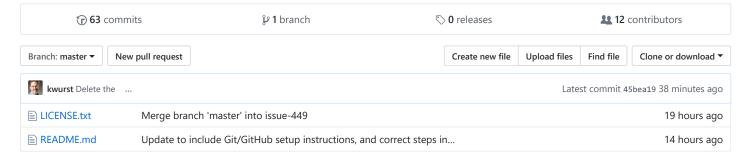
CS-Worcester-CS-348-SP-2018 / git-issue-activity

No description, website, or topics provided.



■ README.md

Version 2018-Spring-1.1-Final, Revised 14 February 2018

CS-348 01, 02 — Spring 2018

GitIssueActivity

Activity for practicing with git and issues on GitHub.

For this activity, the students will practice making changes to a GitHub repository. The repository has a text file that has been "scrambled" - meaning that random words have had their characters randomly rearranged, and considered to be issues to be fixed.

The misspellings were generated by the twiddle.py program and then entered as issues by the report.py program (from this git-twiddler fork.

Prereqisites

- 1. Git installed and configured on your computer.
- 2. GitHub account created and username emailed to professor.
- 3. Invitation to class organization on GitHub accepted.
- 4. SSH key added to GitHub account.

If you have not completed these setup steps, see the instructions at the bottom of this document.

In-Class Activity

- 1. Fork the repository
- 2. Clone the repository
- 3. Add the original repository as an upstream remote: git remote add upstream git@github.com:CS-Worcester-CS-348-SP-2018/git-issue-activity.git
- 4. Repeat as often as possible during the class:
 - i. Assign yourself to an issue on GitHub (e.g. issue #42)
 - ii. Make sure you are on the Master branch: git checkout master
 - iii. Pull any changes that have ocurred since your last pull, and correct merge conflicts git pull upstream master
 - iv. Create a branch named for the issue (e.g. issue-42), and switch to that branch: git checkout -b issue-42

- v. Edit the LICENSE.txt file to correct the issue
- vi. Add the change
- vii. Commit the change, with a message describing the change (include line number and correction, e.g. Correct misspelling, line 1: UGN -> GNU)
- viii. Pull any changes that have ocurred since your last pull, and correct merge conflicts git pull upstream master
- ix. Push the change git push origin issue-42
- x. Make a pull request to have their change merged into the original repository.
 - a. On GitHub switch to the newly pushed branch.
 - b. Click the New Pull Request button.
 - c. Include in the pull request text a notation that it closes the assigned issue (e.g. Closes #42)
 - d. Click Create Pull Request
 - e. Wait for the repository owner to accept the pull request.

Instructions for Setting Up Git and GitHub

Install Git

- 1. Open your browser and go to: http://git-scm.com/downloads
- 2. The picture of the monitor on the page should show the download for your operating system. Click on the download button and save the installer.
- 3. Run the installer.
 - i. On Windows, use the default settings, when given choices.
- 4. Start a command-line/terminal window, as shown below:
 - i. On Windows, go to the Start menu > All Programs > Git > Git Bash
 - ii. On Mac OS X, go to the Applications folder > Utilities folder > Terminal
- 5. Configure Git, so that it knows who you are and can label your changes, and so it colors the output. Type the following commands (replacing Jane Smith and jsmith@worcester.edu with your own name and email address):

```
git config --global user.name "Jane Smith"
git config --global user.email jsmith@worcester.edu
```

Create an SSH Key Pair

SSH (Secure Shell) is a network protocol for encrypted data transmission. It is often used for password-less login to remote servers, and we will be using it to allow you to communicate with the GitLab server without having to type your password every time you want to send code changes to the server. SSH uses public-key encryption, and to use it you need to create a public-private key pair. Your public key is made available to anyone with whom you want to exchange encrypted data. Your private key is used only on your computer and should be protected carefully.

1. Create an ssh key pair by typing:

```
ssh-keygen -t rsa -C jsmith@worcester.edu
```

- i. When it asks you for the filename, use the default, by hitting Enter .
- ii. You should enter a passphrase when prompted. Using a passphrase will assure that no one else can use your private key, even if they gain access to the key file. When you type your passphrase, your typing will not show not even as or *.
- 2. Keep your Git Bash or Terminal window open.

Create Your GitHub Account

- 1. Return to your browser and go to http://github.com
- 2. Create an account. You may choose any username you wish, and use any email address you wish.
- 3. One you have completed the account creation process, let your instructor know what username you chose. You will be added to a group for the course.

- 4. Add your SSH key:
 - i. Go to the *Settings* by clicking on the dropdown under your avatar at the top right of the screen. (That circle can hold a picture of you if you set an avatar under *Settings*")
 - ii. Click on the SSH and GPG Keys in the menu on the left.
 - iii. Clik on the green New SSH Key button.
 - iv. Copy your public key to the clipboard.
 - a. Go back to your terminal window.
 - b. On Windows type: notepad ~/.ssh/id_rsa.pub
 - a. In the Notepad window, select all the text and copy it.
 - c. On Mac OS X, type: pbcopy < ~/.ssh/id_rsa.pub</pre>
 - v. Switch back to your browser and paste your key into the key box.
 - vi. Type a title for your key. (It can be anything you want.)
 - vii. Click the green Add SSH Key button.
 - viii. If you are on Windows, close Notepad.

Copyright and License

©2018 Karl R. Wurst, Worcester State University

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.