PHAS3459
Scientific programming using object-oriented languages

Module 3: Syncing Repositories with GitHub

Ben Waugh & Simon Jolly

Distributed Version Control Systems

- Hopefully most of you will now be up to speed using Git to commit your work locally.
- Quite a few of you have come up against the problem of trying to do work on multiple machines:
  - How do you synchronise work between machines?
  - Doesn't adding the extra complication of using Git make this even harder?
- This is actually exactly what Git is designed for.
- Git is a **Distributed Version Control System**:
  - Developed by Linus Torvalds for writing the Linux kernel.
  - Designed to allow multiple people to work on the same files.
  - Allows repository to be mirrored in multiple locations.

Enter GitHub

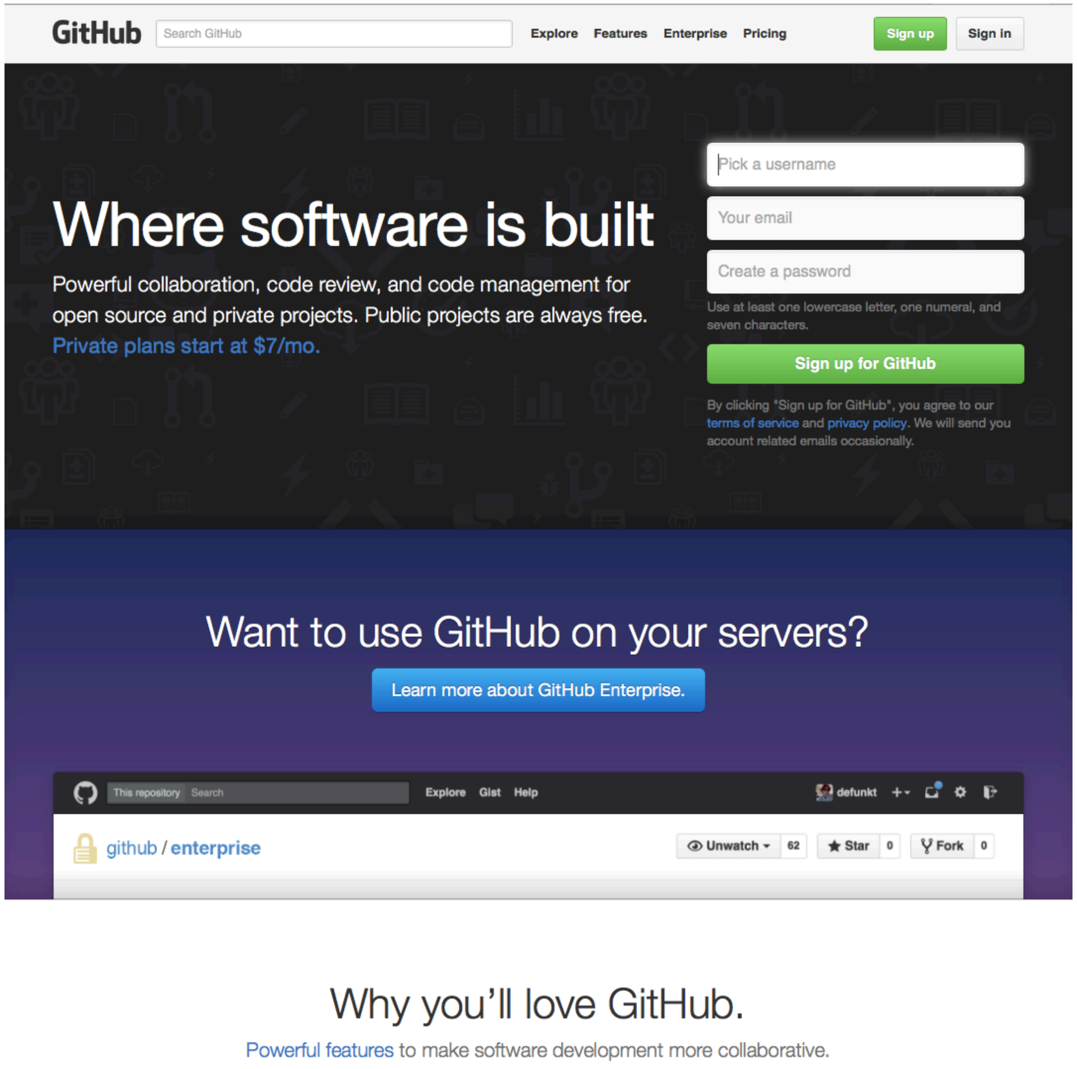
- In order to allow you to access your Git repository from multiple machines, you need somewhere to store your **central** git repository:
  – Acts as the reference from which all other copies of your repository synchronise changes.
  – Allows you to copy the repository (**clone**) to a new machine without setting up everything from scratch.
- This is normally a server that everybody working on the project has access to.
- We will be using Github:
  – Popular online resource for Git repositories(!)
  – Provides free accounts for anyone with academic email(!!)
  – Allows academic accounts to create private repositories(!!!)

Some More Git Terminology

- **Local.** The copy of your git repository on your local machine.
- **Remote.** The central git repository that you use to keep all your local repositories synchronised.
- **Clone.** The act of copying an existing repository to create a new one, complete with your entire Git history.
- **Merge.** Bringing together two repositories that differ. A simple merge simply aggregates all the files within them. More complex merging is required if individual files need to be compared and merged (see below).
- **Pull.** Bringing down all of the new commits and changes from your remote repository and merging them with your local repository. This may be as simple as fast-forwarding your repository to the latest commit or may require more sophisticated merging if you have made conflicting commits to both repositories (which is best avoided…).
- **Push.** The opposite of a pull. Once you have committed your changes to your local repository, you "push" them up to your remote. You can now pull those changes down to your other machines, or allow other users to see your changes.

Setting Up Your GitHub Account

- If you don't already have a GitHub account, you can sign up from the GitHub homepage: https://github.com
- Use your academic "@ucl.ac.uk" email address when signing up (this is important).
- You will be sent an email with a verification link.



GitHub Search GitHub Explore Features Enterprise Pricing Sign up Sign in

Where software is built

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free. Private plans start at $7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.

Want to use GitHub on your servers?

Learn more about GitHub Enterprise.

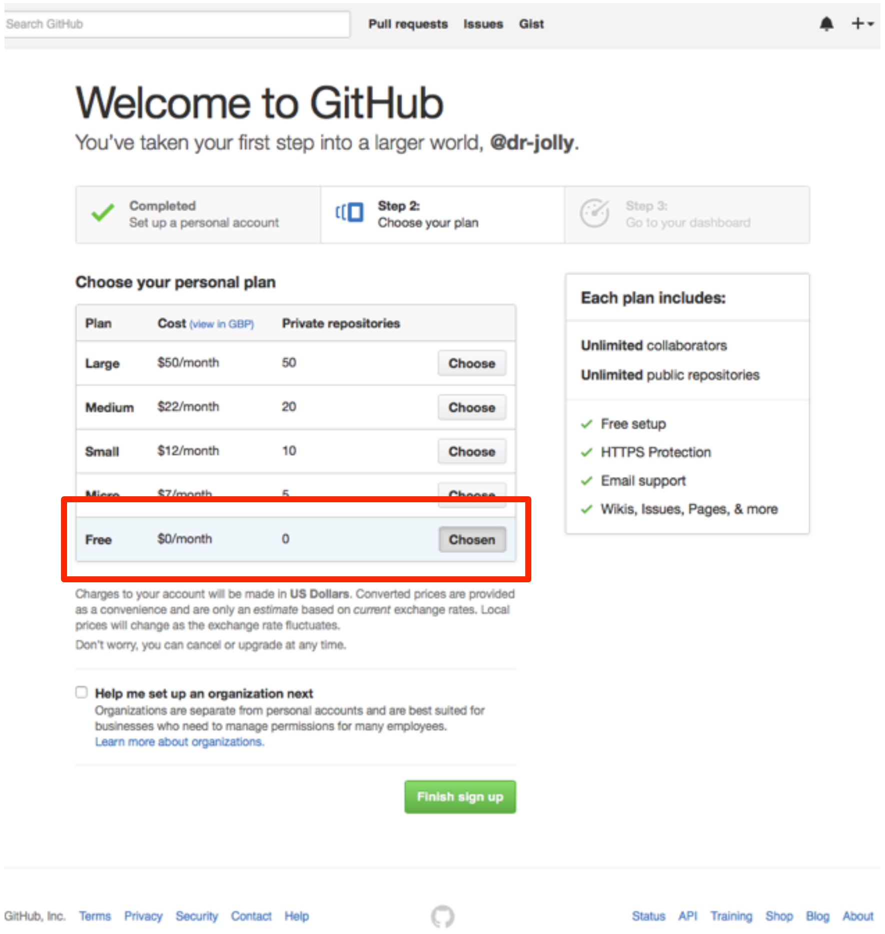
github / enterprise

Why you'll love GitHub.

Powerful features to make software development more collaborative.

Account Setup

Choose a free account:



Welcome to GitHub

You've taken your first step into a larger world, **@dr-jolly**.

Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost (view in GBP)	Private repositories	
Large	$50/month	50	Choose
Medium	$22/month	20	Choose
Small	$12/month	10	Choose
Micro	$7/month	5	Choose
Free	$0/month	0	Chosen

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.

Don't worry, you can cancel or upgrade at any time.

Each plan includes:

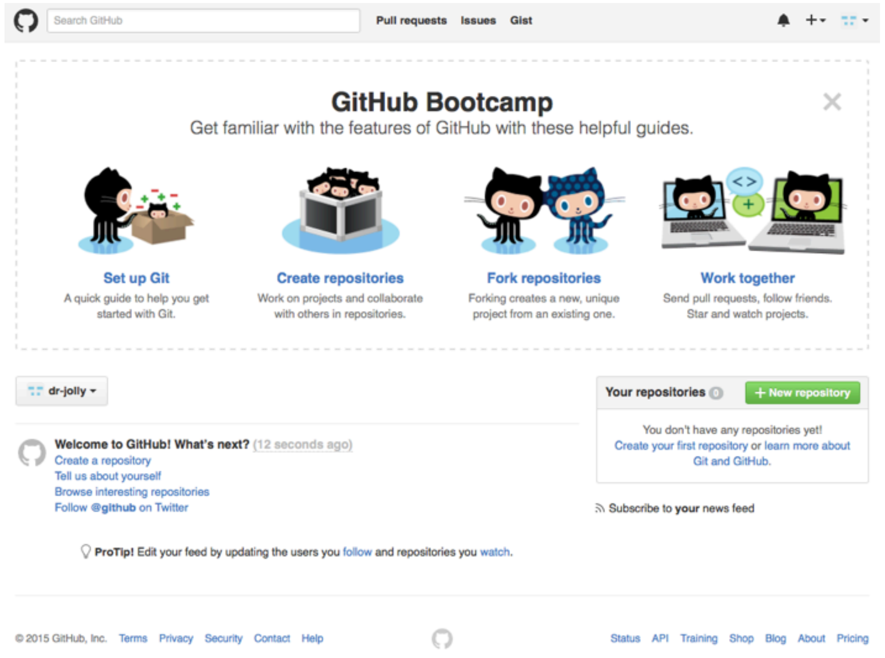
Unlimited collaborators
Unlimited public repositories

- ✓ Free setup
- ✓ HTTPS Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

Help me set up an organization next
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations.

Finish sign up
GitHub, Inc. Terms Privacy Security Contact Help
Status API Training Shop Blog About

Your Github homepage:



GitHub Bootcamp

Get familiar with the features of GitHub with these helpful guides.

Set up Git — A quick guide to help you get started with Git.

Create repositories — Work on projects and collaborate with others in repositories.

Fork repositories — Forking creates a new, unique project from an existing one.

Work together — Send pull requests, follow friends. Star and watch projects.

dr-jolly

Welcome to GitHub! What's next? (12 seconds ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow **@github** on Twitter

Your repositories 0 + New repository

You don't have any repositories yet!
Create your first repository or learn more about Git and GitHub.

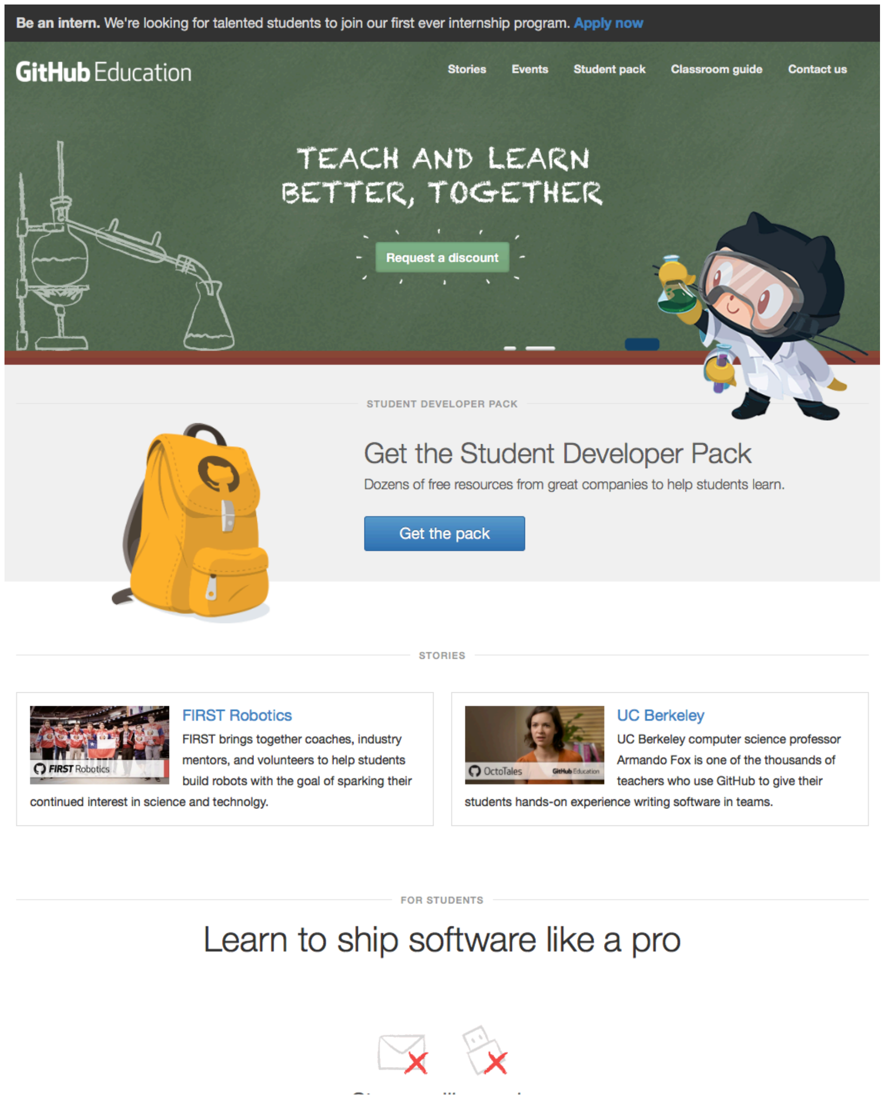
Subscribe to **your** news feed

ProTip! Edit your feed by updating the users you follow and repositories you watch.

© 2015 GitHub, Inc. Terms Privacy Security Contact Help
Status API Training Shop Blog About Pricing

Registering For An Academic Account

- In order to create free private repositories, you need to register for an academic account.
- Go to https://education.github.com
- You need to have clicked the link in the GitHub verification email before you can access the registration page.



Be an intern. We're looking for talented students to join our first ever internship program. Apply now

GitHub Education

Stories Events Student pack Classroom guide Contact us

TEACH AND LEARN BETTER, TOGETHER

Request a discount

STUDENT DEVELOPER PACK

Get the Student Developer Pack

Dozens of free resources from great companies to help students learn.

Get the pack

STORIES

FIRST Robotics

FIRST brings together coaches, industry mentors, and volunteers to help students build robots with the goal of sparking their continued interest in science and technolgy.

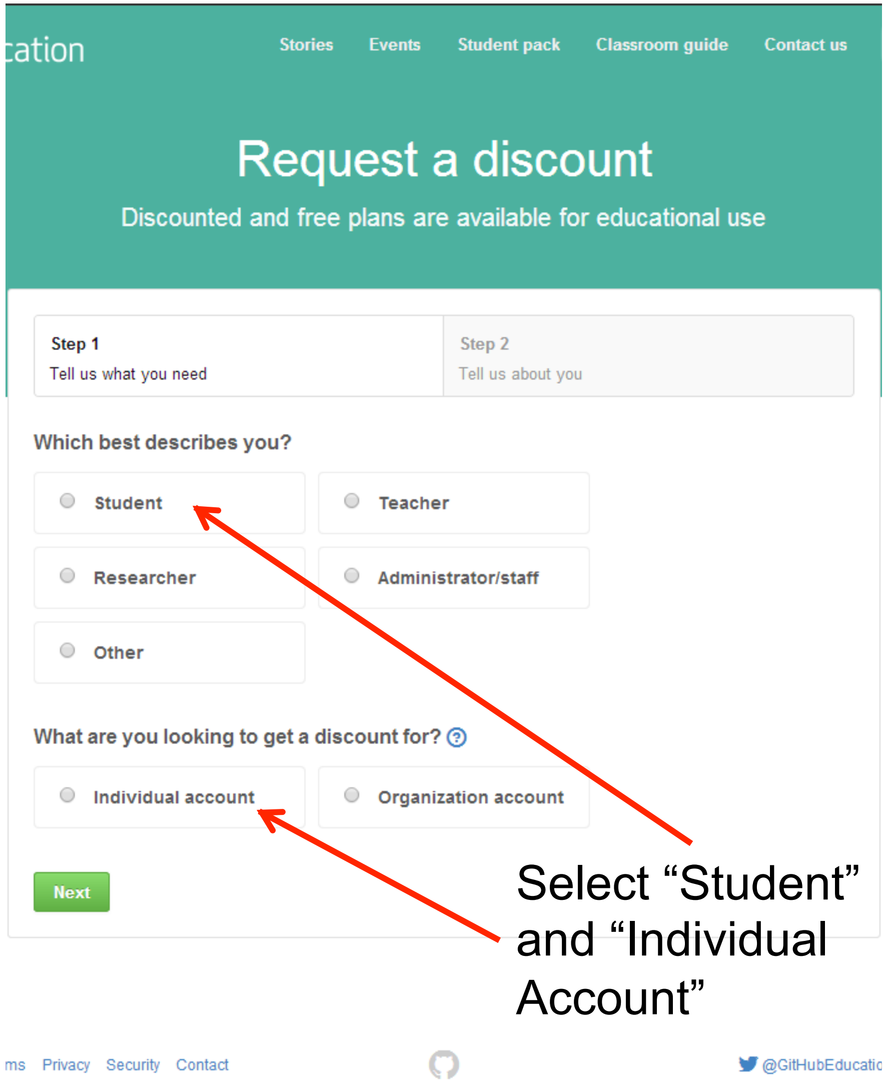
UC Berkeley

UC Berkeley computer science professor Armando Fox is one of the thousands of teachers who use GitHub to give their students hands-on experience writing software in teams.

FOR STUDENTS

Learn to ship software like a pro

Register Your Academic Details



cation Stories Events Student pack Classroom guide Contact us

Request a discount

Discounted and free plans are available for educational use

Step 1 Tell us what you need | Step 2 Tell us about you

Which best describes you?

- Student
- Teacher
- Researcher
- Administrator/staff
- Other

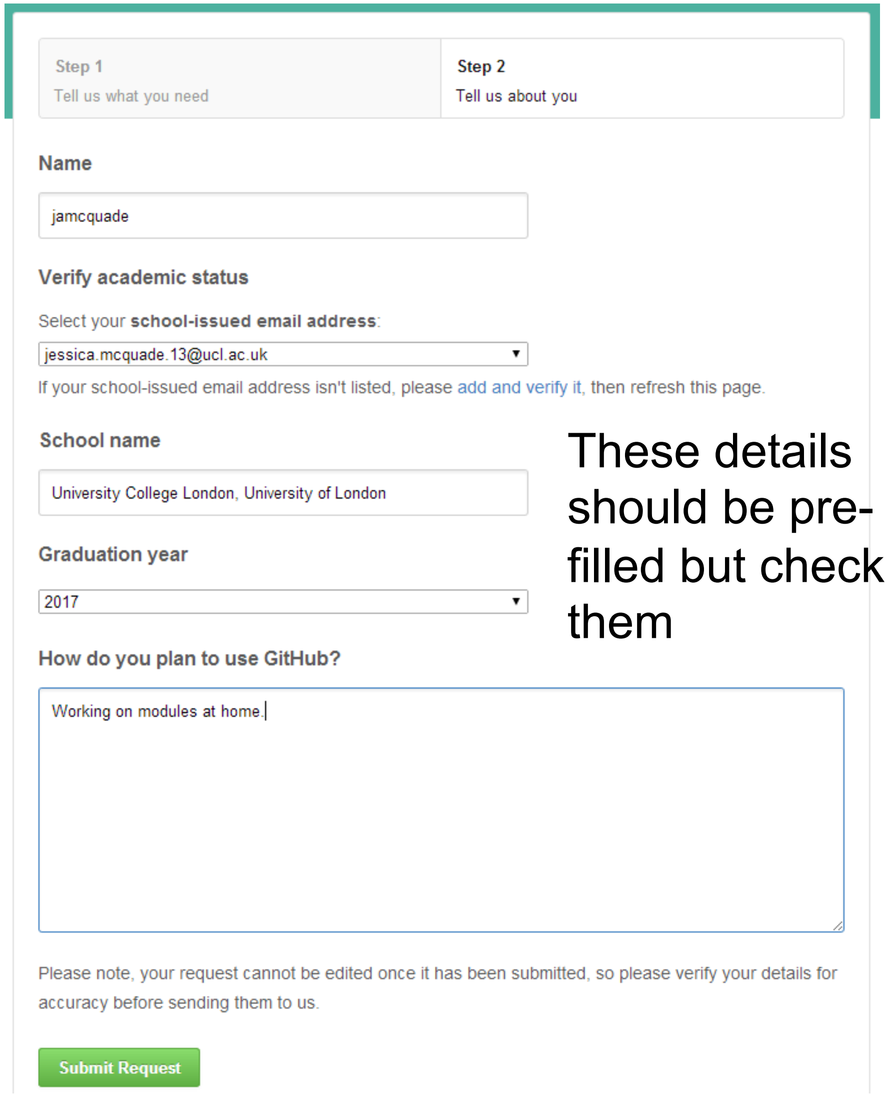
What are you looking to get a discount for?

- Individual account
- Organization account

Next

ms Privacy Security Contact @GitHubEducatic

Select "Student" and "Individual Account"



Step 1 Tell us what you need | Step 2 Tell us about you

Name

jamcquade

Verify academic status

Select your **school-issued email address**:

jessica.mcquade.13@ucl.ac.uk

If your school-issued email address isn't listed, please add and verify it, then refresh this page.

School name

University College London, University of London

Graduation year

2017

How do you plan to use GitHub?

Working on modules at home.

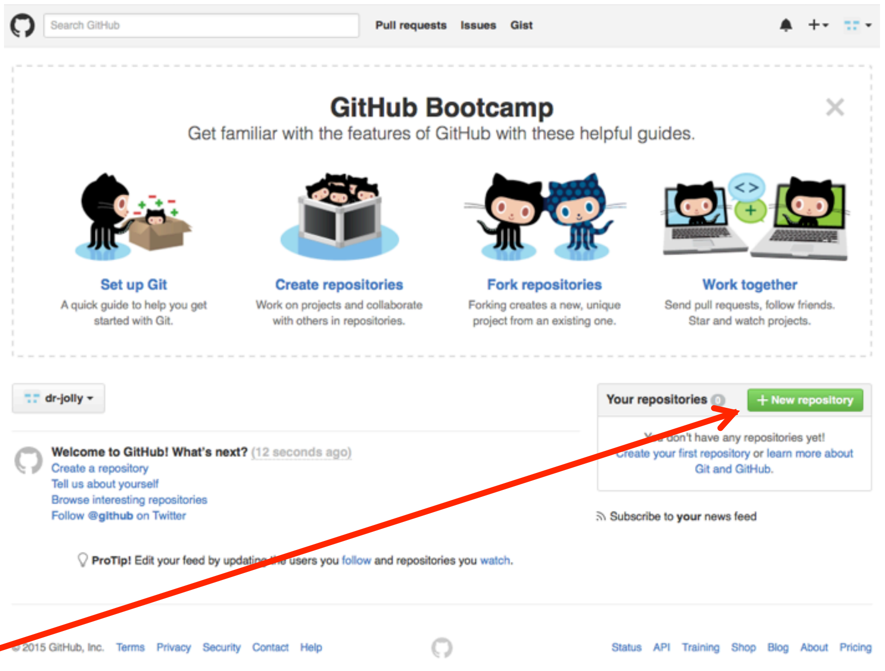
Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.

Submit Request

These details should be pre-filled but check them

GitHub Academic Registration

- You should get a confirmation email reasonably quickly telling you that you have been approved for an academic account.
- Now go back to your GitHub homepage and select “New Repository”.
- We need to create a “bare” repository into which we can clone our existing repository in Eclipse.



GitHub Bootcamp

Get familiar with the features of GitHub with these helpful guides.

Set up Git
A quick guide to help you get started with Git.

Create repositories
Work on projects and collaborate with others in repositories.

Fork repositories
Forking creates a new, unique project from an existing one.

Work together
Send pull requests, follow friends. Star and watch projects.

You don't have any repositories yet!
Create your first repository or learn more about Git and GitHub.

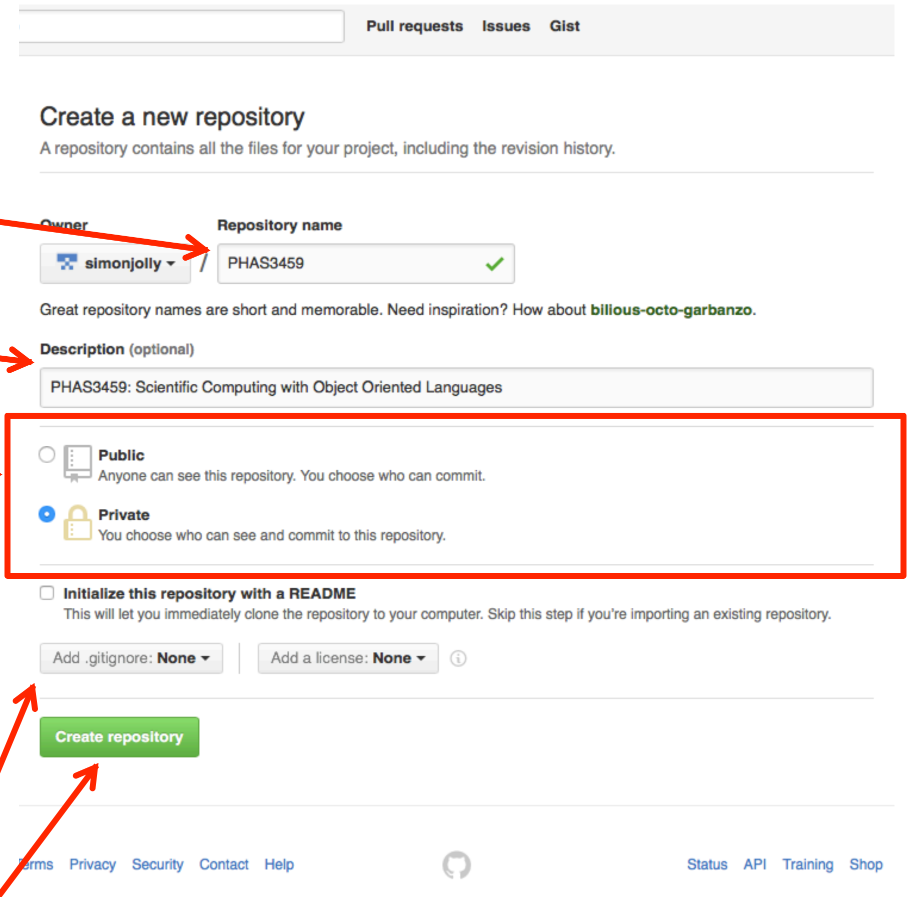
Welcome to GitHub! What's next? (12 seconds ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter

Subscribe to your news feed

ProTip! Edit your feed by updating the users you follow and repositories you watch.

GitHub New Repository

- You can call the repository anything you like, but "PHAS3459" might make sense…
- The Description is optional.
- **MAKE SURE THE REPOSITORY IS PRIVATE!** Public repositories can be seen by anybody with access to the web and a search engine...
- Leave the remaining options as-is.
- Select "Create Repository".



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: simonjolly / Repository name: PHAS3459

Great repository names are short and memorable. Need inspiration? How about bilious-octo-garbanzo.

Description (optional): PHAS3459: Scientific Computing with Object Oriented Languages

( ) Public — Anyone can see this repository. You choose who can commit.

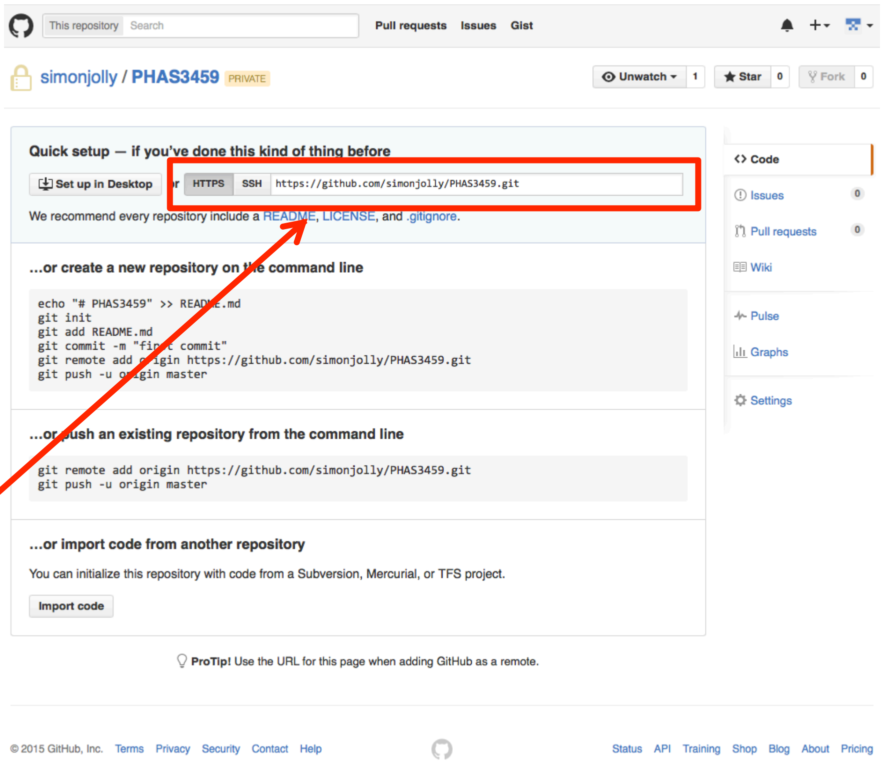
(●) Private — You choose who can see and commit to this repository.

[ ] Initialize this repository with a README — This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Your New Remote Repository

- Once the repository has been created, you'll be shown the Quick Setup screen.
- Don't make any changes here! You don't want to add anything to the repository in GitHub.
- Make a note of the HTTPS URL given at the top: you'll need this in Eclipse.
- Make sure you choose HTTPS (GitHub won't sync via SSH without setting up ssh keys).



simonjolly / PHAS3459 PRIVATE

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/simonjolly/PHAS3459.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# PHAS3459" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/simonjolly/PHAS3459.git
git push -u origin master
```

…or push an existing repository from the command line

```
git remote add origin https://github.com/simonjolly/PHAS3459.git
git push -u origin master
```

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

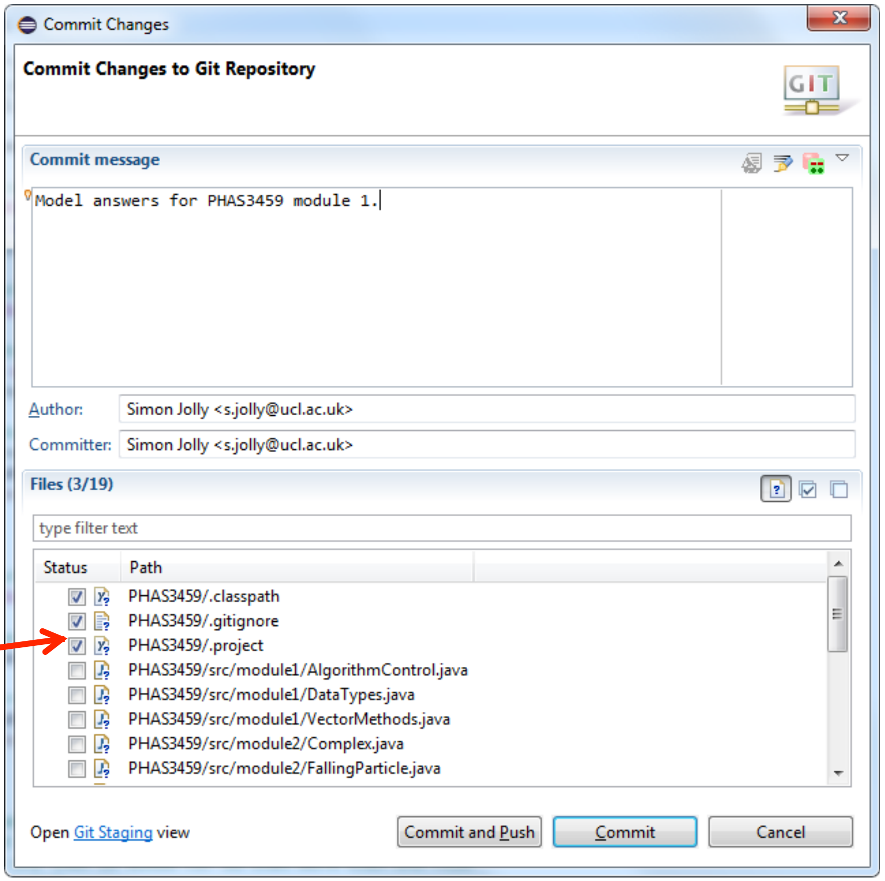
Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

© 2015 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

Commit Eclipse Project Files

- First of all, make sure you have a working Git repository in Eclipse!
- Save and commit your latest work.
- If you want your work to properly sync between multiple versions of Eclipse, make sure you have committed the following 3 files to your repository:
  - .classpath
  - .gitignore
  - .project



```
Model answers for PHAS3459 module 1.
```

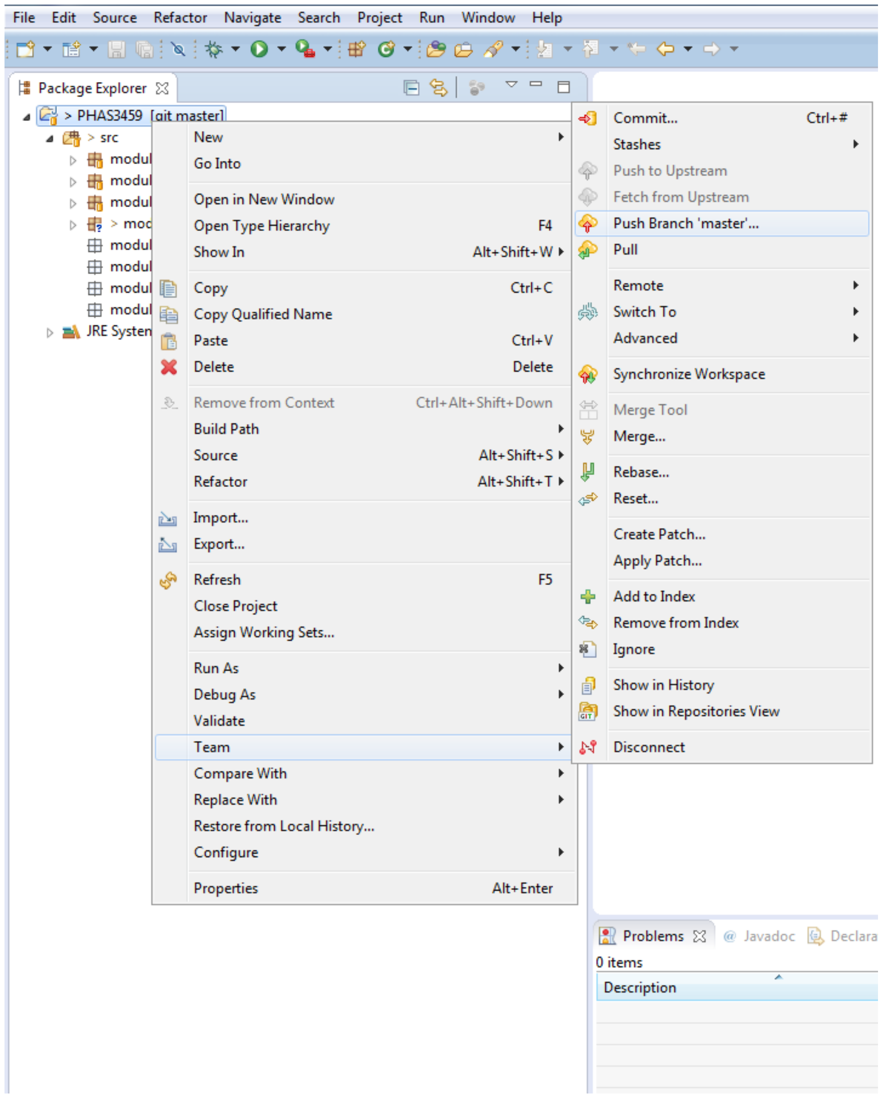
Author: Simon Jolly <s.jolly@ucl.ac.uk>

Committer: Simon Jolly <s.jolly@ucl.ac.uk>

Files (3/19)

Status	Path
✓	PHAS3459/.classpath
✓	PHAS3459/.gitignore
✓	PHAS3459/.project
	PHAS3459/src/module1/AlgorithmControl.java
	PHAS3459/src/module1/DataTypes.java
	PHAS3459/src/module1/VectorMethods.java
	PHAS3459/src/module2/Complex.java
	PHAS3459/src/module2/FallingParticle.java

Push Your Local Repository

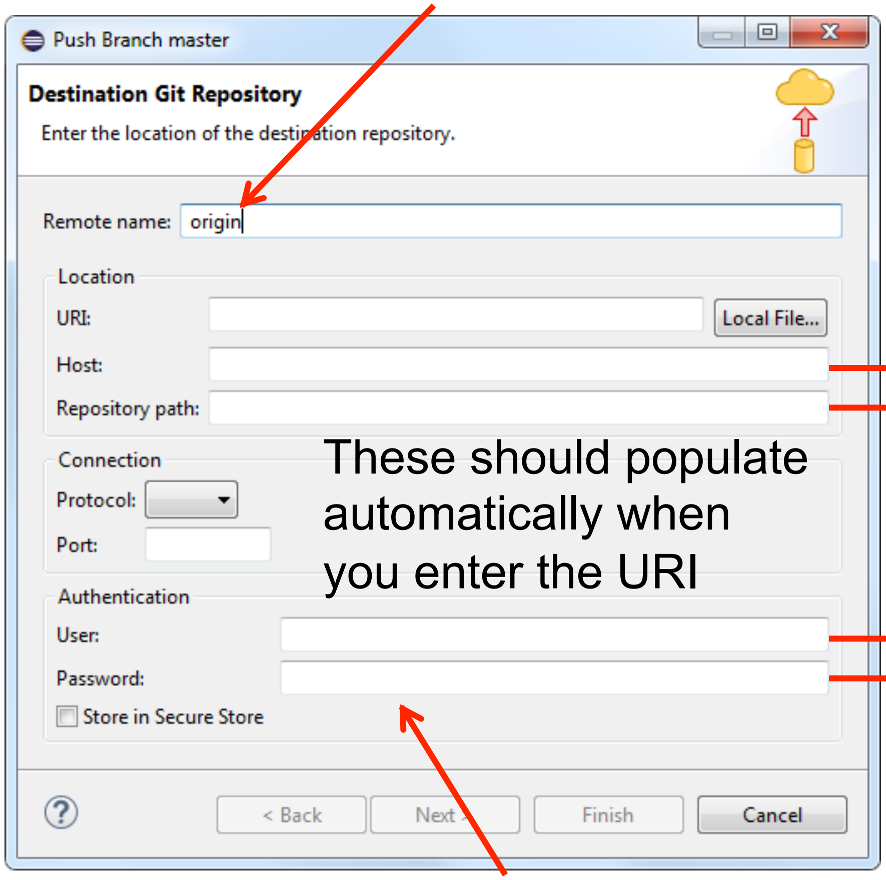


- Now you need to set up the remote to push your repository to.
- We can do this automatically by asking Eclipse to push a repository that doesn't have a remote associated with it.
- Right-click on your project and select "Team → Push Branch 'master'..."

Set Up Remote Repository

Call your Remote name “origin”

Enter in the address for your repository on GitHub



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI:

Host:

Repository path:

Connection

Protocol:

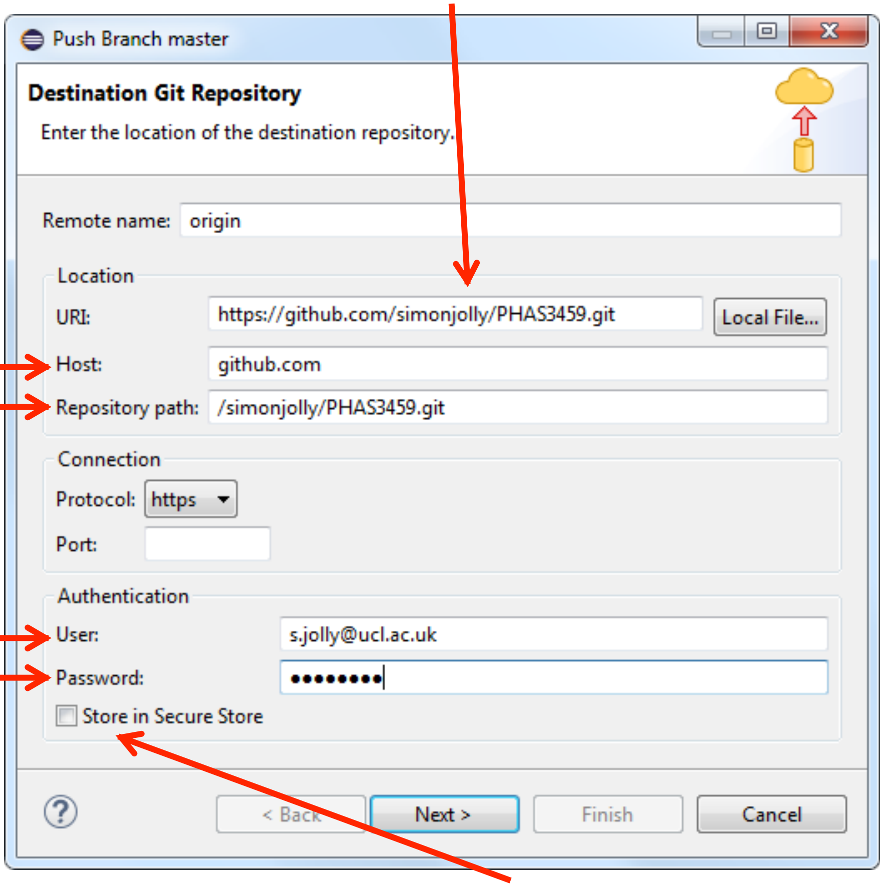
Port:

Authentication

User:

Password:

☐ Store in Secure Store



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI: https://github.com/simonjolly/PHAS3459.git

Host: github.com

Repository path: /simonjolly/PHAS3459.git

Connection

Protocol: https

Port:

Authentication

User: s.jolly@ucl.ac.uk

Password: ●●●●●●●●●

☐ Store in Secure Store

These should populate automatically when you enter the URI

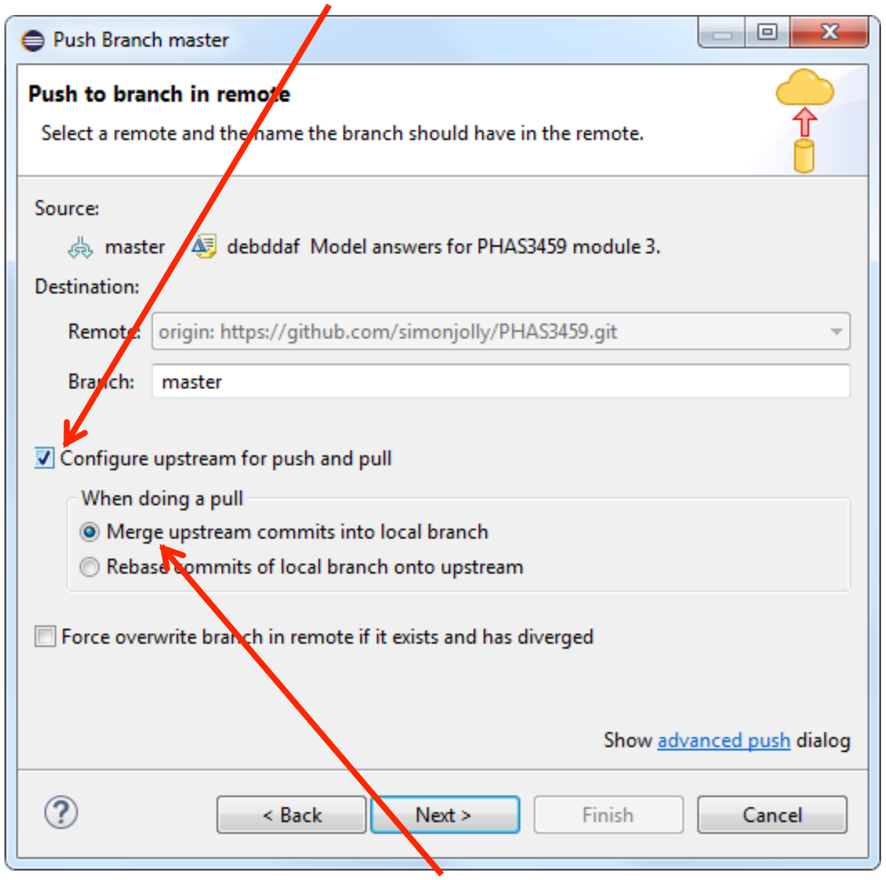
Enter in your login details for GitHub (“User” can either be your username or email)

Select “Store in Secure Store” if you want to save your credentials

Push To Remote Repository

Make sure "Configure upstream…" is selected

Make sure the repository matches with GitHub



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master debddaf Model answers for PHAS3459 module 3.

Destination:

Remote: origin: https://github.com/simonjolly/PHAS3459.git

Branch: master

[✓] Configure upstream for push and pull

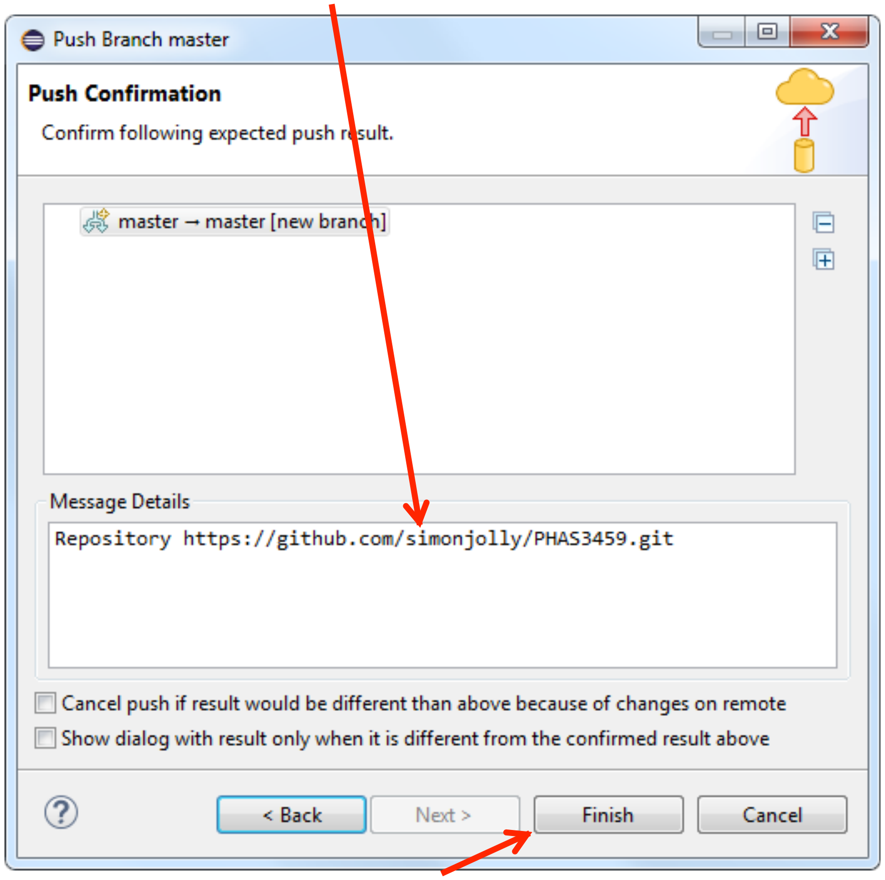
When doing a pull

(●) Merge upstream commits into local branch

( ) Rebase commits of local branch onto upstream

[ ] Force overwrite branch in remote if it exists and has diverged

Show advanced push dialog



Push Branch master

Push Confirmation

Confirm following expected push result.

master → master [new branch]

Message Details

```
Repository https://github.com/simonjolly/PHAS3459.git
```

[ ] Cancel push if result would be different than above because of changes on remote

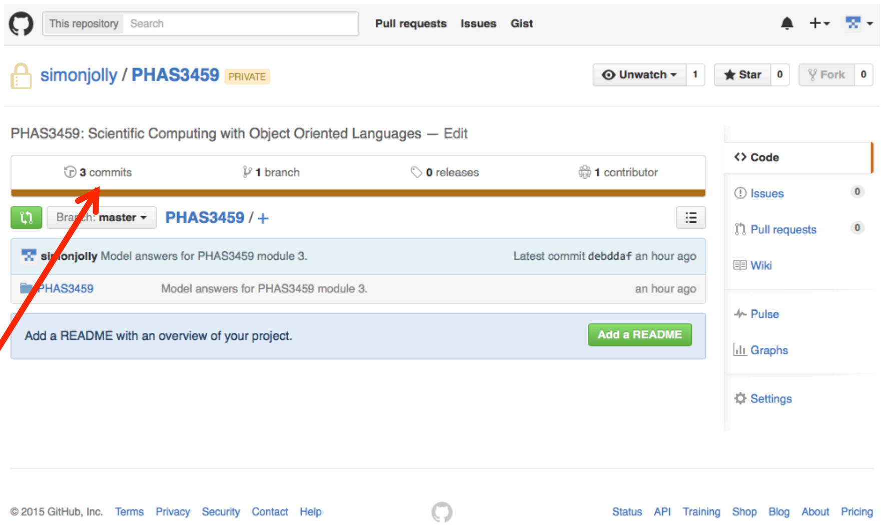
[ ] Show dialog with result only when it is different from the confirmed result above

Select "Merge upstream commits…"

Click "Finish".

Synced Repository On GitHub

- You should now have successfully synced your repository with GitHub!
- If you go back to GitHub, you should now be able to see the repository you've just pushed over.
- Click on the "commits" link and make sure GitHub matches what you see when you right-click on the Project in Eclipse and select "Show in History".


This repository Search Pull requests Issues Gist

simonjolly / PHAS3459 PRIVATE

Unwatch 1 Star 0 Fork 0

PHAS3459: Scientific Computing with Object Oriented Languages — Edit

3 commits 1 branch 0 releases 1 contributor

Branch: master PHAS3459 / +

simonjolly Model answers for PHAS3459 module 3. Latest commit debddaf an hour ago

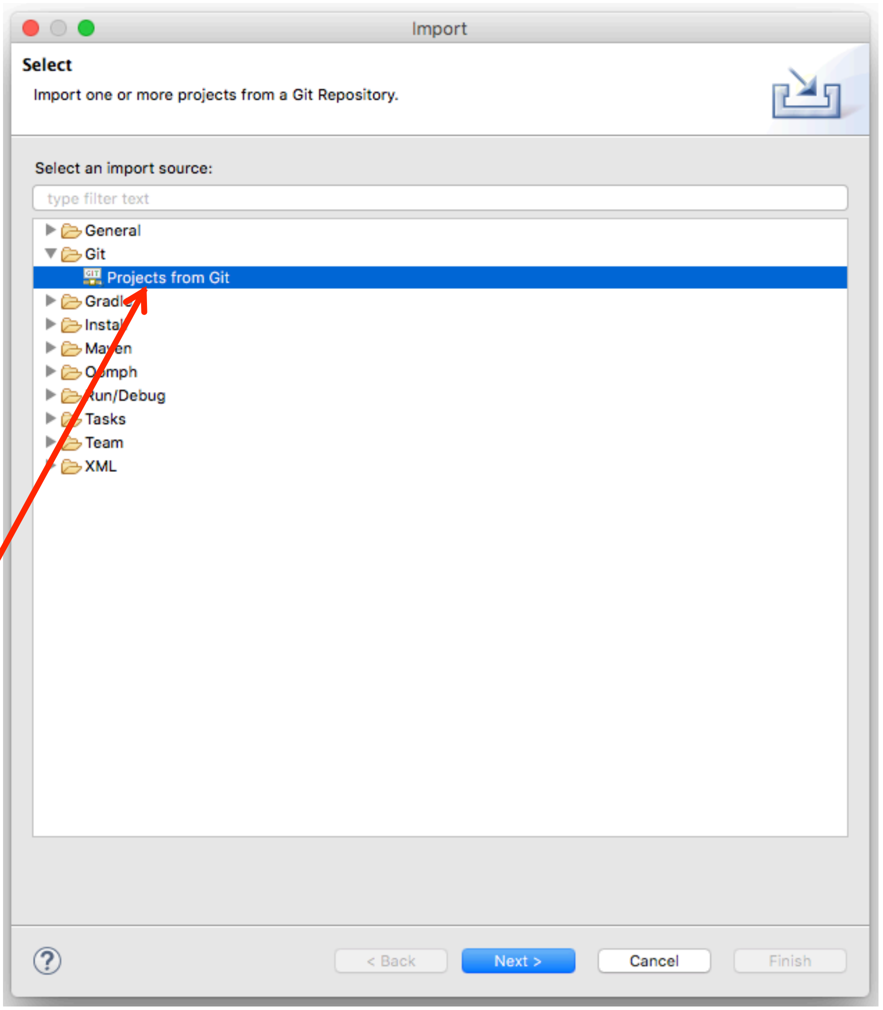
PHAS3459 Model answers for PHAS3459 module 3. an hour ago

Add a README with an overview of your project. Add a README

© 2015 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

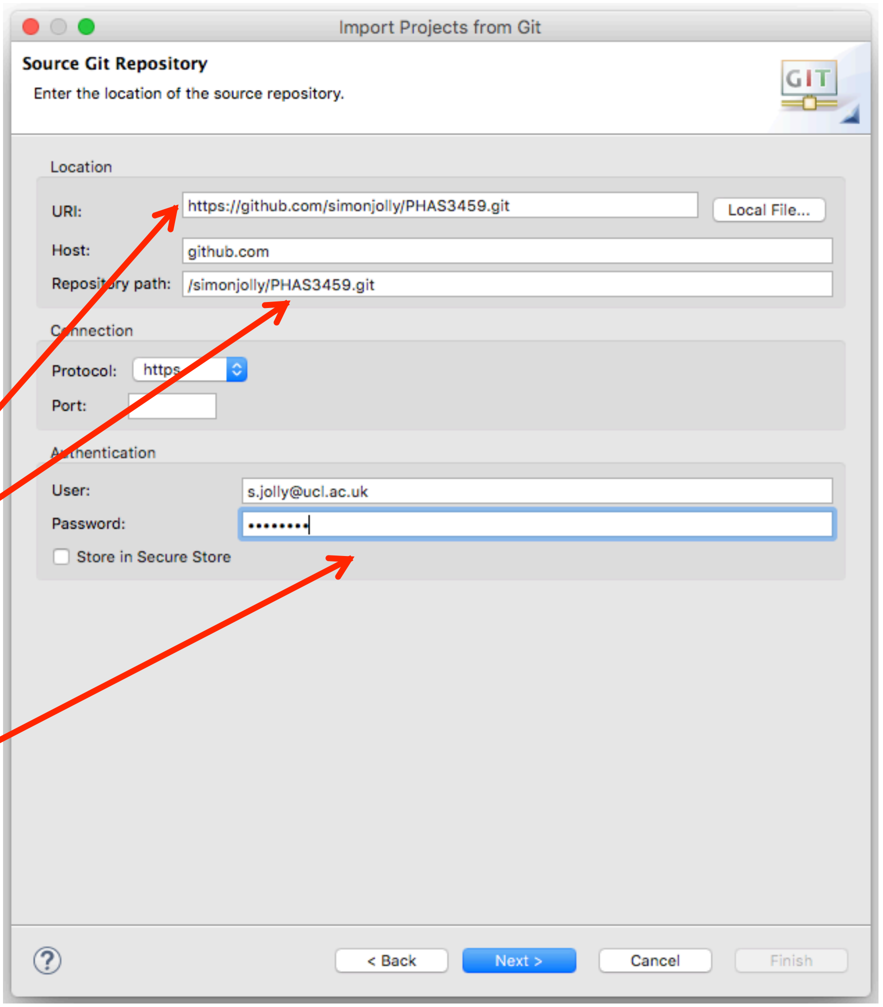
Setting Up A New Project

- Now we'd like to pull that repository from GitHub to another local machine to synchronise a second machine with the same repository.
- Start off with a completely bare workspace in Eclipse (make sure you create a new directory on your local machine called "workspace").
- Select "File → Import…"
- From the Import window, select "Git → Projects from Git…"



Import Repository From GitHub

- From "Select Repository Source", choose "Clone URI".
- Under "Source Git Repository", enter the information from GitHub for the repository you created previously:
  - Enter the URI listed for your repository on the GitHub web page.
  - The remaining entries should populate automatically.
  - Enter your username and password for GitHub.
- Click "Next".


Import Projects from Git

Source Git Repository

Enter the location of the source repository.

Location

URI:	https://github.com/simonjolly/PHAS3459.git
Host:	github.com
Repository path:	/simonjolly/PHAS3459.git

Connection

Protocol:	https
Port:	

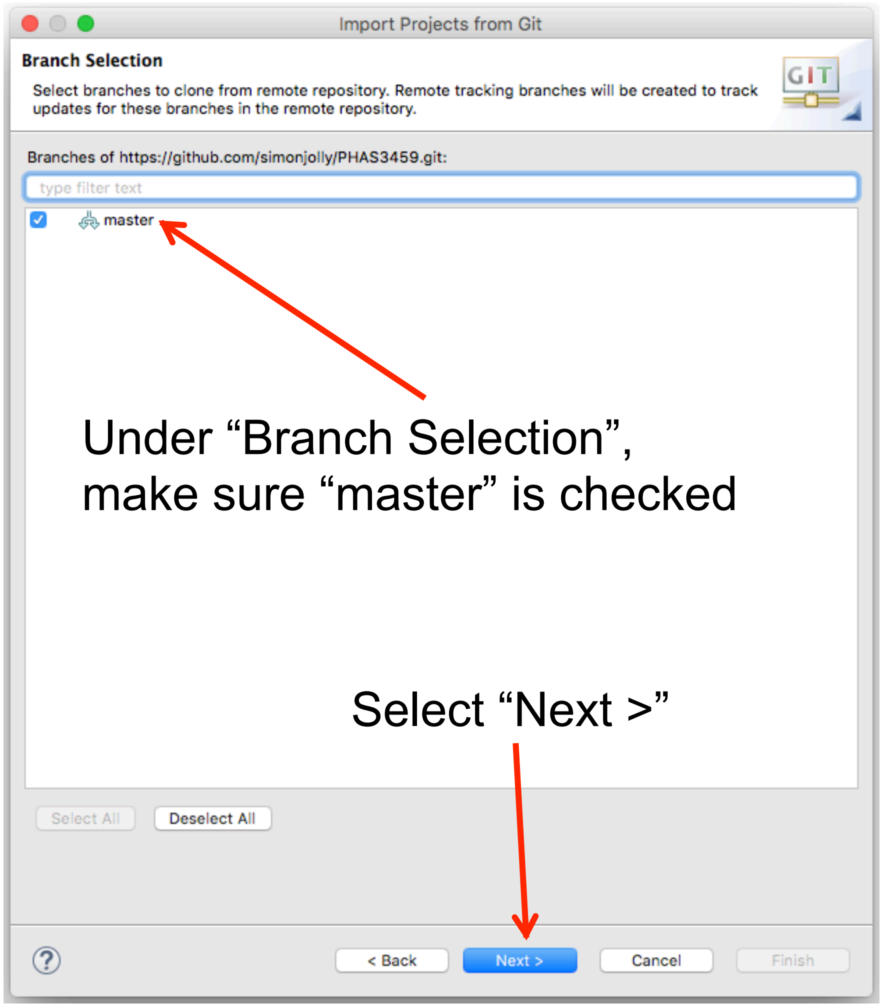
Authentication

User:	s.jolly@ucl.ac.uk
Password:	••••••••

☐ Store in Secure Store

Import Projects From Git

Select "Clone submodules"



Import Projects from Git

Branch Selection

Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

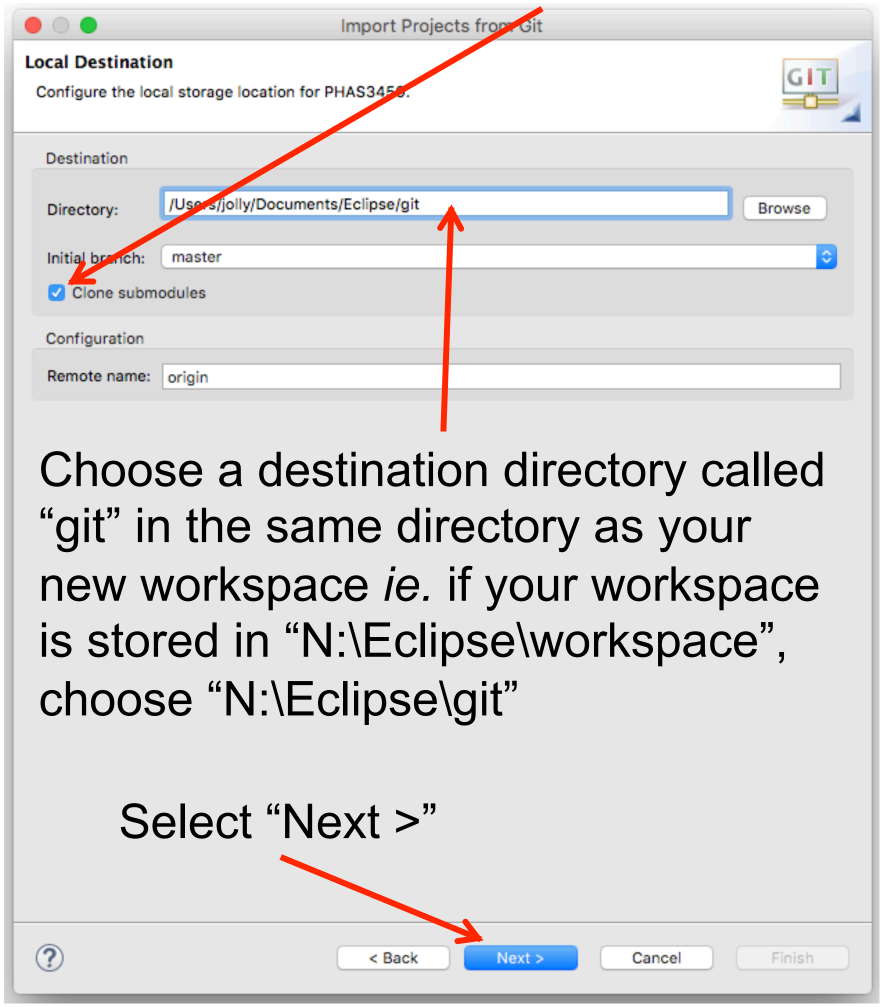
Branches of https://github.com/simonjolly/PHAS3459.git:

type filter text

✓ master

Under "Branch Selection", make sure "master" is checked

Select "Next >"



Import Projects from Git

Local Destination

Configure the local storage location for PHAS3459.

Destination

Directory: /Users/jolly/Documents/Eclipse/git

Initial branch: master

✓ Clone submodules

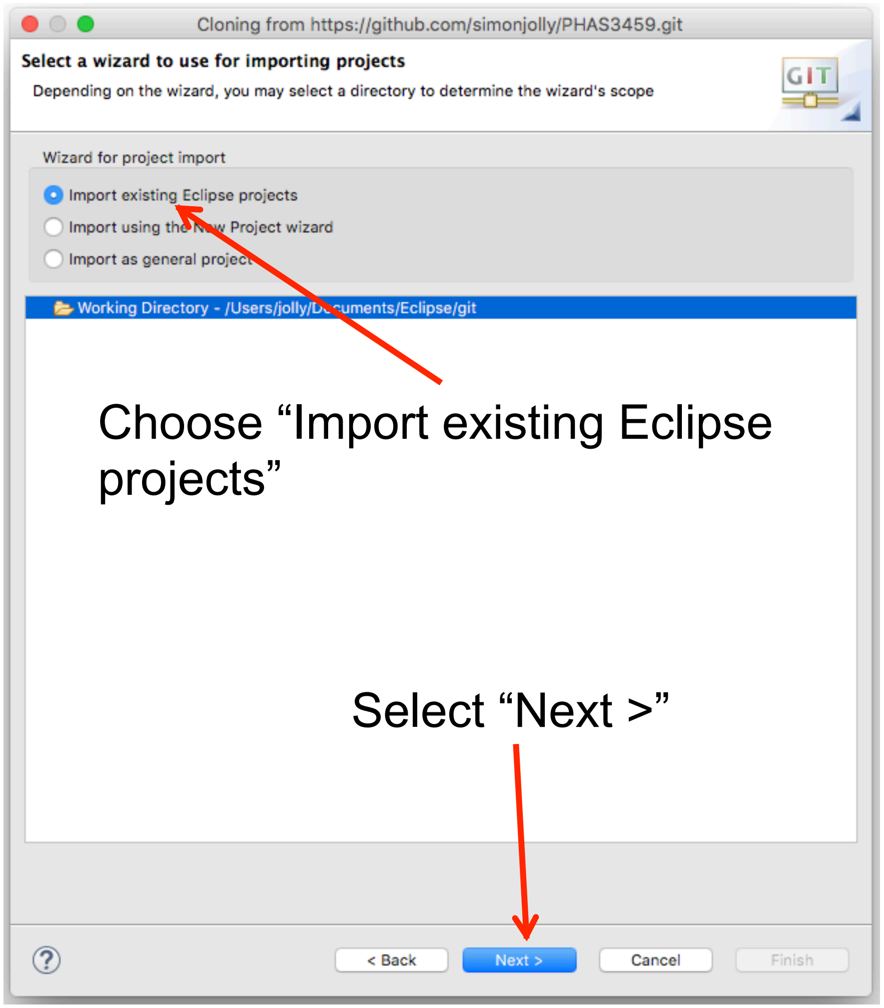
Configuration

Remote name: origin

Choose a destination directory called "git" in the same directory as your new workspace *ie.* if your workspace is stored in "N:\Eclipse\workspace", choose "N:\Eclipse\git"

Select "Next >"

Import As Eclipse Project



Cloning from https://github.com/simonjolly/PHAS3459.git

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

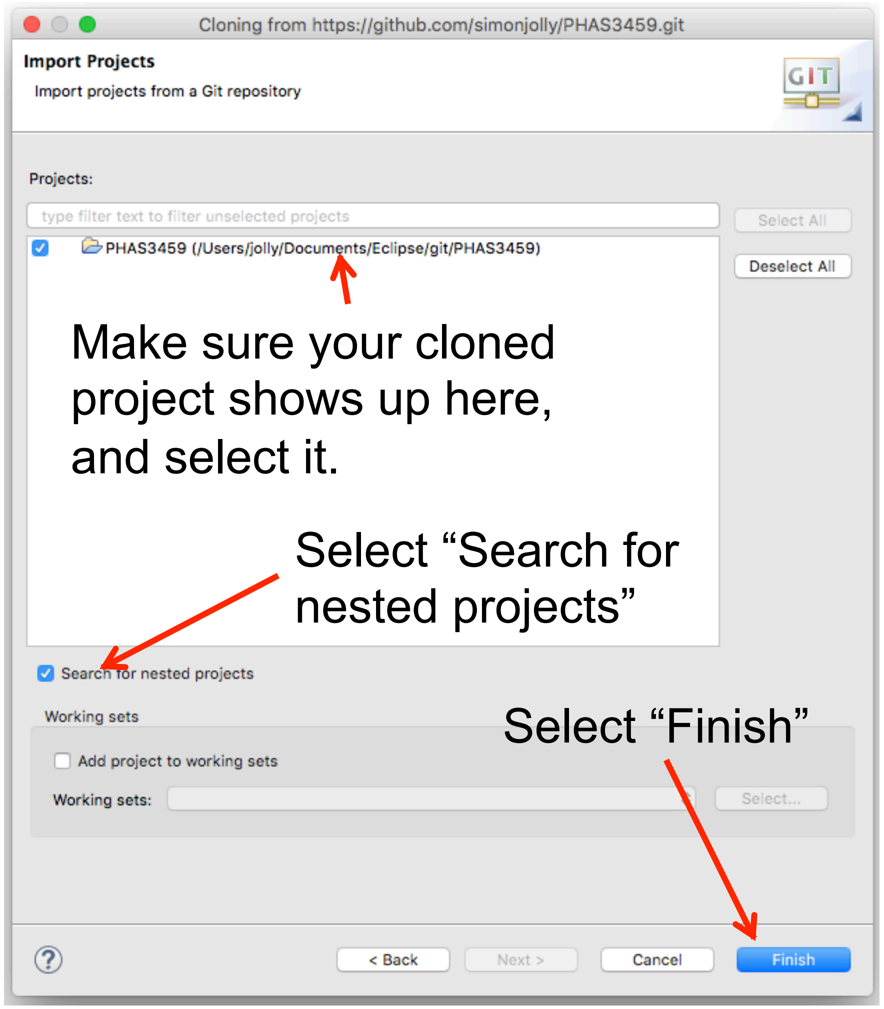
Wizard for project import

- [✓ Import existing Eclipse projects]
- Import using the New Project wizard
- Import as general project

Working Directory - /Users/jolly/Documents/Eclipse/git

Choose “Import existing Eclipse projects”

Select “Next >”



Cloning from https://github.com/simonjolly/PHAS3459.git

Import Projects

Import projects from a Git repository

Projects:

type filter text to filter unselected projects

- [✓] PHAS3459 (/Users/jolly/Documents/Eclipse/git/PHAS3459)

- [✓] Search for nested projects

Working sets

- [ ] Add project to working sets

Working sets:

Make sure your cloned project shows up here, and select it.

Select “Search for nested projects”

Select “Finish”

Your “New” Git Repository

- You should have now cloned your Git repository into your new workspace.
- This will allow you to keep multiple versions of the same Eclipse workspace synchronised across various machines.
- To update your local repository, select “Team → Push to Upstream”.
- To update your remote (GitHub) repository, select “Team → Pull”.
- The golden rules are:
  - Before starting work, always pull down the latest commits from your remote repository.
  - Once you’ve finished work, always push your latest commits to your remote repository.
  - **KEEP YOUR GITHUB REPOSITORY PRIVATE!!**
- Thanks to Richard Tweed and Jessica McQuade for acting as guinea pigs...