Working With a Shared Repository

tutorial #tutorial02a

James L. Parry
B.C. Institute of Technology

Tutorial Goals

This tutorial will walk you through working with a shared repository.

The goal is to add your name to the appropriate list, in the appropriate order, on the Winter 2015 Students list.

That can't be so hard, can it? <-- said with a straight face.

Suggestion: you may want to skim the slideshow first, before working your way through it.

What's So Hard About This?

1. Fork a github project
2. Clone your fork locally
3. Create a topic branch
4. Commit changes to your branch
5. Push your changes to your fork
6. Send a pull request to the original project

Make Sure You Are Signing

Many projects insist that submitted code be "signed", so they can be assured of the identity of the contributer.

The first part of this is making sure that Git is configured with your name and email. This might be doable through your IDE, or you might have to use the command prompt, shown right.

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

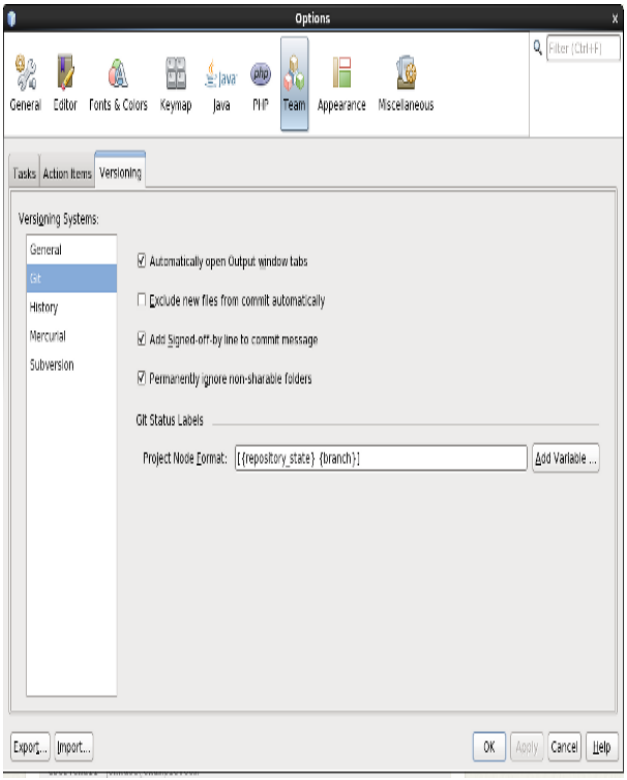
```
$ git config --list
user.name=John Doe
user.email=johndoe@example.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

Tell NetBeans to Add a Signoff

In Tools/Options/Team, set the Git open to "Add Signed-off-by line to any commit message automatically.

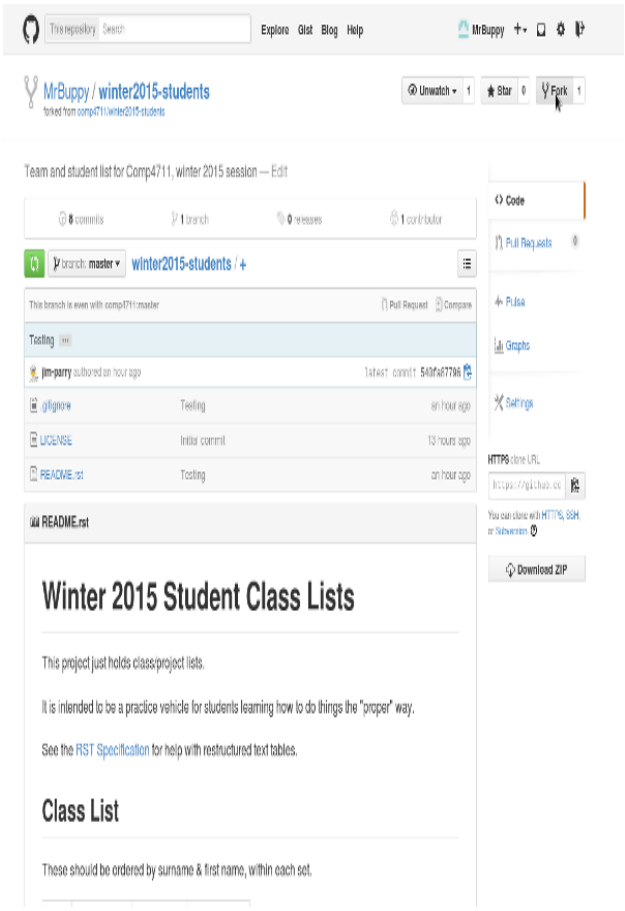
Many projects insist that submitted code be "signed", so they can be assured of the identity of the contributer.

This signoff might be acceptable for many projects, but some will insist that commits be "digitally signed". This process is more involved than you need for our purposes.



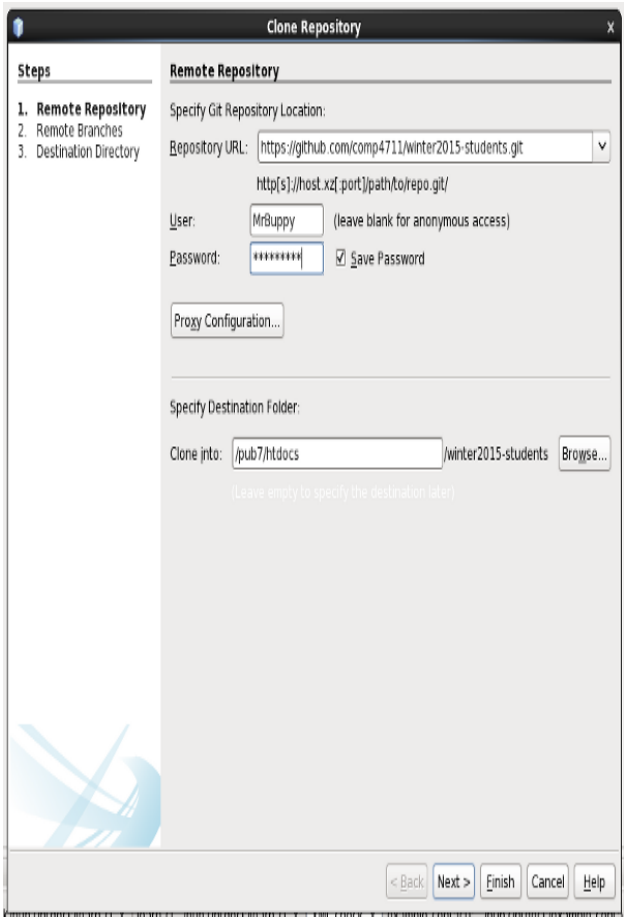
Fork Our Project

On the Github website, fork the Winter 2015 Students list repository setup for this exercise.



Clone your fork locally

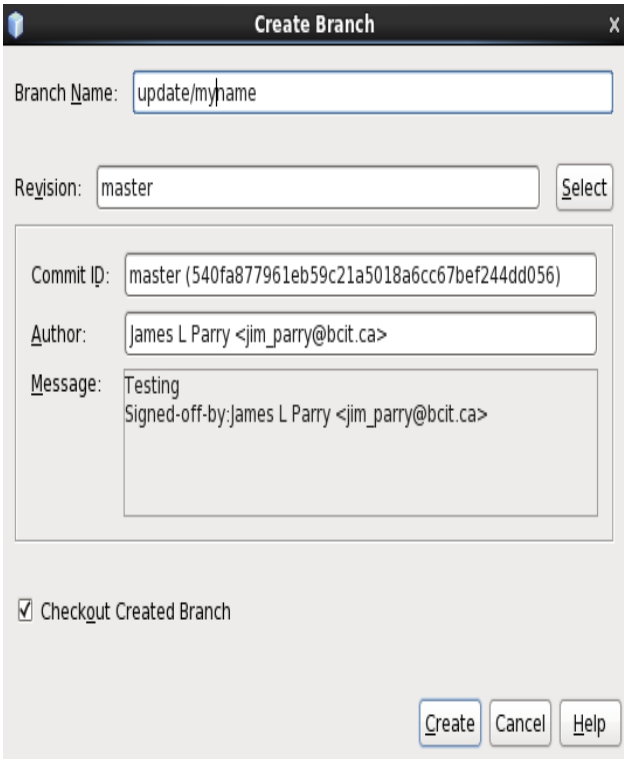
You should remember this from last week :)



Create a topic branch

Create a new branch, based on "master" and check it out.

If you forget the checkout, things will likely get messed up :(



Make Your Changes

Add your name to README.rst, at the right spot.

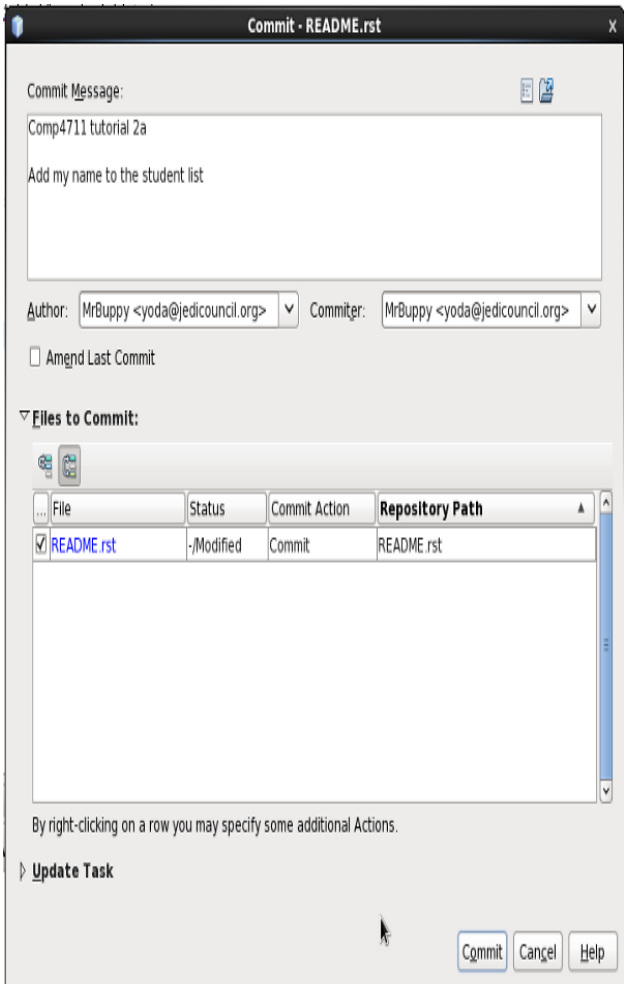
Notice that I had to expand the width of "column 1", so that the column boundaries would line up.

```
**********
Class List
**********

These should be ordered by surname & first name, w

===  ========  =======  =============
Set  Username  Surname  First Name
===  ========  =======  =============
===  ========  =======  =============

*********
 These should be ordered by surname & first name

====  ========  =======  =============
Set   Username  Surname  First Name
====  ========  =======  =============
Jedi  MrBuppy   Yoda     Minch
====  ========  =======  =============
```

Commit changes to your branch

You should remember this from last week, too!

Remember to make sure your commit is signed, and that it has an appropriate message.

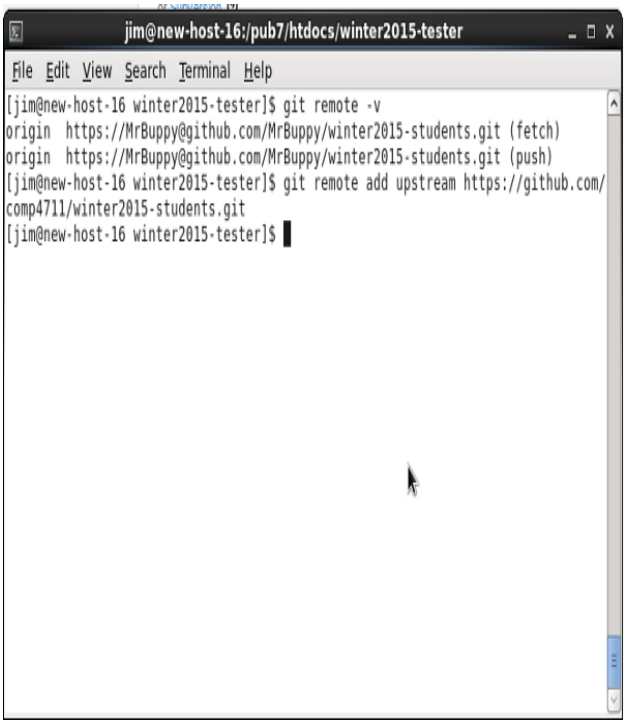


Synchronize Your Fork With the Main Repo

The first time you do this, you will need to add the remote repository, and you should call it "upstream".

You can then synchronize your repo by

1. Checkout the "master" branch
2. git pull upstream master
3. git push origin master
4. Checkout your topic branch again.

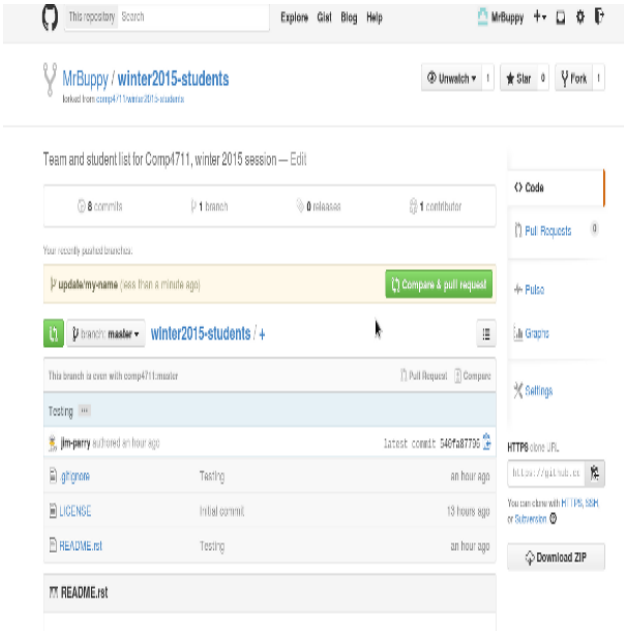


Push your changes to your fork

From your IDE, or the command line, push your changes to your server-side repository.

Send a pull request to the original project

If you switch to your github repository in your browser, you will see that github is offering to "compare & pull request".



Initiate the Compare & Pull Request

Select the "Compare & pull request" link.

Near the top of the page, it should say that the pull request can be automaticaly merged. If it doesn't, you did something wrong.

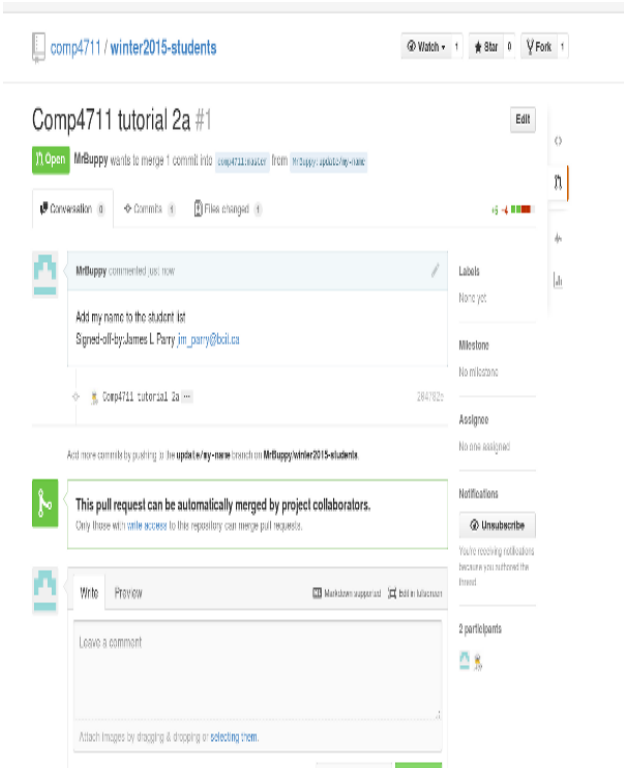
Scroll down to the bottom of the page displayed, and make sure that the changes are what you expect.

If all looks ok, click "Create pull request".

Wait For It...

At this point, you should see the main repository, with your pull request center screen.

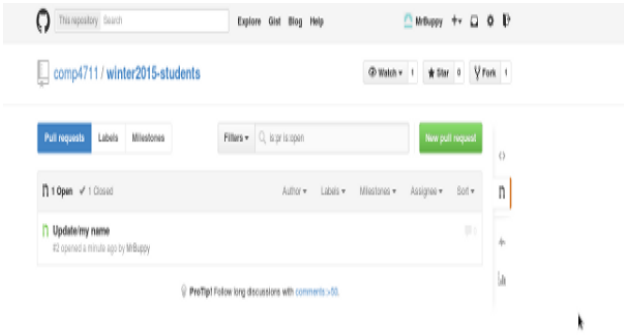
You can comment on it, but you will have to wait for the repo maintainer to merge or reject your pull request.



Waiting For It?

The maintainer is likely not waiting with baited breath for your specific change. You will probably work on something else, and then revisit the main repo site to see what is happening.

If your pull request is still in progress, it will be in the queue of requests.



Are We Done Yet?

Your pull request might get rejected for any number of reasons. You will get an email explaining the decision.

Once your pull request has been merged, resynchronize your "master" branch with the "upstream" repo, and you can delete the topic branch on your github repo and locally.

This probably sounds like a lot of work, but it will start to come naturally to you, and you will appreciate the wisdom of shared code management, and some of the steps here, once you have two or more developers sharing a repository!

Congratulations!

You have completed tutorial #tutorial02a: Working With a Shared Repository

If you would take a minute to provide some feedback, we would appreciate it!

The next activity in sequence is: tutorial02b Setup CodeIgniter

You can use your browser's back button to return to the page you were on before starting this activity, or you can jump directly to the course homepage, organizer, or reference page.