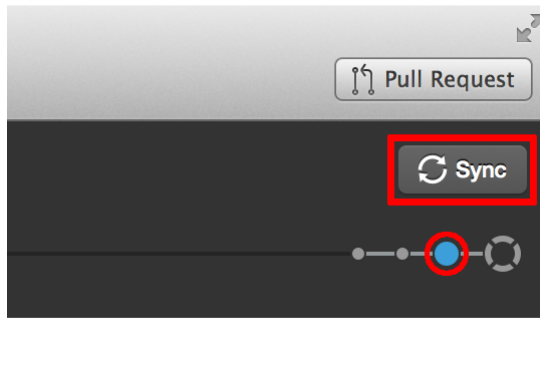
Now we need to *sync* your local version of the project with the one on GitHub by *pushing* the changes to the remote repository. To do this, click the “Sync” button as depicted above.

- Both partners should look at the project on GitHub to verify that the changes (along with the commit message) have been pushed to GitHub.

4 Pulling & Pushing Changes

Partner *A* should now *pull* the changes that partner *B* made.

- Pull the changes your partner just made by clicking the “Sync” button in the top-right of your GitHub client:



Upon success, the changes should be pulled into your project. Click the last update in the project “timeline” as in the picture above to see what changes your partner made on that revision.

- Now make your own changes to the project. In particular, change the `album.js` JavaScript source file by changing the “default” album from the Beatle’s *White Album* to your own favorite album.
- Commit and push the changes to the GitHub repository by repeating steps 3–4 in Activity 3 above.
- Partner *B* should now pull these changes so that everyone has the same copy. To do so, partner *B* should click the “Sync” button in the top-right as above.

5 Repeat

Time permitting, repeat this hacktivity with a different project with the partners *A/B* swapped to get the experience from both perspectives.