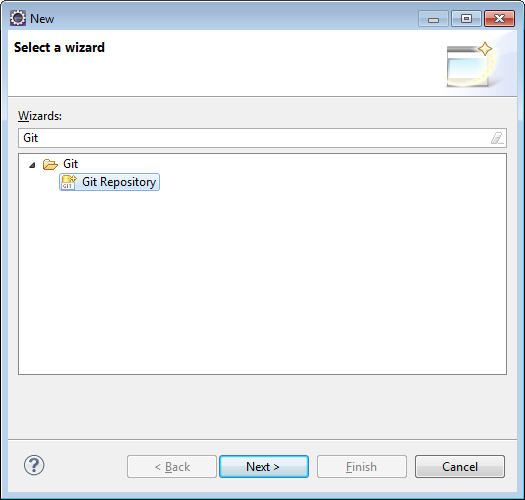
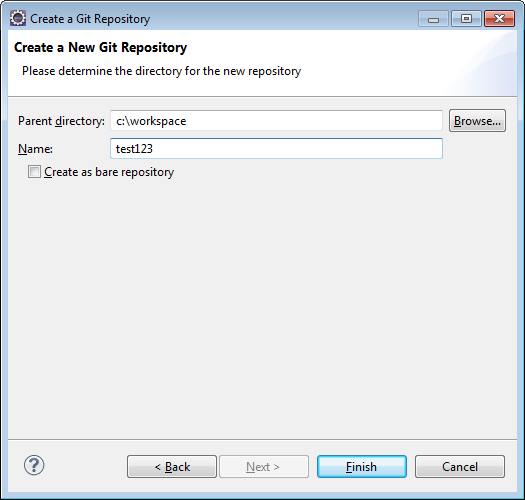
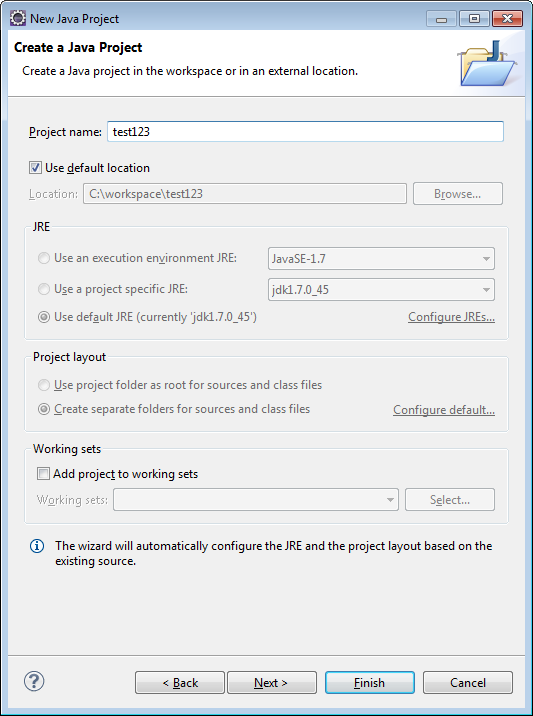
# EGit Tutorial

By Henry Chan [hchan@apache.og](mailto:hchan@apache.og)

1. Create a Git Repository with Eclipse



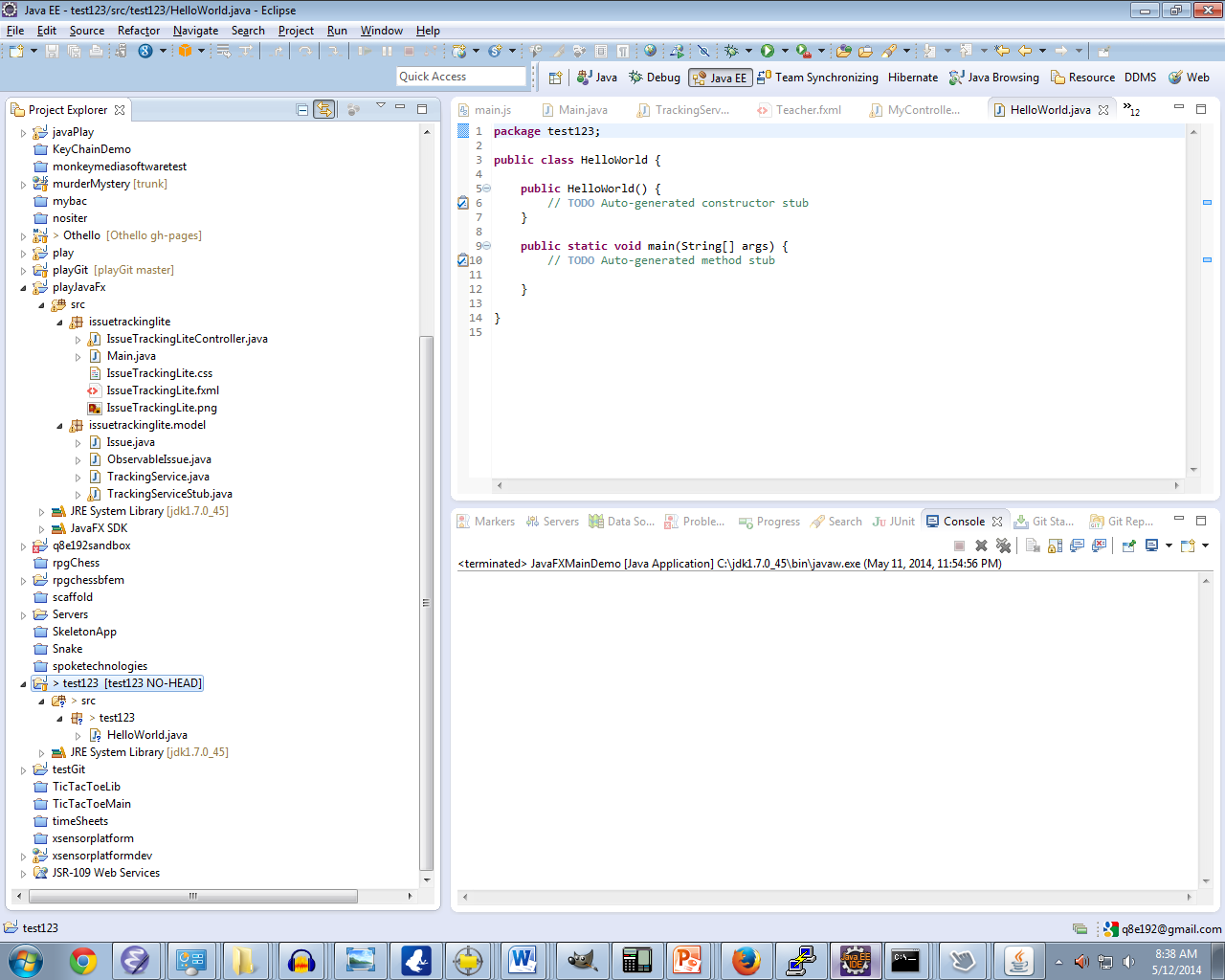
1. 
2. Create a New Java Project (Project Name should be the same as the name above), i.e. test123



1. Click Finish, and you should see:

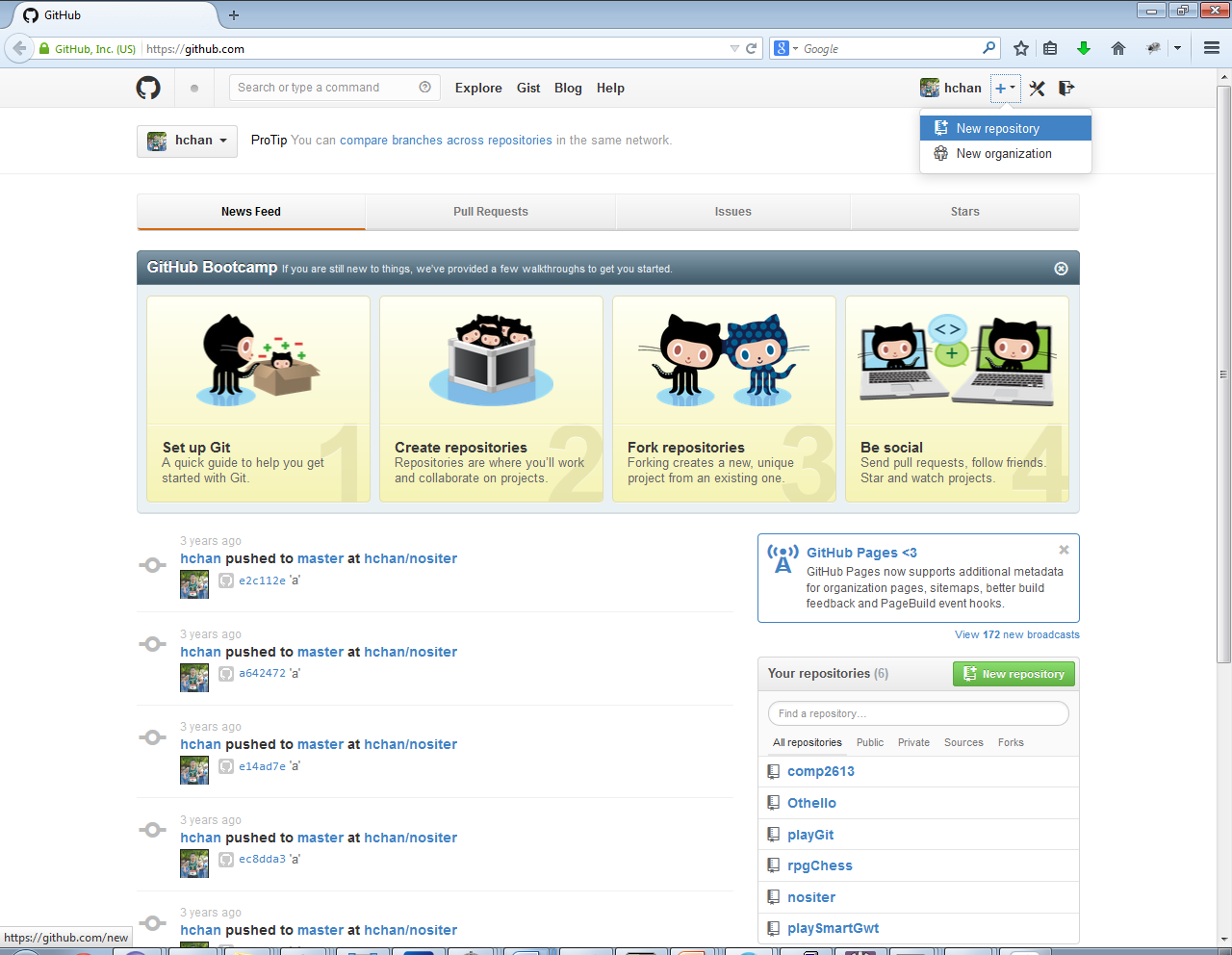
Your Project in the left window (Project Explorer) with a NO-HEAD in square brackets

1. Create a HelloWorld in that project

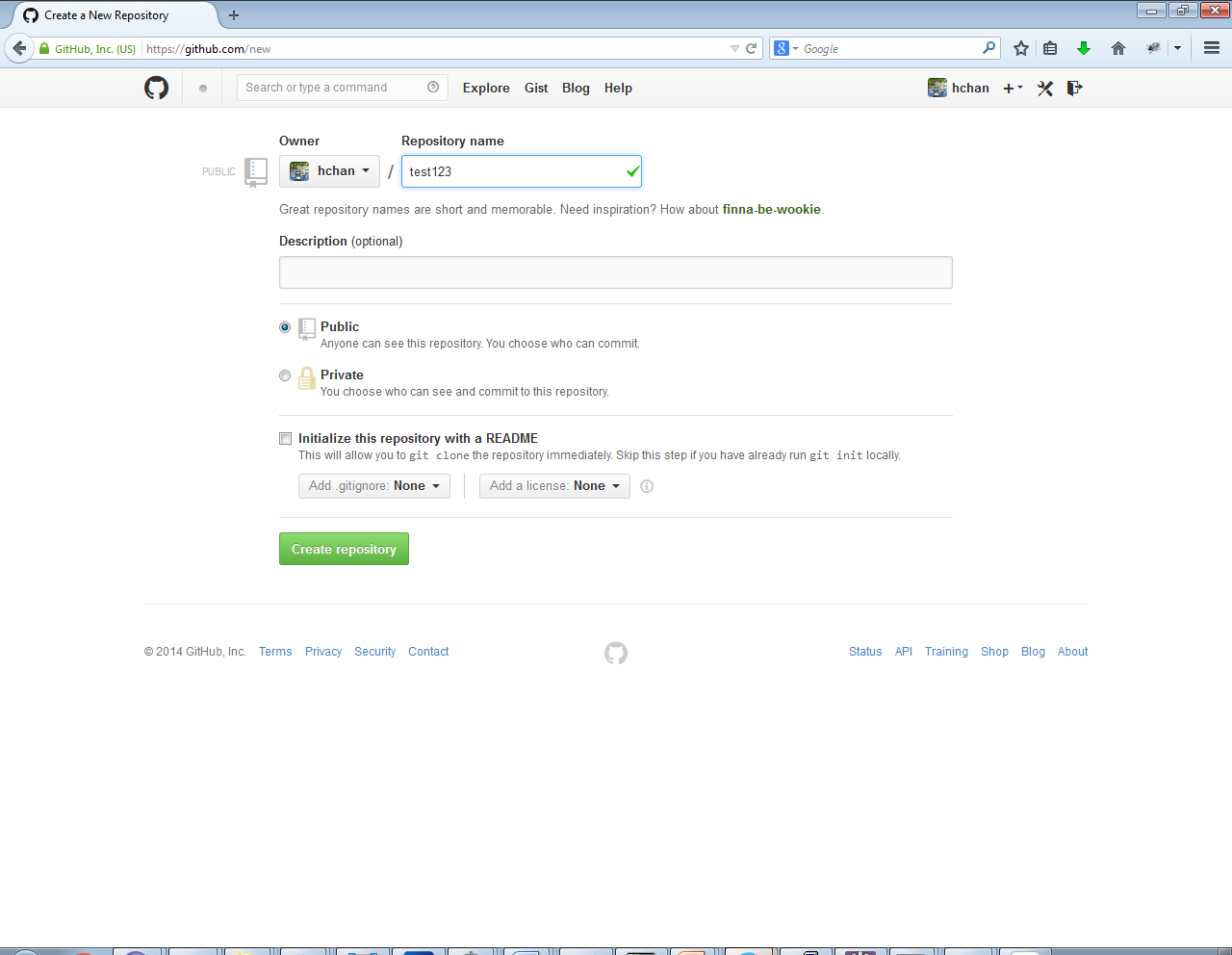


1. You are now ready to commit to GitHub

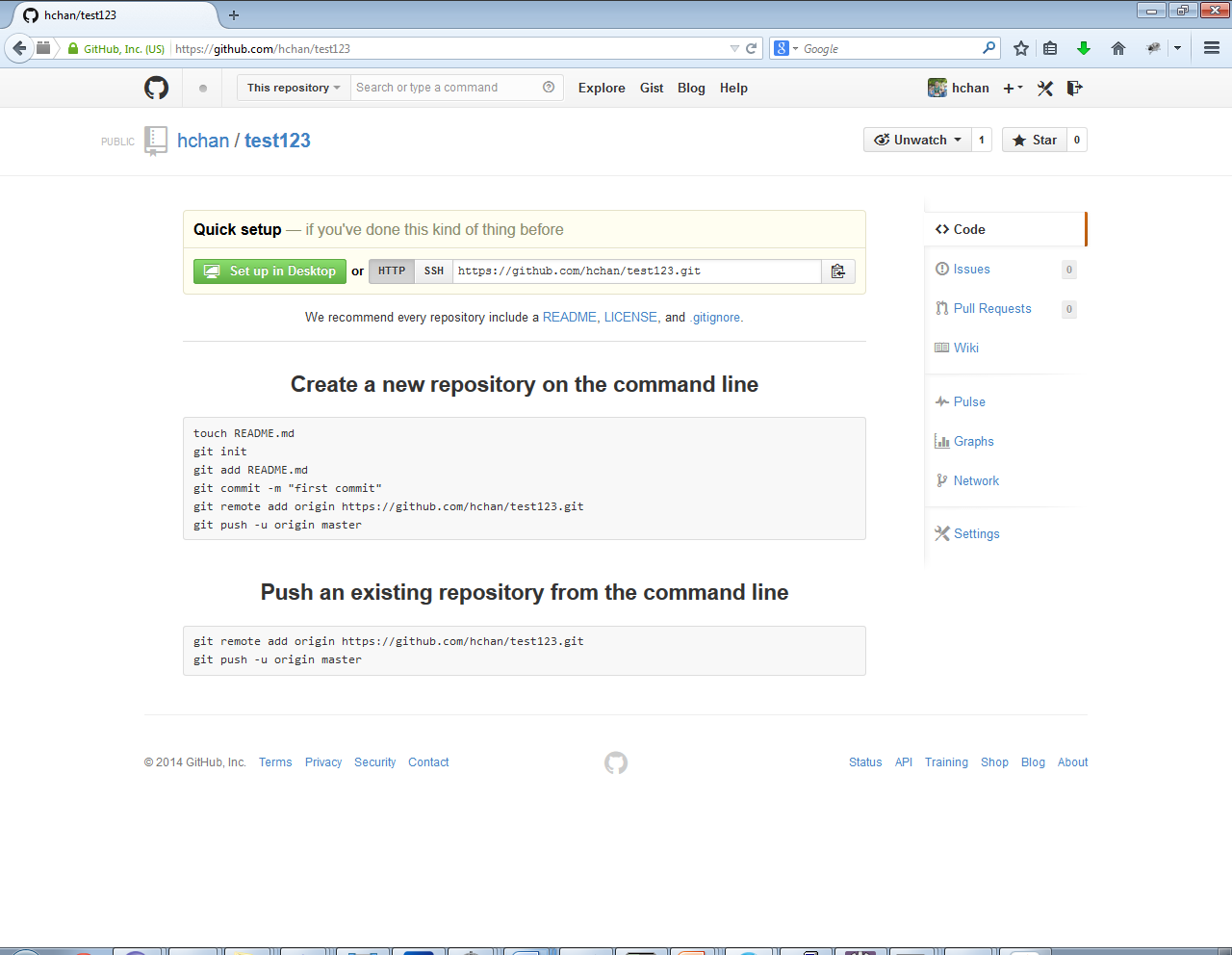
Open a Browser to <https://github.com/> Create an account if you do not already have one



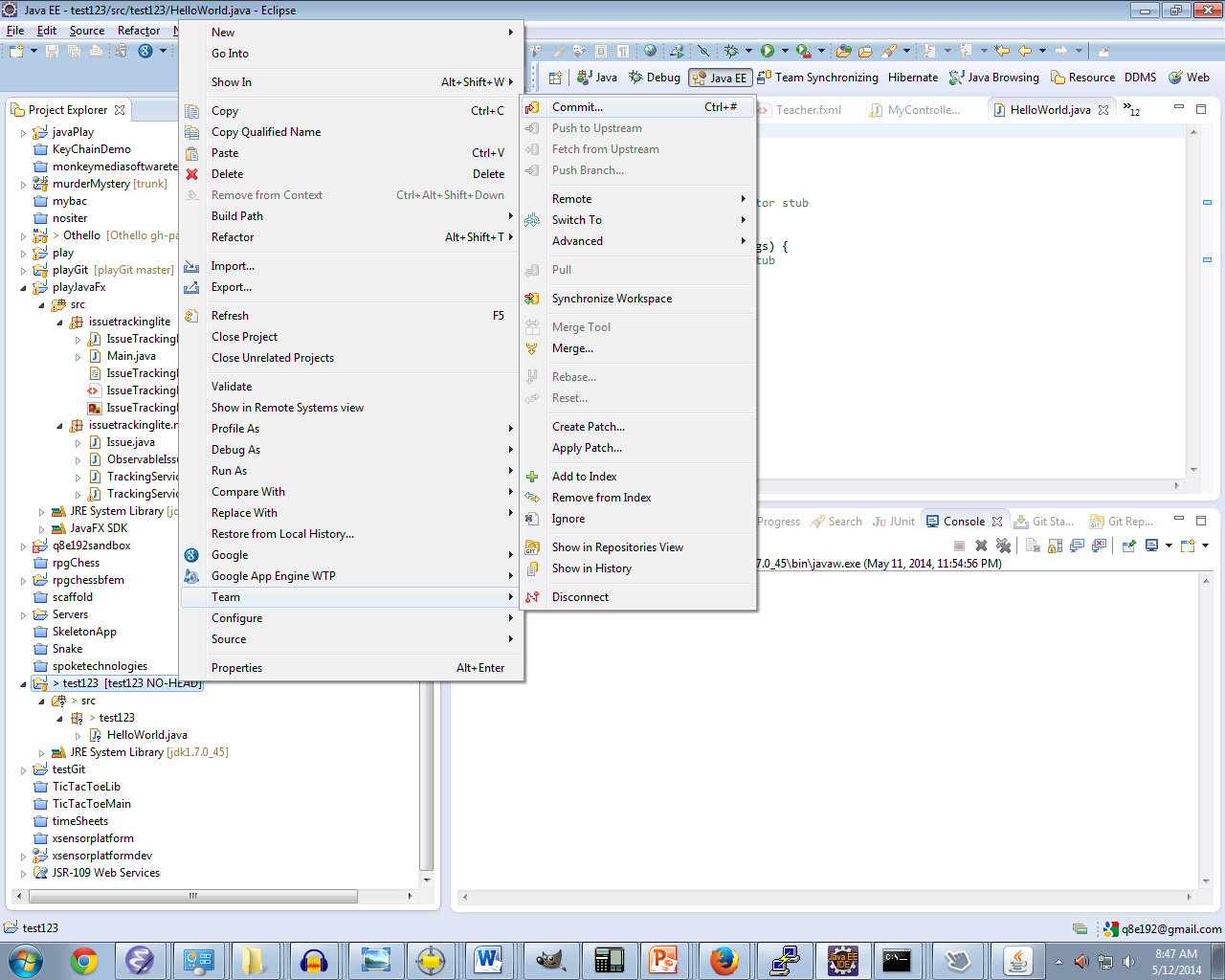
1. Use the same name as in step2 (i.e. test1234) for the repository name



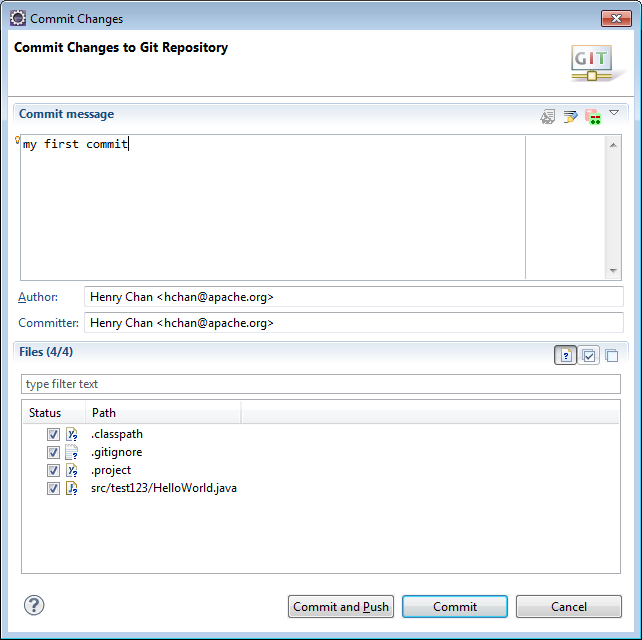
1. Click the Green Create Repository button



1. Take note of the url <https://github.com/hchan/test123.git> (copy and keep it in your buffer)
2. Go back to Eclipse, Right Click your Project->Team->Commit…

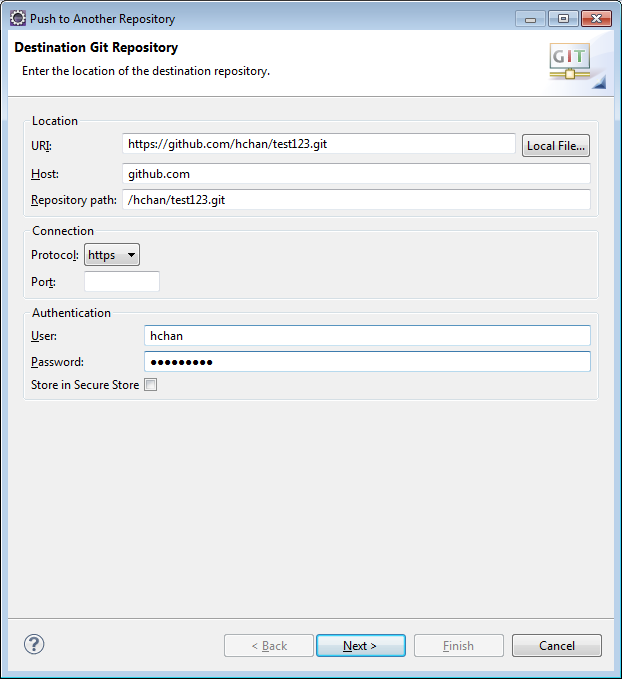


1. Write a comment, make sure all the files are checked



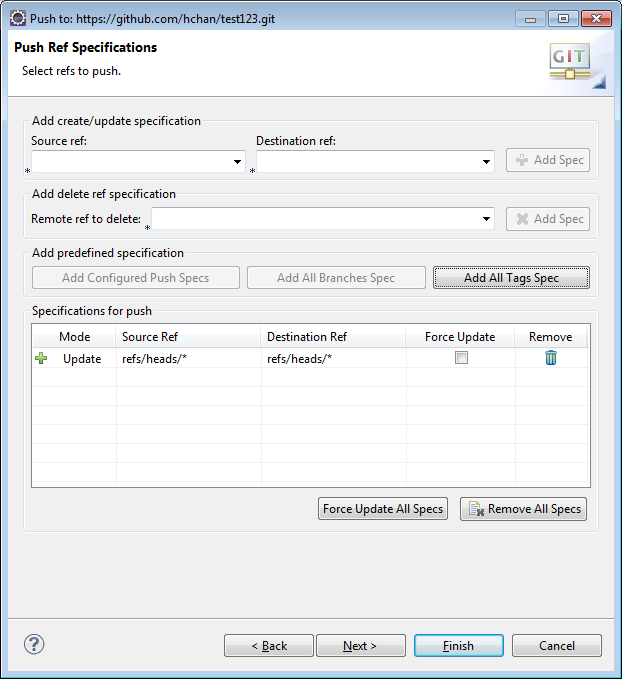
1. Click the Commit and Push button

Fill in your URI (Step 9), fill in your Git UserID/Password



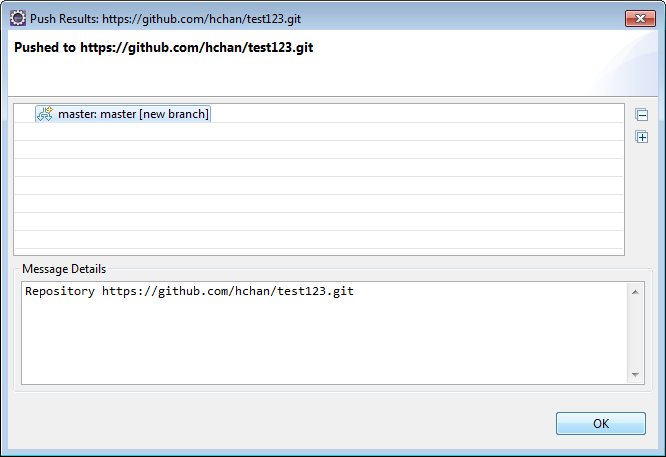
Then click Next

1. Click All Branches Spec

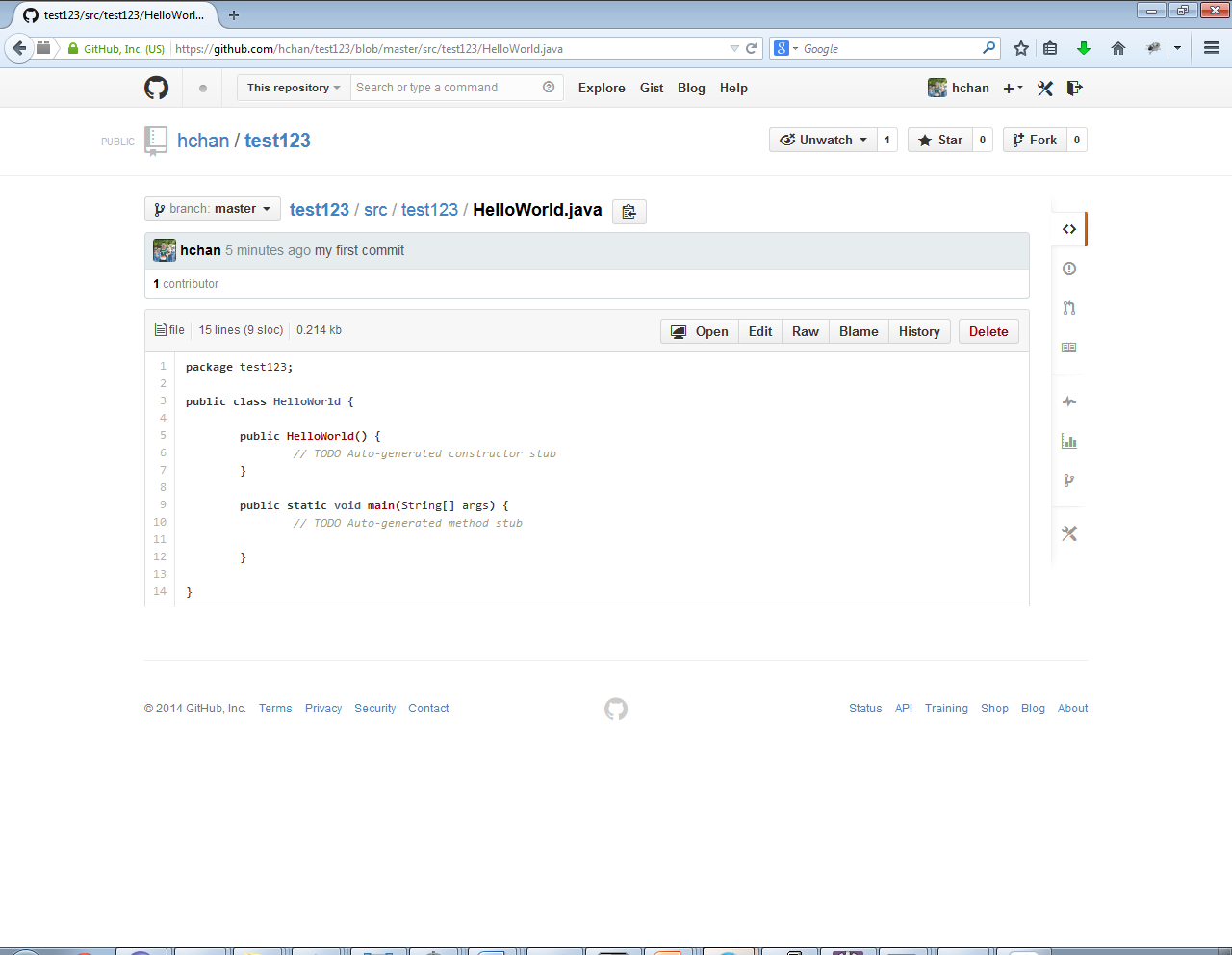


Then click Finish

1. Congrats if you got this far!

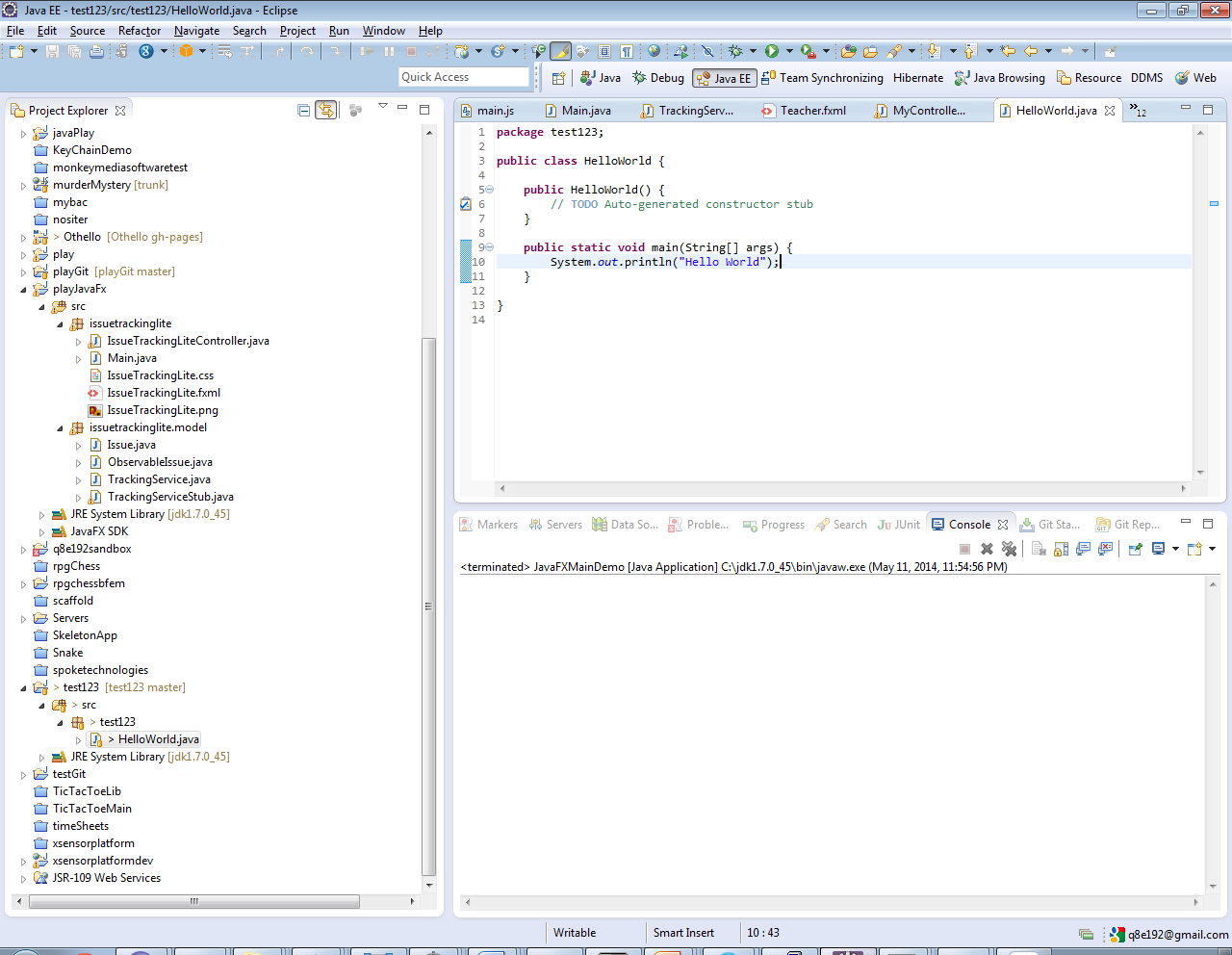


1. Got back to your browser and visit your new repository and check if your HelloWorld file is up there:

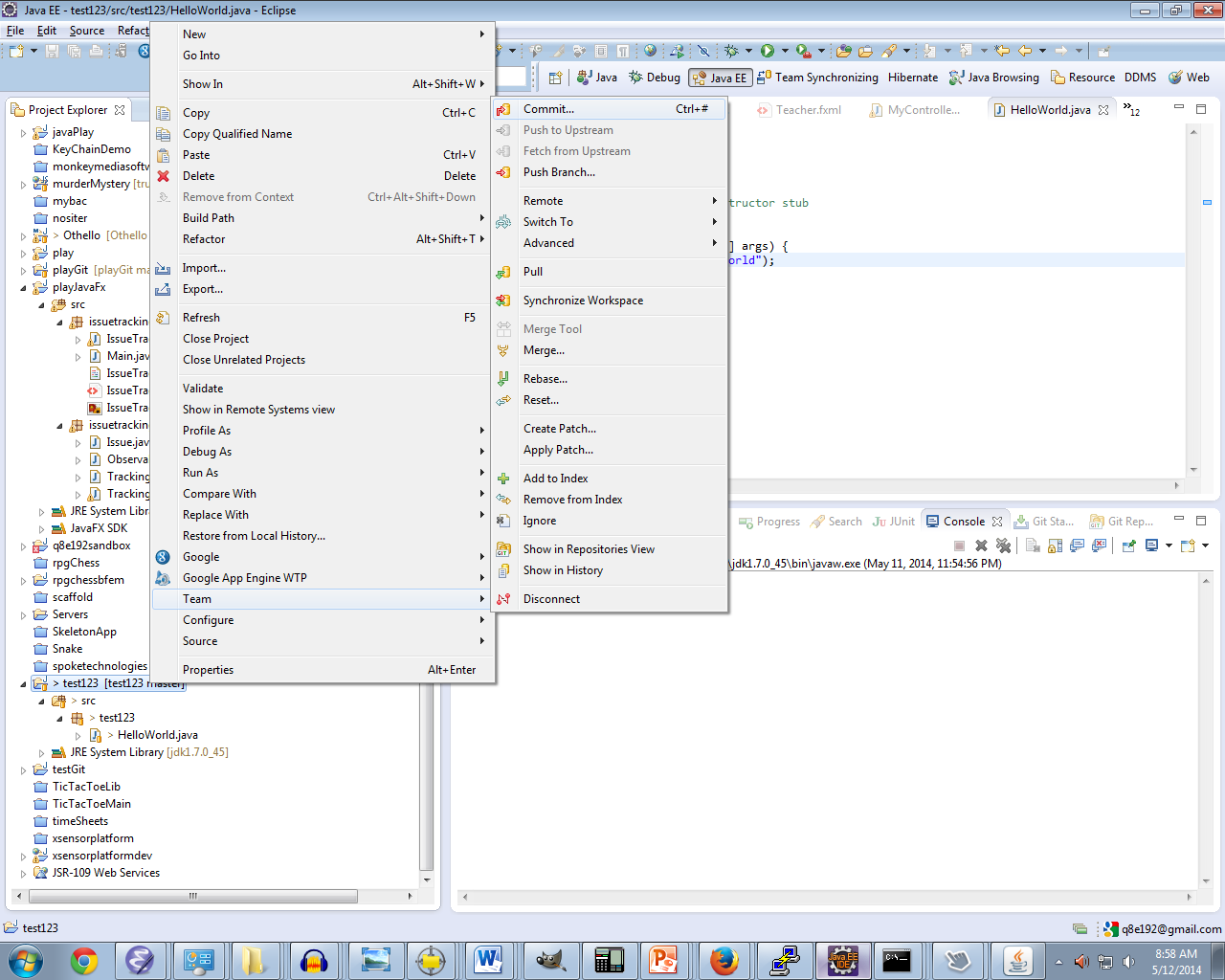


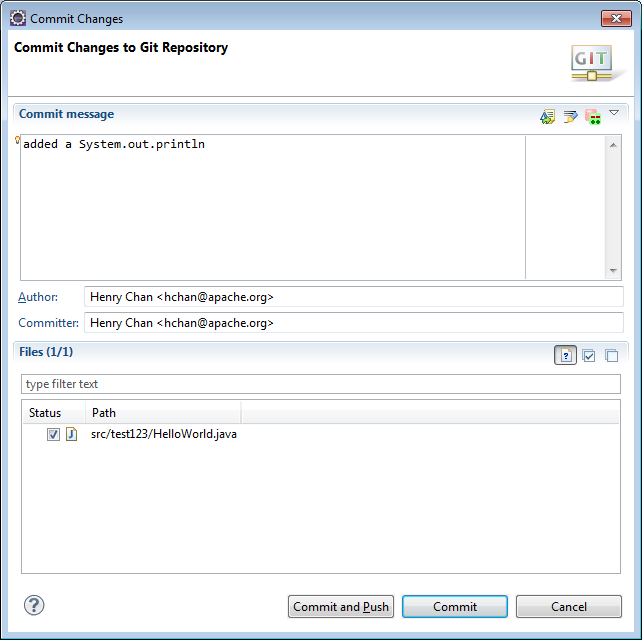
## Making Changes to your code…

1. Add a System.out.println to HelloWorld

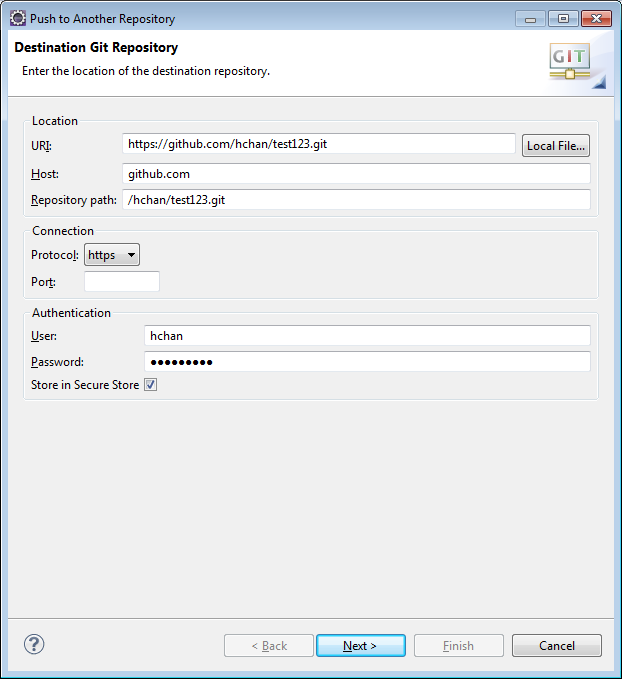


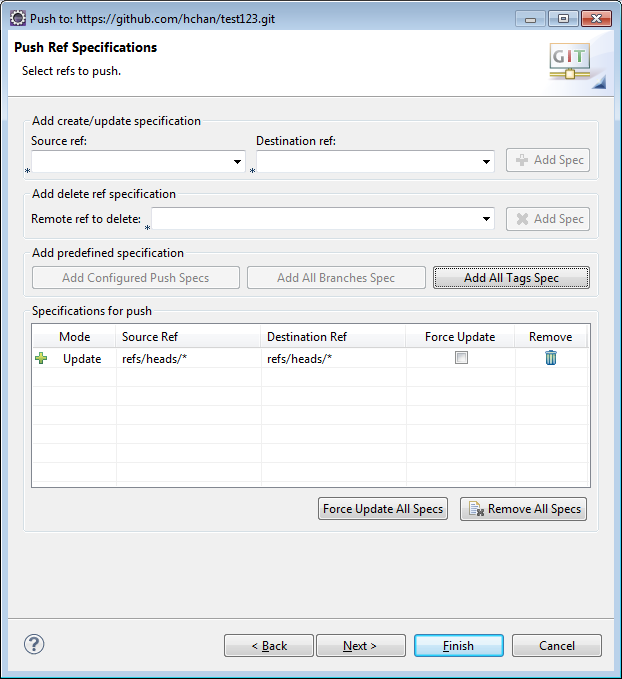
1. Commit your code



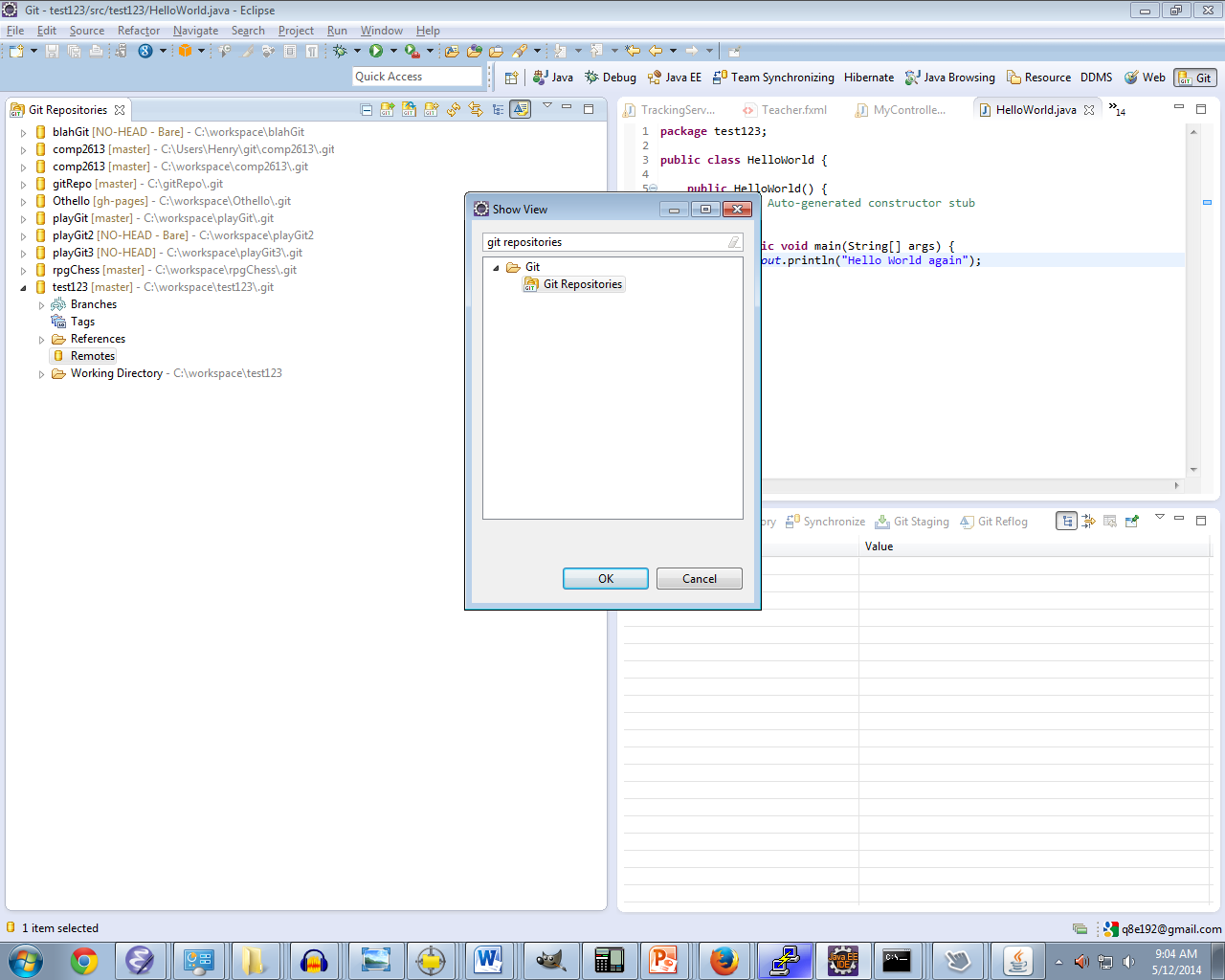
1. 

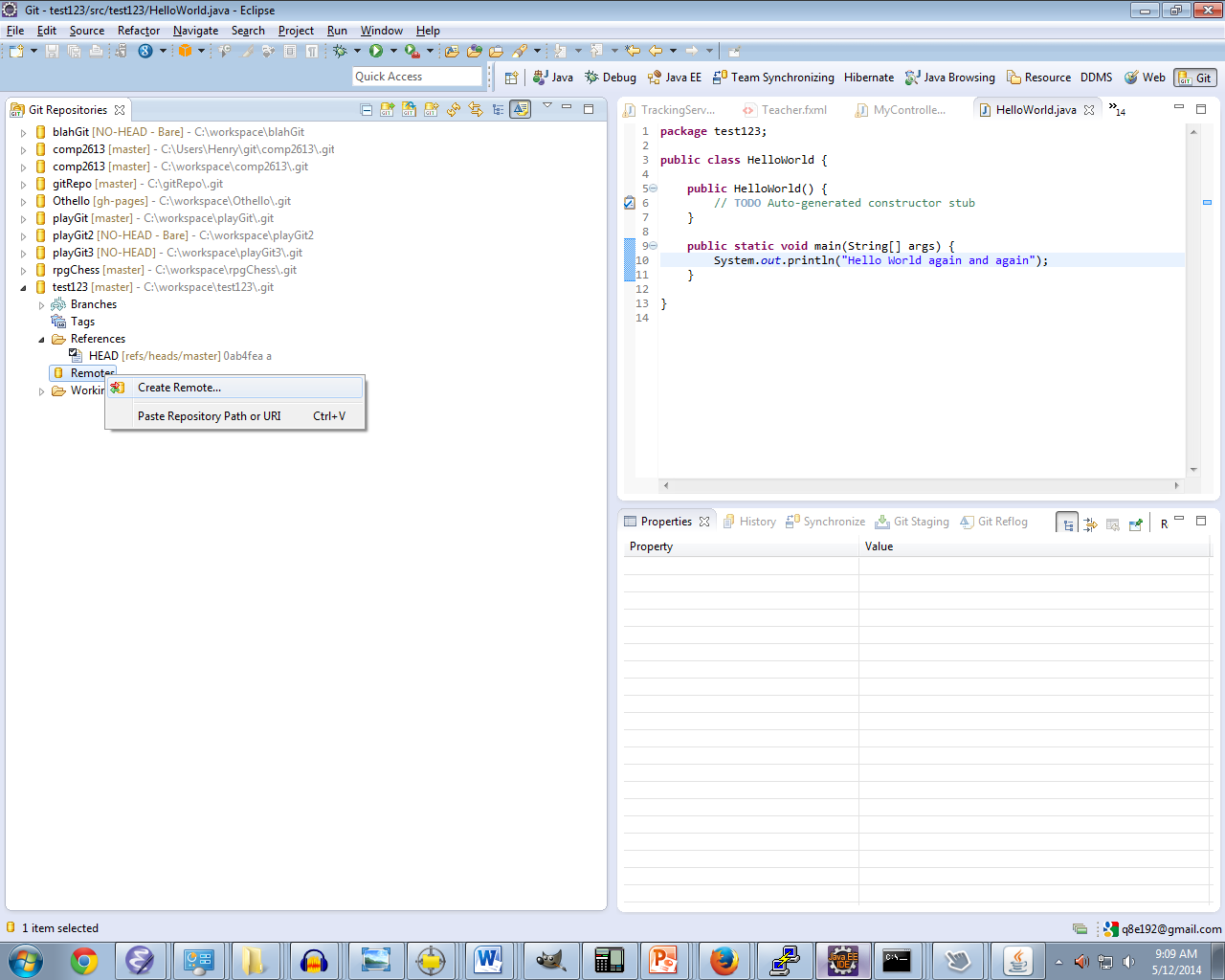
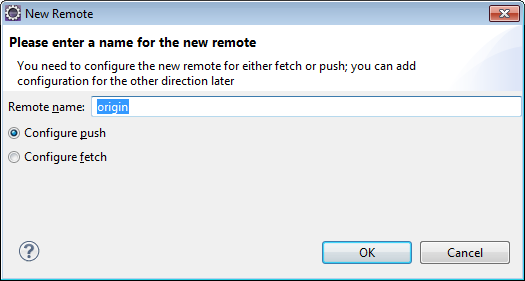
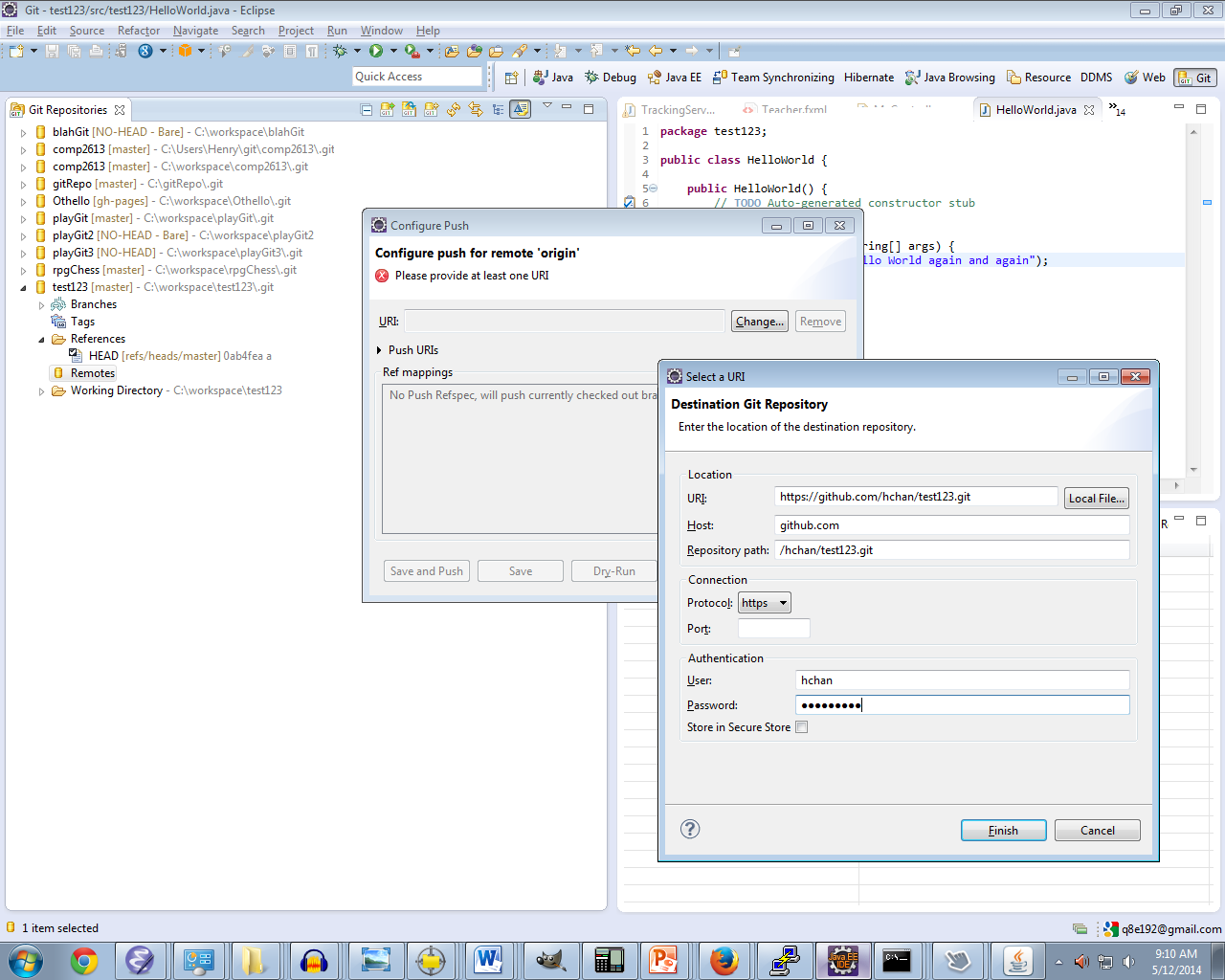
Commit and Push

1. 

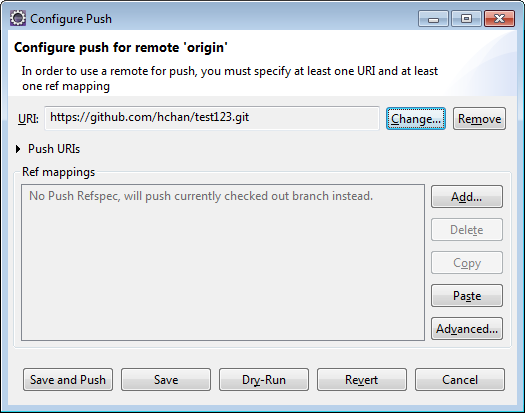


1. Check your commit on your browser via the Github.com site
2. Optional: as you can see, Step 4) is kinda repetitive… to save your remote URL,



1. 
2. 
3. 

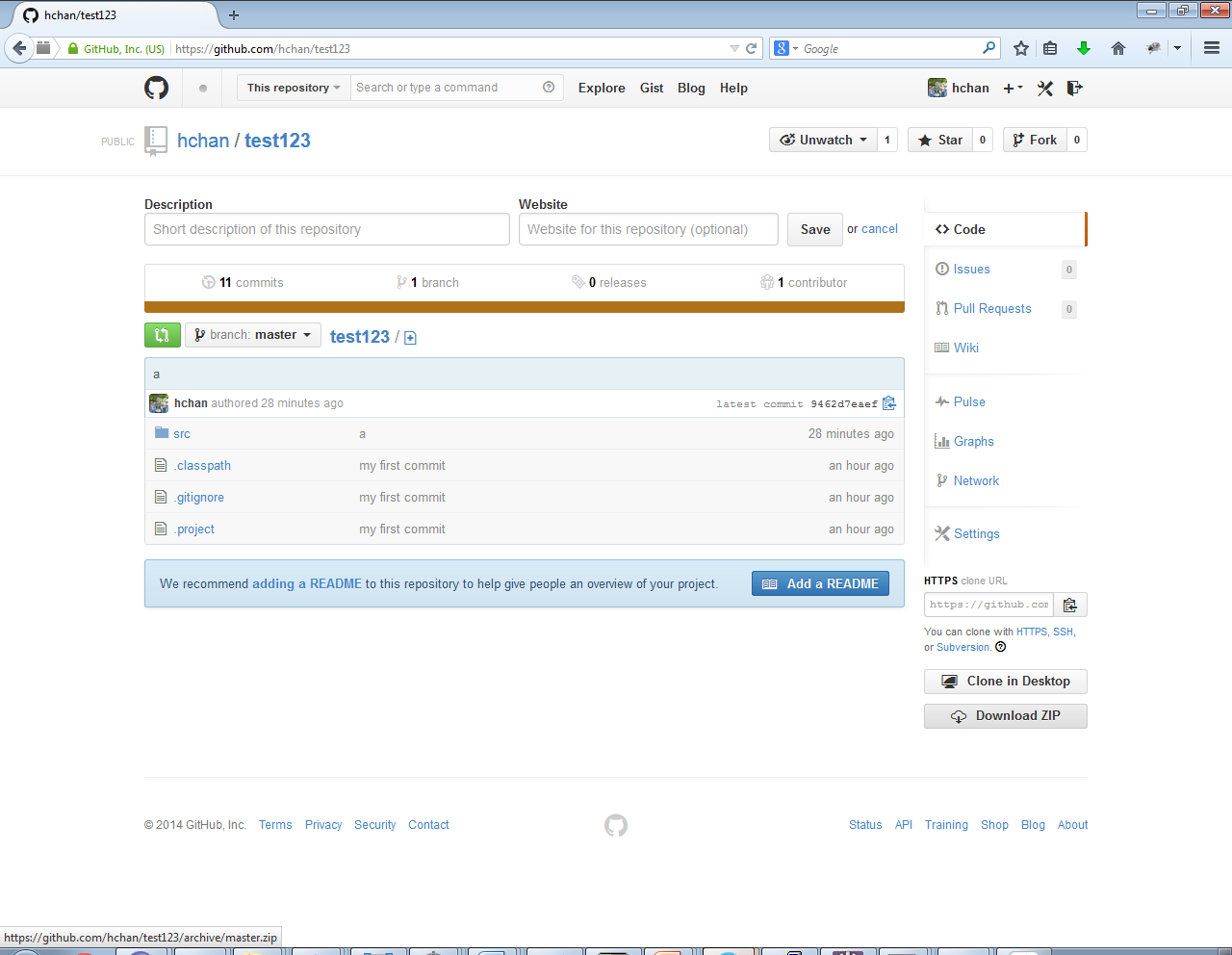
Click the Change… and then fill in the Destination Git Repository

1. 
2. Save and Push, and next / finish
3. ALWAYS check your changes are up by visiting github via your browser!!

## Downloading someone else’s Git Repository or working with Multiple Computers?

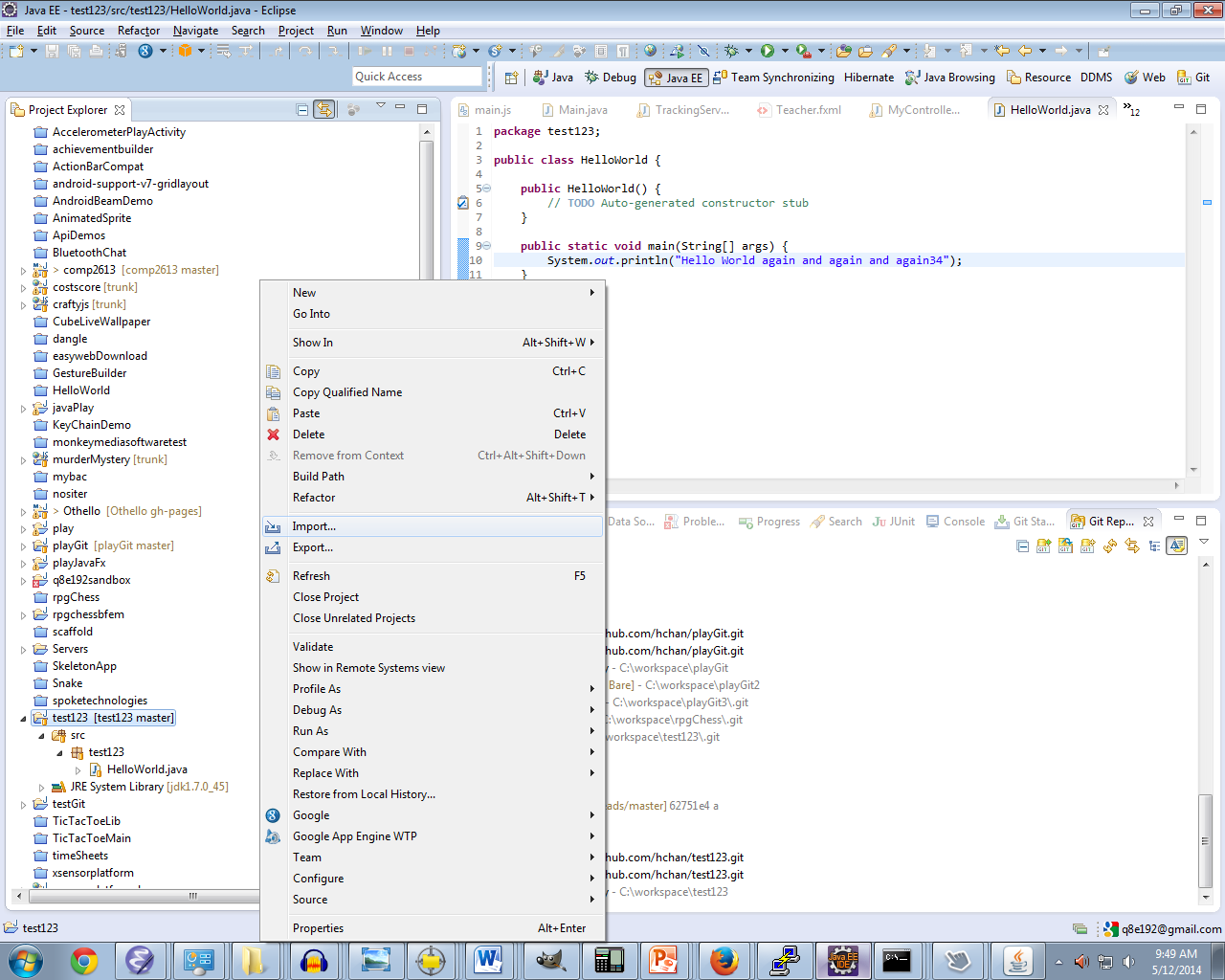
If you want to download someone’s Git Repository, there are a few ways. From the Browser, there is

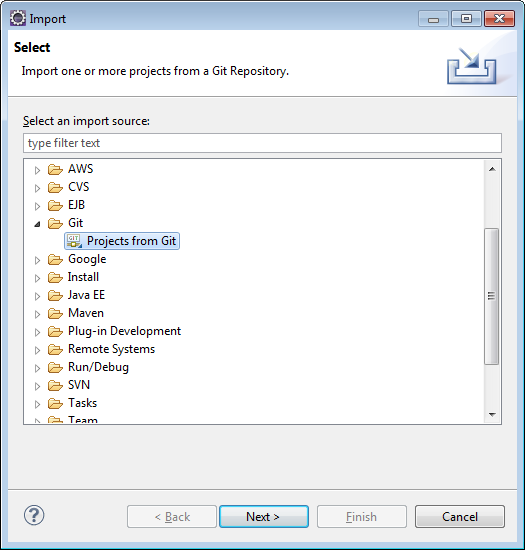
Download Zip option:

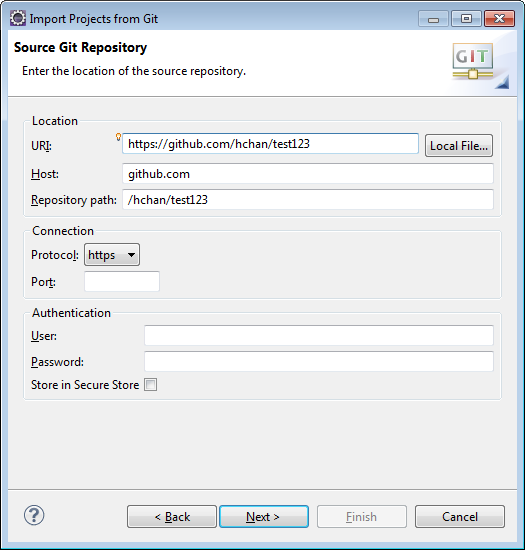
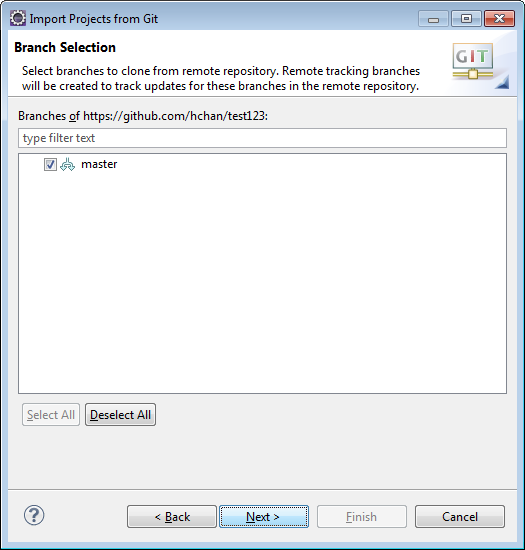
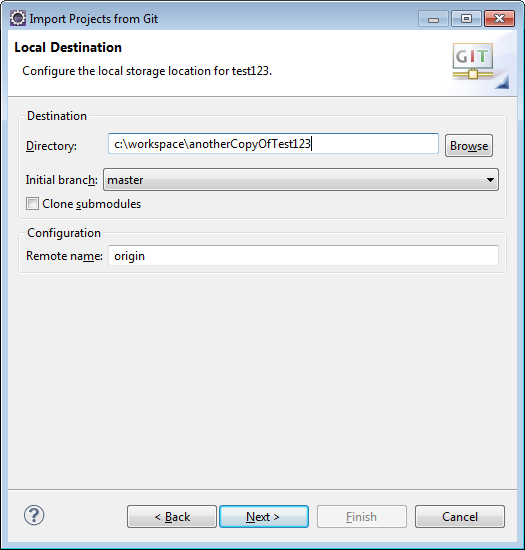
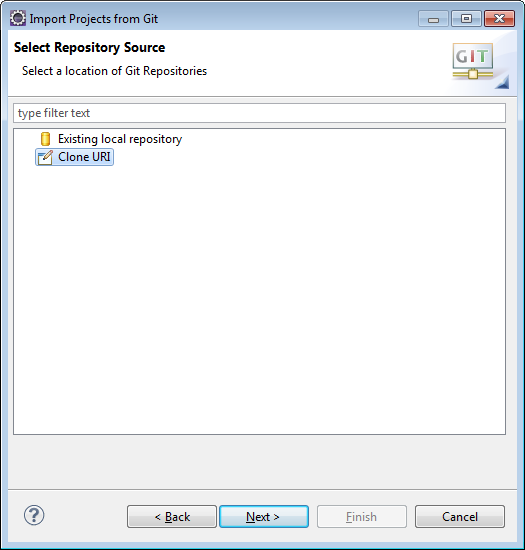


Or Clone in Desktop.

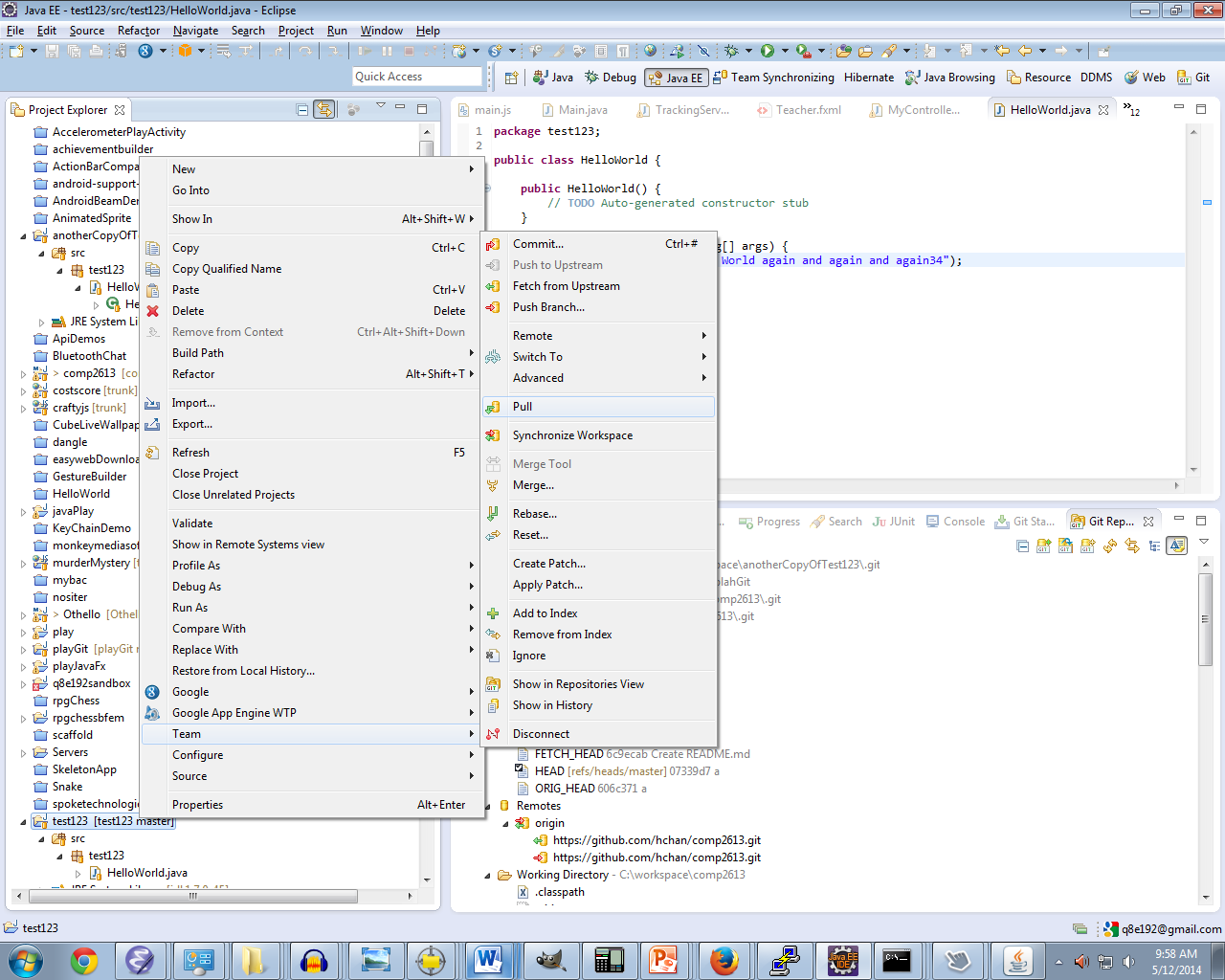
In the Egit (Eclipse Git) world…

1. Right Click Project Explore->Import…
2. 

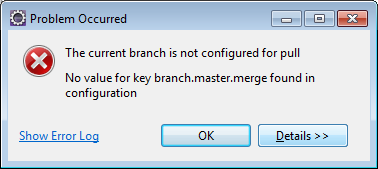


1. Click Clone URI. You’ll see the word “Clone” a lot in Git. Long story short, when dealing with Git, there is a Local and Remote Repository. You Clone locally, but Commit/Push to Remote. Note that you won’t need a username/password for GitHub cloning… Downloads are public, Commits are password protected (at least in the default way that GitHub is setup)
2. 
3. 
4. 
5. Next, and Finish

Final notes and words of wisdom … if you have 2 computers that use the same remote git repositoriy, you should “sync” up often. That’s what, Git Pull is about:



If you get an error like:



There are 2 ways to fix this 1) (Preferred), kill your Project in Eclipse and re-import everything

Or 2)

Google: The current branch is not configured for pull

from the Git Repositories tab:

* right click on your local repository
* select **Properties**
* press **New Entry...**
* enter the following two keys:

(1)

Key = branch.master.remote

Value = origin

(2)

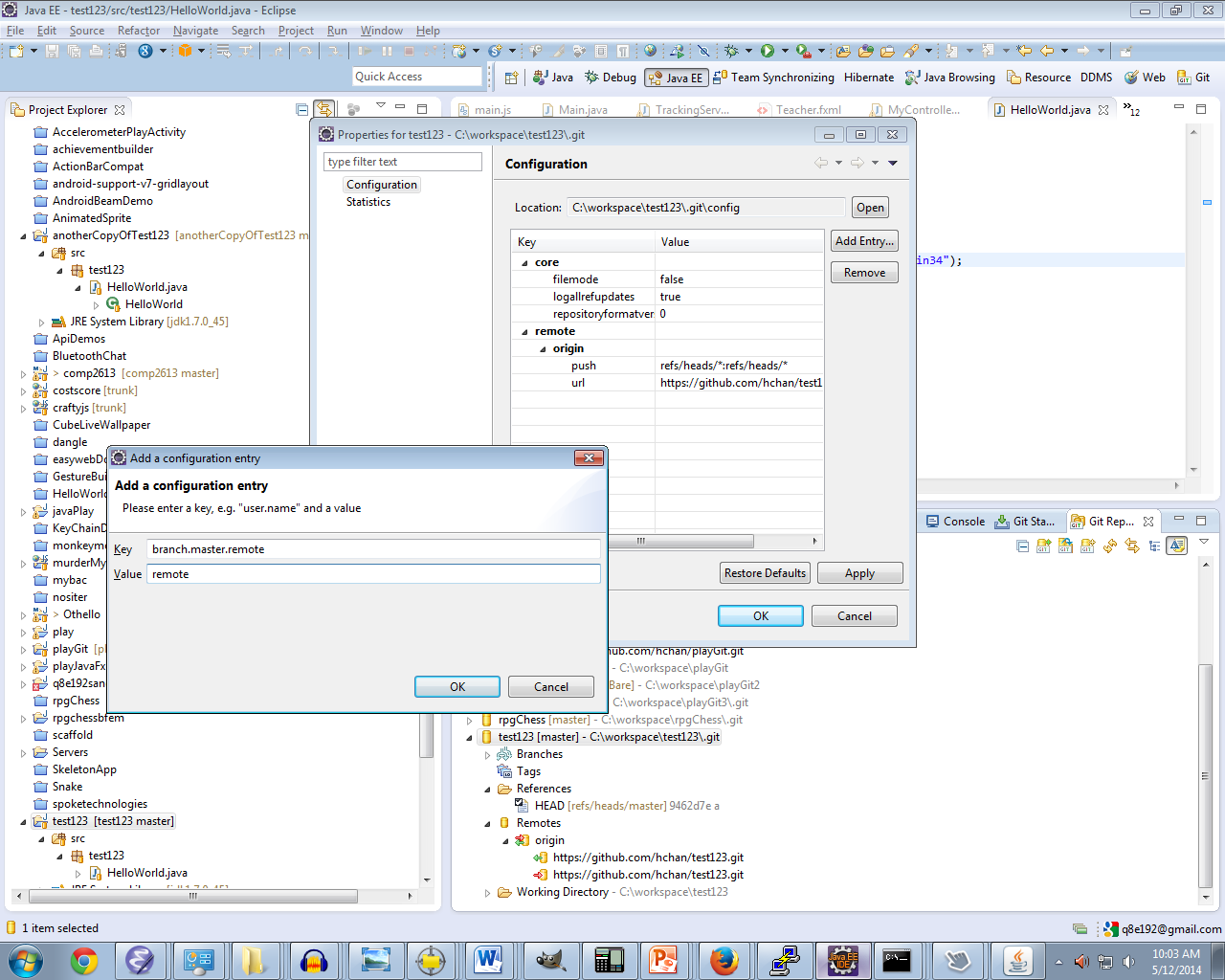
Key = branch.master.merge

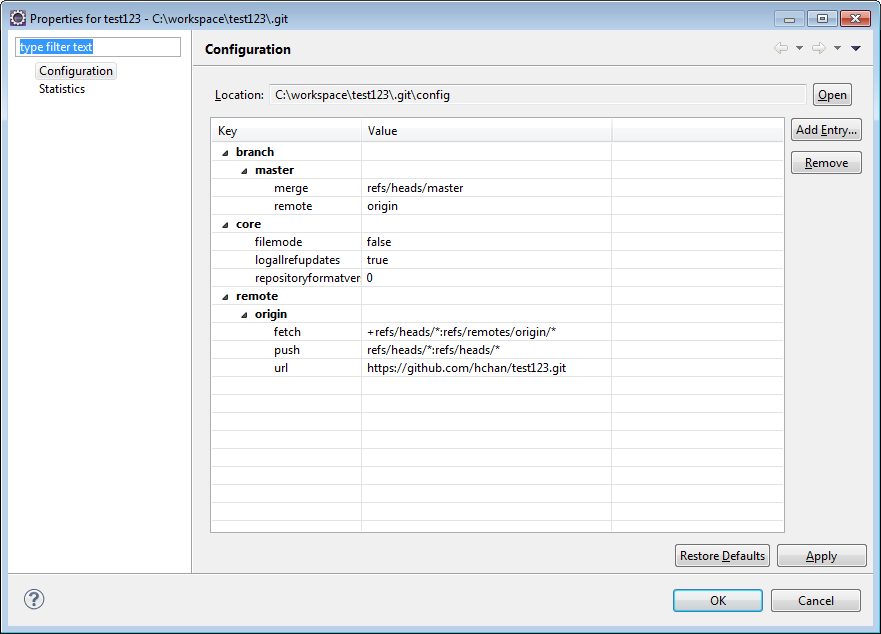
Value = refs/heads/master

3

Key = remote.origin.fetch

Value = +refs/heads/\*:refs/remotes/origin/\*





## Resolving Conflicts (Advanced)

High level: after you go a Git Pull, you get “Red” conflicts by certain files.

1. Fix those files to your heart’s content
2. Use Git Staging (Window->Show View->Other…->Git Staging
3. Click and DRAG your changes from Unstaged Changes to Staged Changes
4. Commit and Push