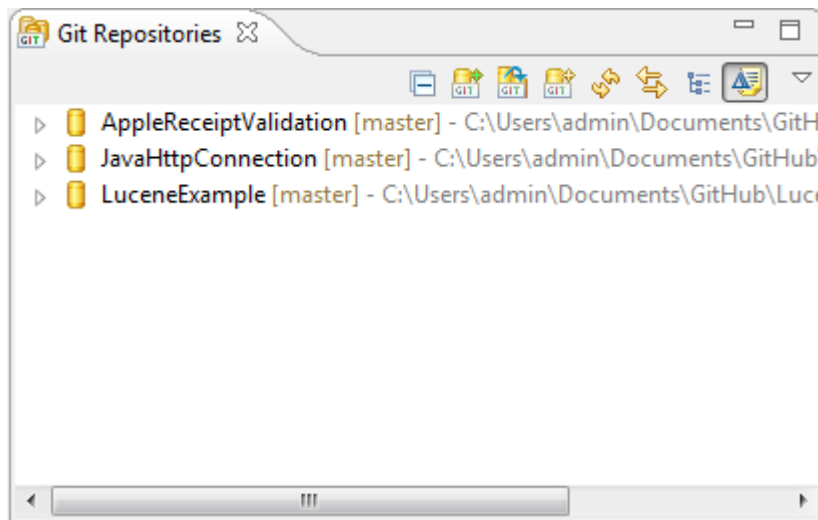



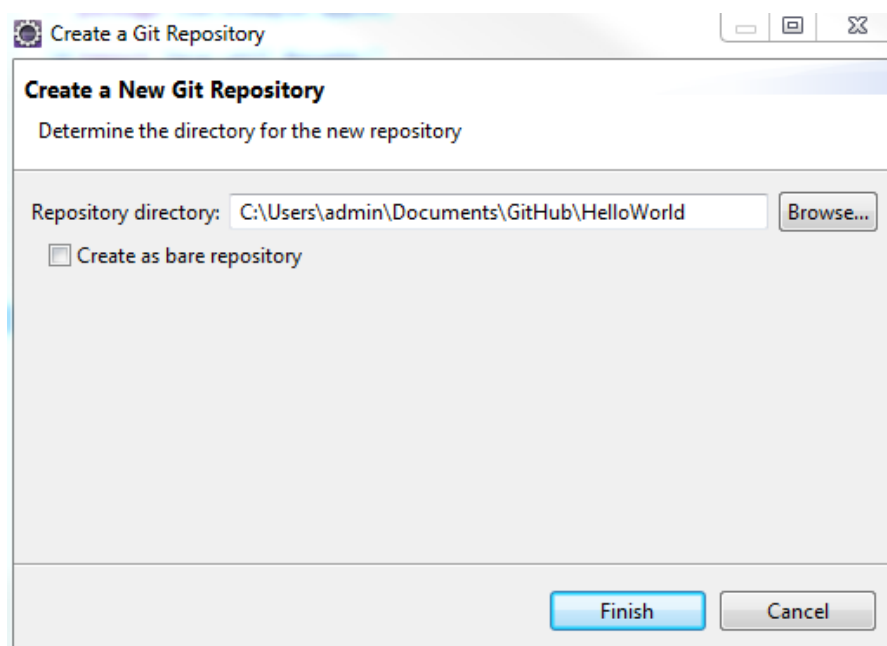
# GitHub Guide

## 1. Create a new local repository using eclipse.

Make sure you can see your Git repositories by clicking on *Window > Show View > Other...* and selecting 'Git Repositories' from the list. You should see this in the bottom left under the project explorer in eclipse:



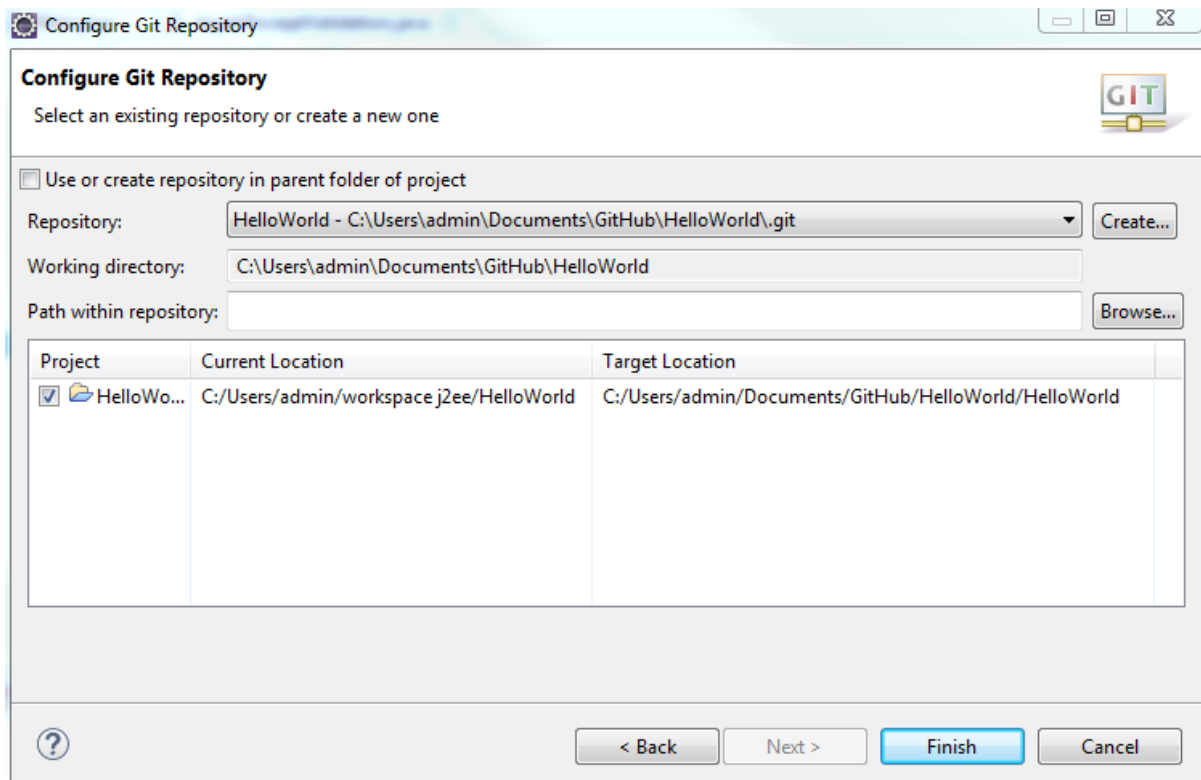
Click on  and type in the directory you wish to create the repository in in the 'Repository directory:' field:



Then click 'Finish'.

## 2. Share the project with the local repository

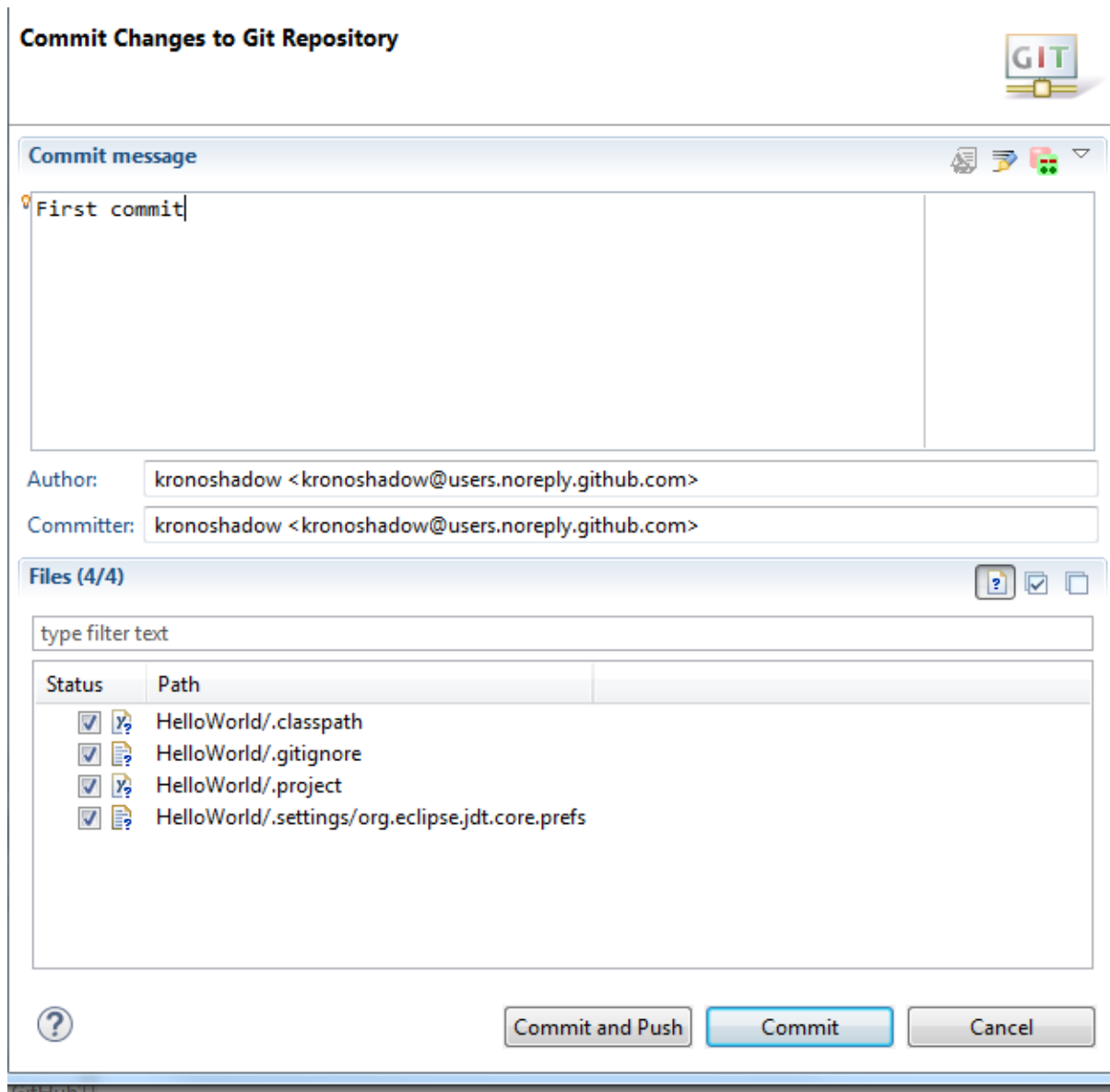
Right click on the project you wish to share then click on *Team > Share Project...* Then select 'Git' as the repository type and click 'Next'. Select the repository on the next page like so:



Click 'Finish'. The project should now be associated with the new local repository you just created earlier. Remember that your project folder will move from the eclipse work directory into the directory containing the local repository, so any absolute classpaths to JARs or libraries will need to be changed to point to the local repository directory.

### 3. Commit code to local repository

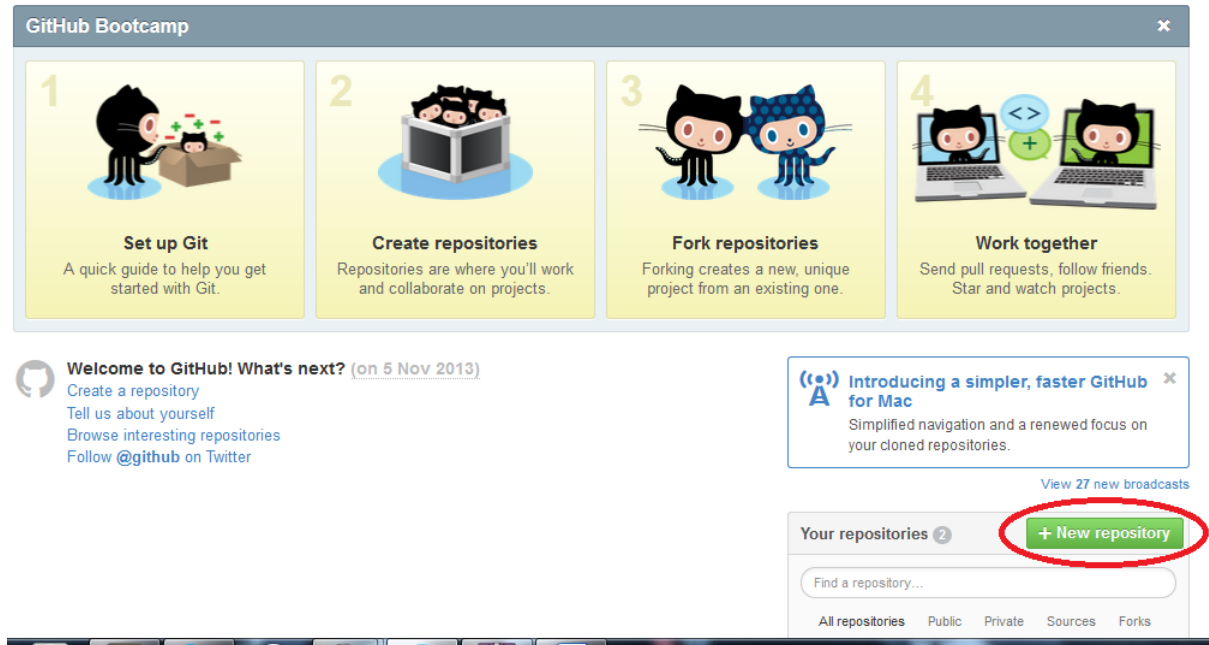
Commit the code by right clicking on the project folder then go to *Team > Commit...* Then write a commit description, select the files you want to commit and click 'Commit':



If this is your first commit, git will create a 'master' branch automatically. Clicking on 'Commit and Push' will commit the code, but an error will pop up about not configuring the push, this is done in the next step.

## 4. Connect local repository with GitHub (configure push)

Login into GitHub and create a new repository by clicking on the green 'New repository' button:



Search or type a command ⌕ [Explore](#) [Gist](#) [Blog](#) [Help](#) kronoshadow +

Owner kronoshadow / Repository name HelloWorld ✓

Great repository names are short and memorable. Need inspiration? How about [tripping-batman](#).

Description (optional)

☒ **Public**  
Anyone can see this repository. You choose who can commit.

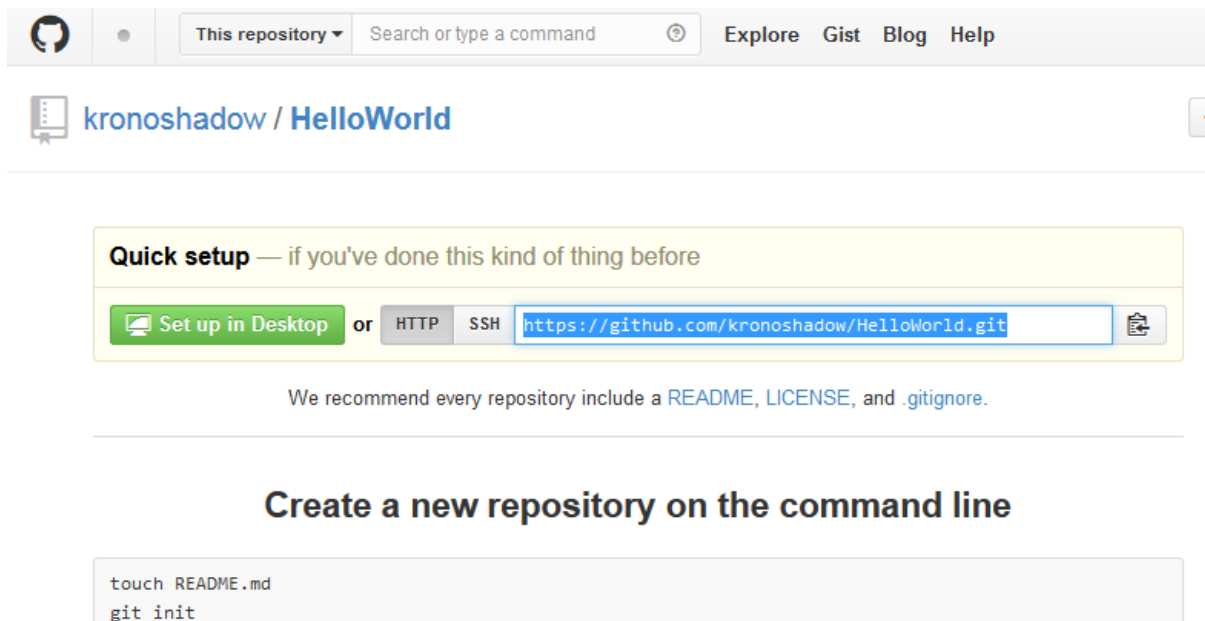
☐ **Private**  
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

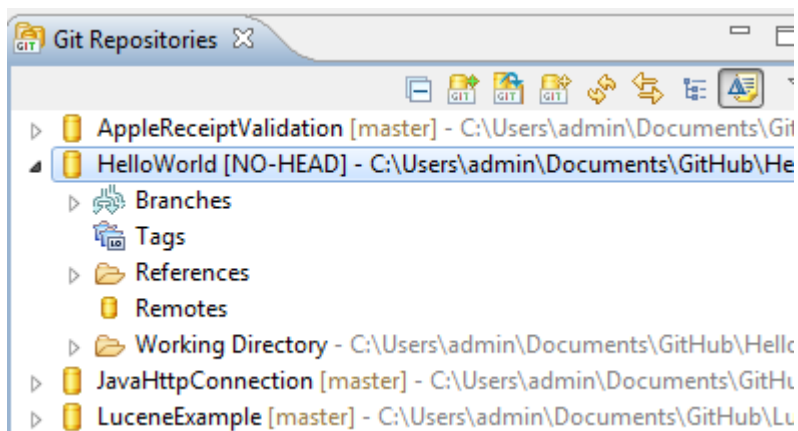
Add .gitignore: **None** | Add a license: **None** ?

**Create repository**

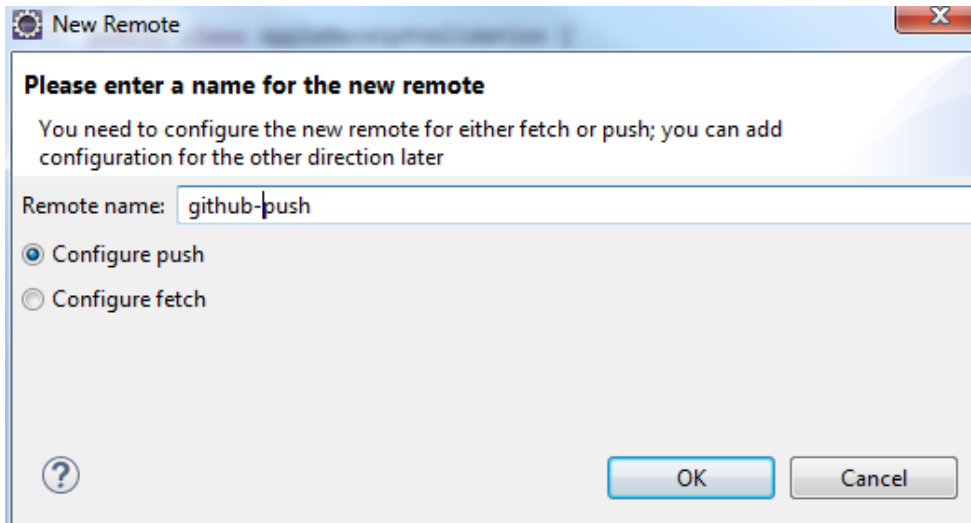
Then copy the HTTP git link and paste it somewhere safe for later:



In eclipse, go to the Git repositories view and expand the repository you would like to connect GitHub with so that it looks like this:



Then right click 'Remotes' and select 'Create remote...'. Give a name to the connection (eg: 'github-push') and select configure push then click Ok:



The 'New Remote' dialog box has a title bar with a gear icon and a close button. The main content area has a header 'Please enter a name for the new remote' followed by a sub-header 'You need to configure the new remote for either fetch or push; you can add configuration for the other direction later'. Below this is a text field labeled 'Remote name:' containing the text 'github-push'. There are two radio buttons: 'Configure push' (selected) and 'Configure fetch'. At the bottom left is a help icon (question mark in a circle). At the bottom right are 'OK' and 'Cancel' buttons.

**New Remote**

Please enter a name for the new remote

You need to configure the new remote for either fetch or push; you can add configuration for the other direction later

Remote name: github-push

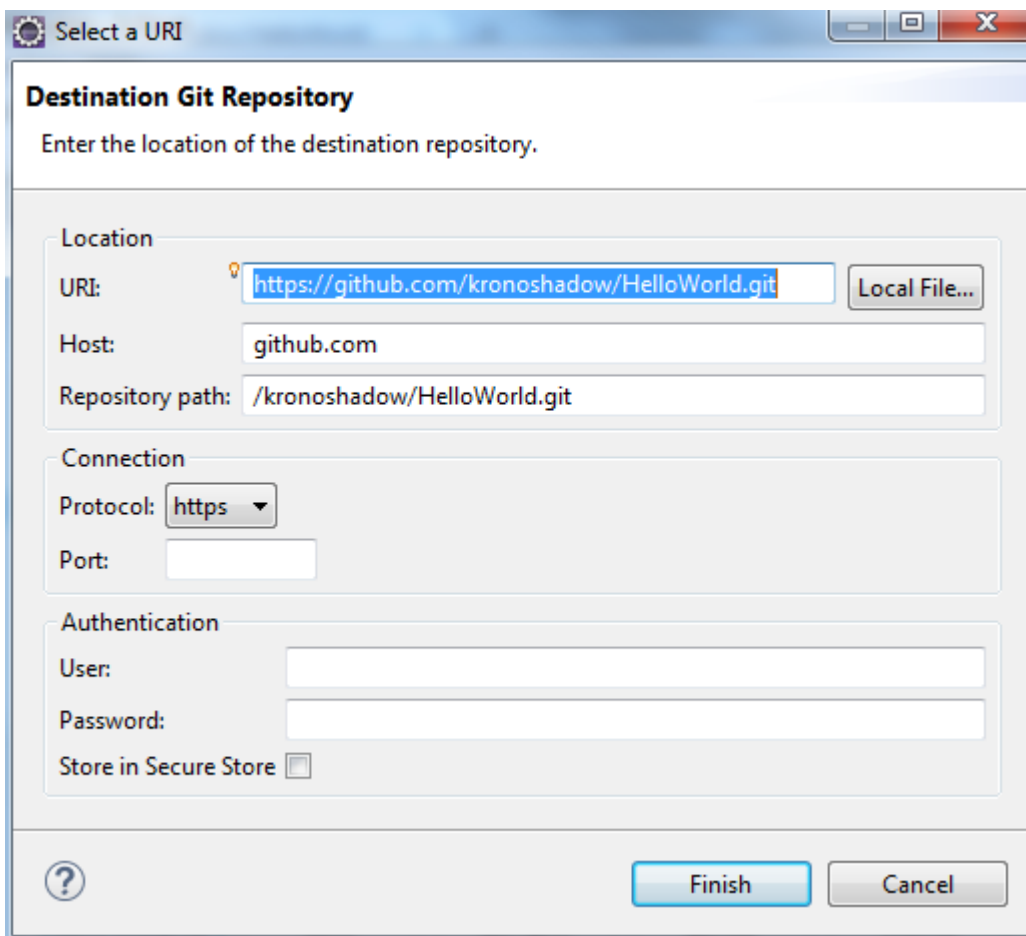
☒ Configure push

☐ Configure fetch

?

OK Cancel

On the next page, next to the URI field, click 'Change...'. In the popup, paste the HTTP git URL from earlier into the 'URI' field and click 'Finish':



The 'Select a URI' dialog box has a title bar with a gear icon and window control buttons. The main content area has a header 'Destination Git Repository' followed by a sub-header 'Enter the location of the destination repository.'. Below this is a 'Location' section with a 'URI:' label and a text field containing 'https://github.com/kronoshadow/HelloWorld.git'. To the right of the URI field is a 'Local File...' button. Below the URI field are 'Host:' and 'Repository path:' labels with text fields containing 'github.com' and '/kronoshadow/HelloWorld.git' respectively. Below the 'Location' section is a 'Connection' section with a 'Protocol:' dropdown menu set to 'https' and a 'Port:' text field. Below the 'Connection' section is an 'Authentication' section with 'User:' and 'Password:' text fields, and a 'Store in Secure Store' checkbox. At the bottom left is a help icon (question mark in a circle). At the bottom right are 'Finish' and 'Cancel' buttons.

**Select a URI**

**Destination Git Repository**

Enter the location of the destination repository.

Location

URI: https://github.com/kronoshadow/HelloWorld.git Local File...

Host: github.com

Repository path: /kronoshadow/HelloWorld.git

Connection

Protocol: https

Port:

Authentication

User:

Password:

Store in Secure Store ☐

?

Finish Cancel

Then in the 'Ref mappings' section, click the bottom button called 'Advanced...'. In the popup, select the dropdown for the 'Source ref' field and select 'master [branch]' then click on 'Add Spec' button on the right and then 'Finish':

### Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref:  

refs/heads/master

Destination ref:  
refs/heads/master

+ Add Spec

Add delete ref specification

Remote ref to delete: \*

x Add Spec

Add predefined specification

Add Configured Push Specs

Add All Branches Spec

Add All Tags Spec

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove

Force Update All Specs

Remove All Specs

☐ Save specifications in 'github-push' configuration

The final configuration should look like this:

**Configure push for remote 'github-push'**



In order to use a remote for push, you must specify at least one URI and at least one ref mapping


URI:

► Push URIs

Ref mappings


Click 'Save and Push' to upload the repository to GitHub. Type in your GitHub username and password if eclipse asks you for it. If you go into the GitHub repository you should see your changes:


 This repository ▼  ⌕ [Explore](#) [Gist](#) [Blog](#) [Help](#) 


 **kronoshadow / HelloWorld** Unwatch ▼ 1


**Description**



**Website**  
  or [cancel](#)

 1 commit



 1 branch


 0 releases

 1 contributor

 branch: master ▼ **HelloWorld / +** 

First commit

 **kronoshadow** authored 11 minutes ago latest commit 4188008dee 

 HelloWorld First commit 11 minutes ago

We recommend [adding a README](#) to this repository to help give people an overview of your project.