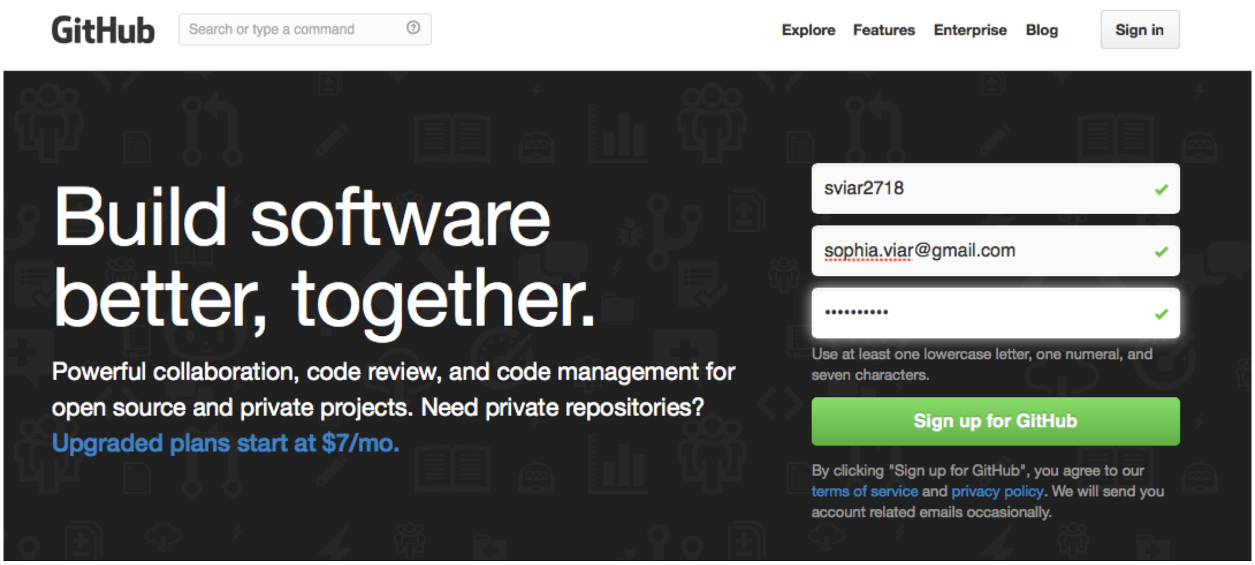
Step 1: First get an email account or use your parents email account.
Step 2: Open a browser and navigate to https://github.com
Step 3: Pick a username and password on GitHub to get started:



Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at $7/mo.

sviar2718 ✓

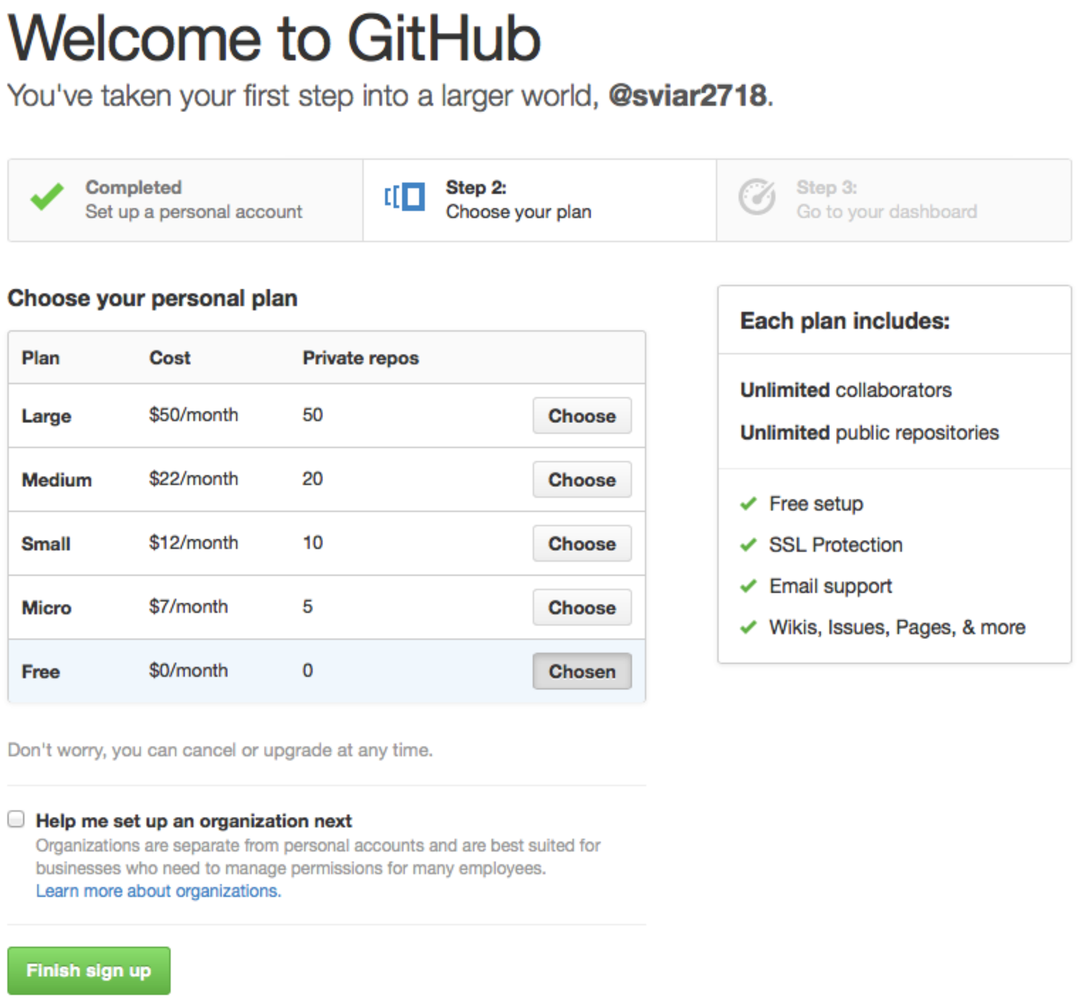
sophia.viar@gmail.com ✓

•••••••••• ✓

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

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

Click on the “Sign up for GitHub”. Now you should see:



Welcome to GitHub

You've taken your first step into a larger world, **@sviar2718**.

✓ Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost	Private repos	
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

Each plan includes:

Unlimited collaborators
Unlimited public repositories

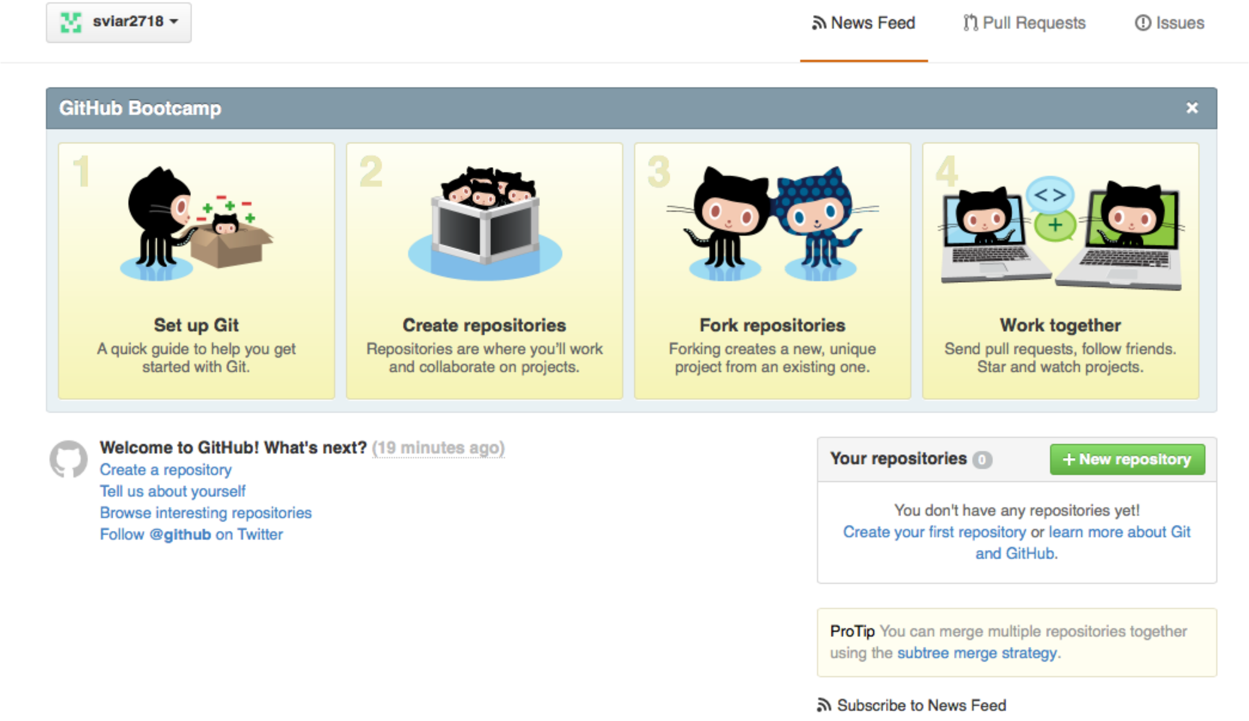
✓ Free setup
✓ SSL Protection
✓ Email support
✓ Wikis, Issues, Pages, & more

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

□ **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.

You will want to select “Free” and click on “Finish sign Up”

Now you should see the following screen:



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

Create repositories
Repositories are where you'll work and collaborate on projects.

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

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

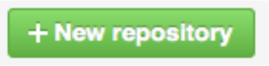
Welcome to GitHub! What's next? (19 minutes ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow **@github** on Twitter

Your repositories 0

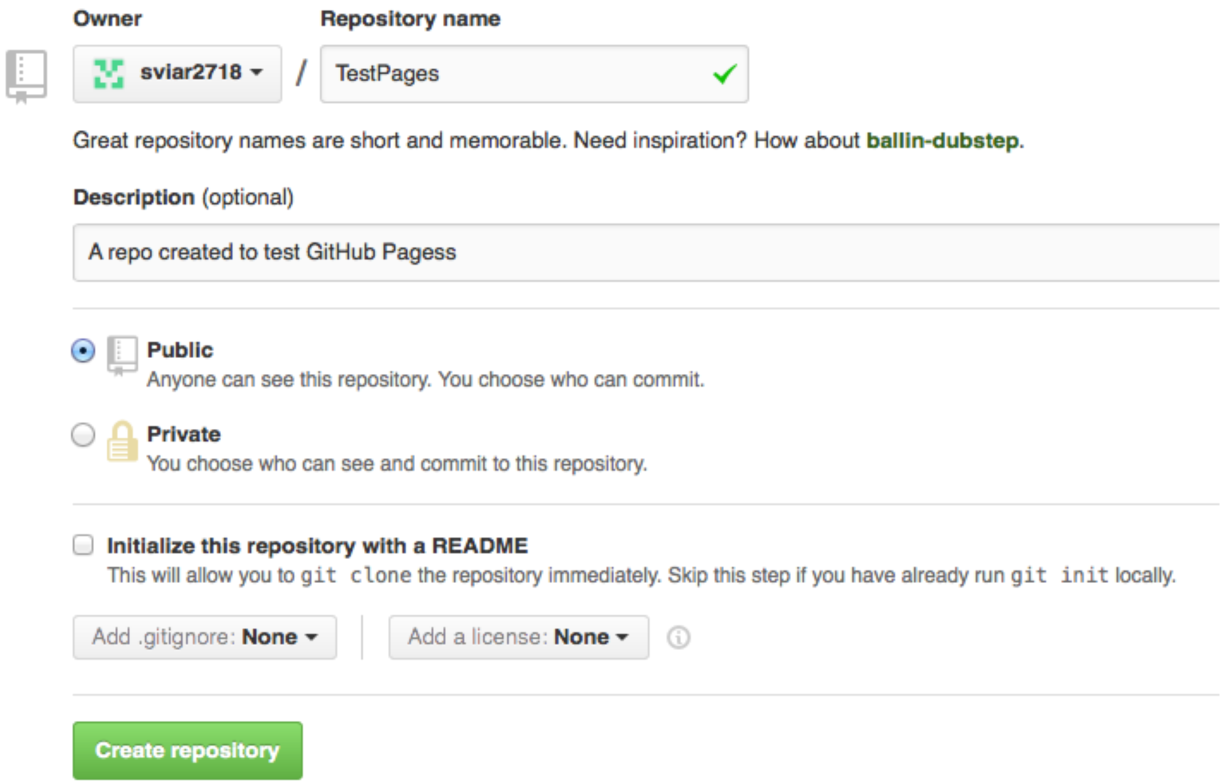
You don't have any repositories yet!
Create your first repository or learn more about Git and GitHub.

ProTip You can merge multiple repositories together using the subtree merge strategy.

Click on



Now give your repository a name like “TestPages” and a description like “A repo created to test GitHub Pages”. You’ll need to keep it “public” since you have a free account. You should see the following:



Owner sviar2718 / **Repository name** TestPages ✓

Great repository names are short and memorable. Need inspiration? How about **ballin-dubstep**.

Description (optional)

A repo created to test GitHub Pagess

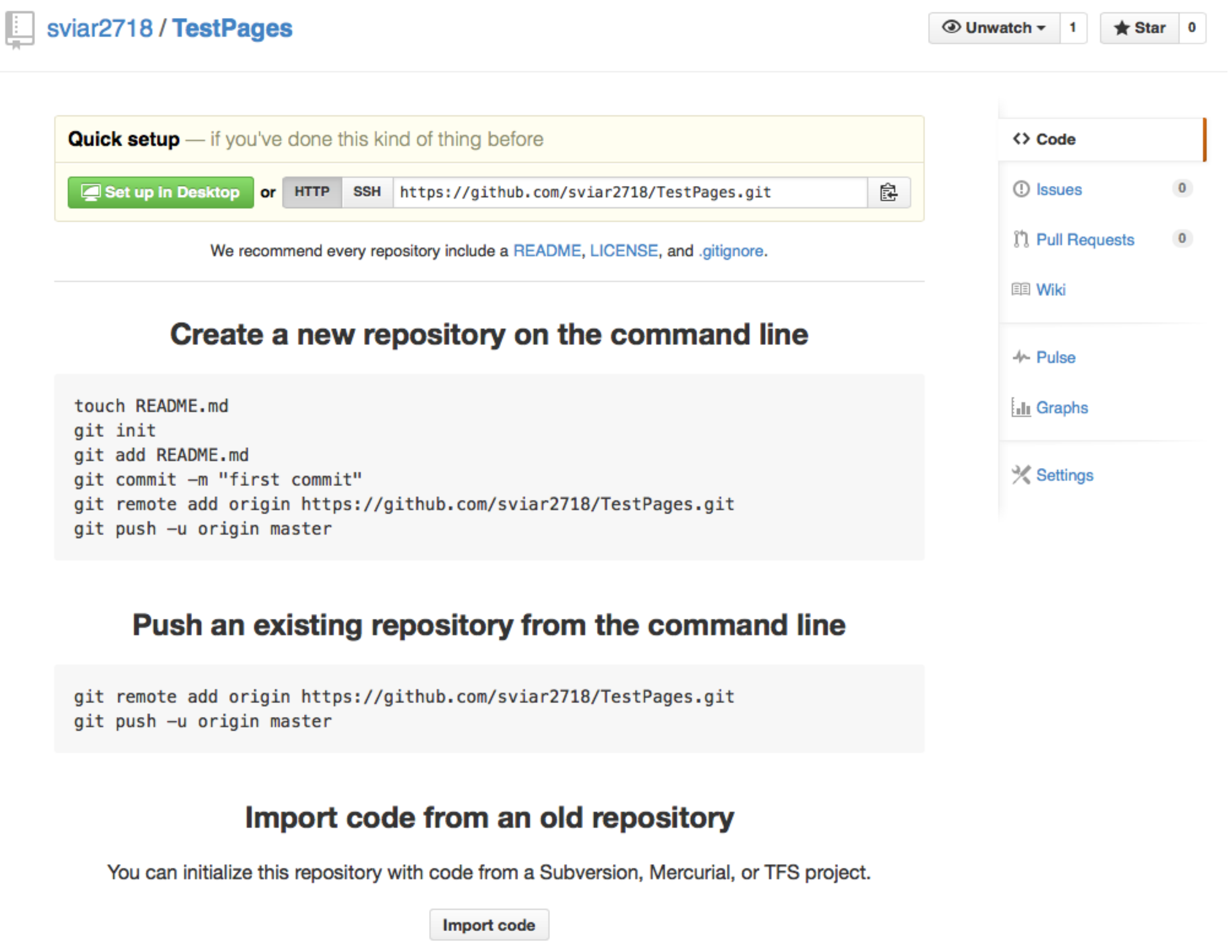
● **Public**
Anyone can see this repository. You choose who can commit.

○ **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**

Now click on “Create repository”. You should see:



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

https://github.com/sviar2718/TestPages.git

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

Create a new repository on the command line

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/sviar2718/TestPages.git
git push -u origin master
```

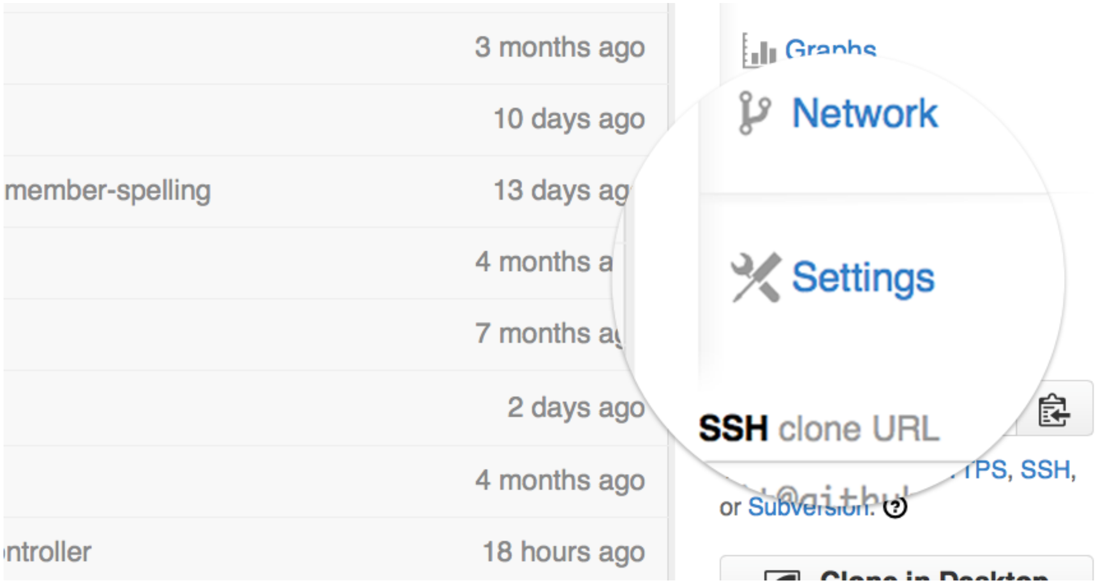
Push an existing repository from the command line

```
git remote add origin https://github.com/sviar2718/TestPages.git
git push -u origin master
```

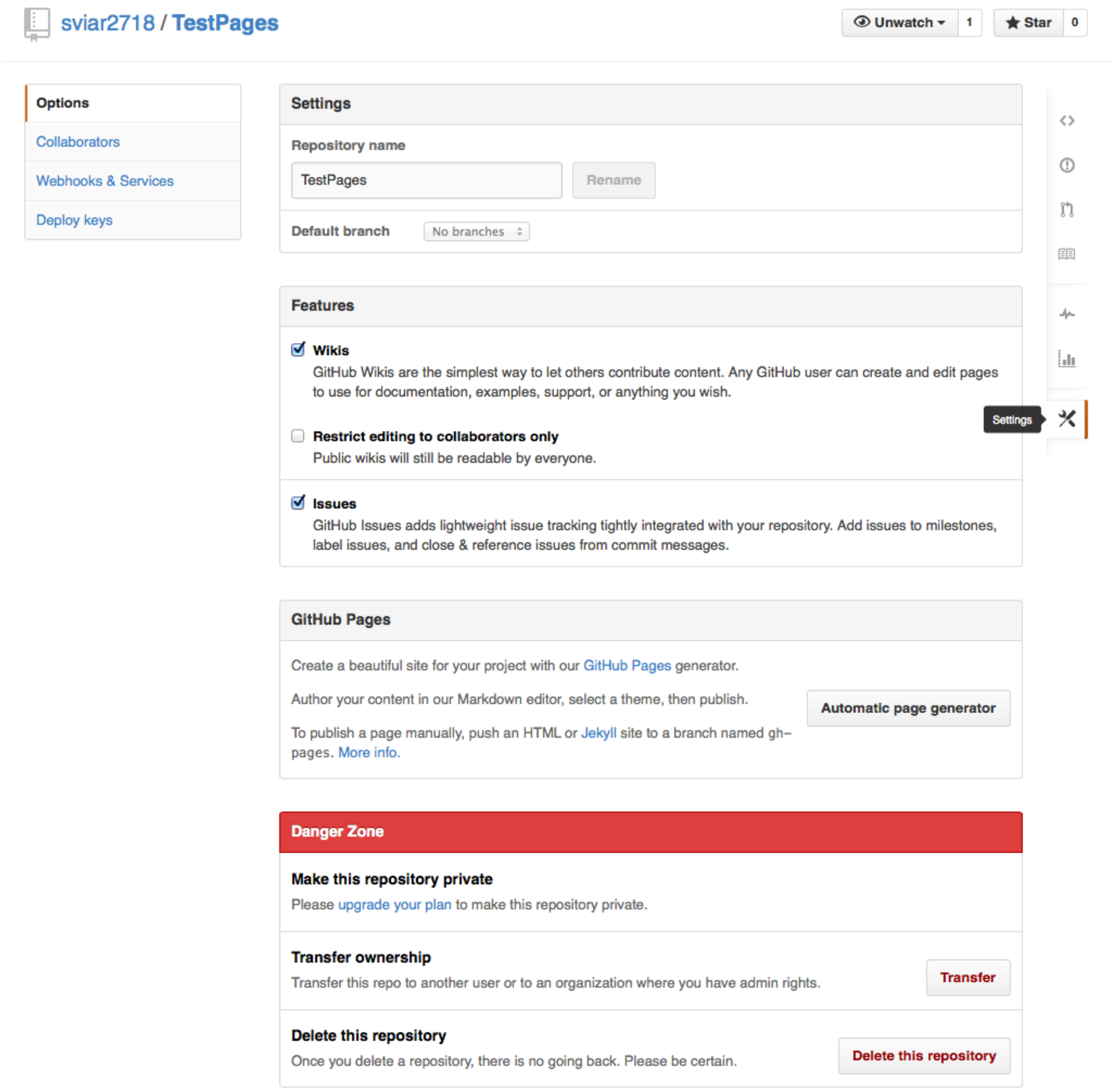
Import code from an old repository

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

On the right hand side,
click on Settings.



You should see:



Settings

Repository name: TestPages

Default branch: No branches

Features

☑ Wikis
GitHub Wikis are the simplest way to let others contribute content. Any GitHub user can create and edit pages to use for documentation, examples, support, or anything you wish.

□ Restrict editing to collaborators only
Public wikis will still be readable by everyone.

☑ Issues
GitHub Issues adds lightweight issue tracking tightly integrated with your repository. Add issues to milestones, label issues, and close & reference issues from commit messages.

GitHub Pages

Create a beautiful site for your project with our GitHub Pages generator.

Author your content in our Markdown editor, select a theme, then publish.

To publish a page manually, push an HTML or Jekyll site to a branch named gh-pages. More info.

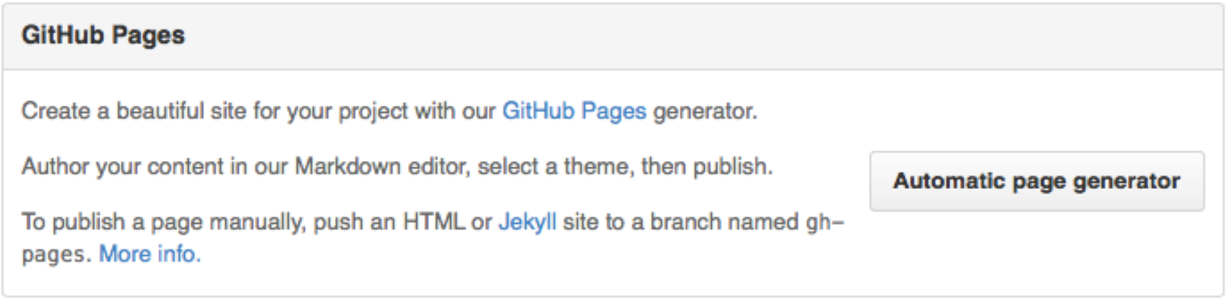
Danger Zone

Make this repository private
Please upgrade your plan to make this repository private.

Transfer ownership
Transfer this repo to another user or to an organization where you have admin rights.

Delete this repository
Once you delete a repository, there is no going back. Please be certain.

Click on the “Automatic page generator”:



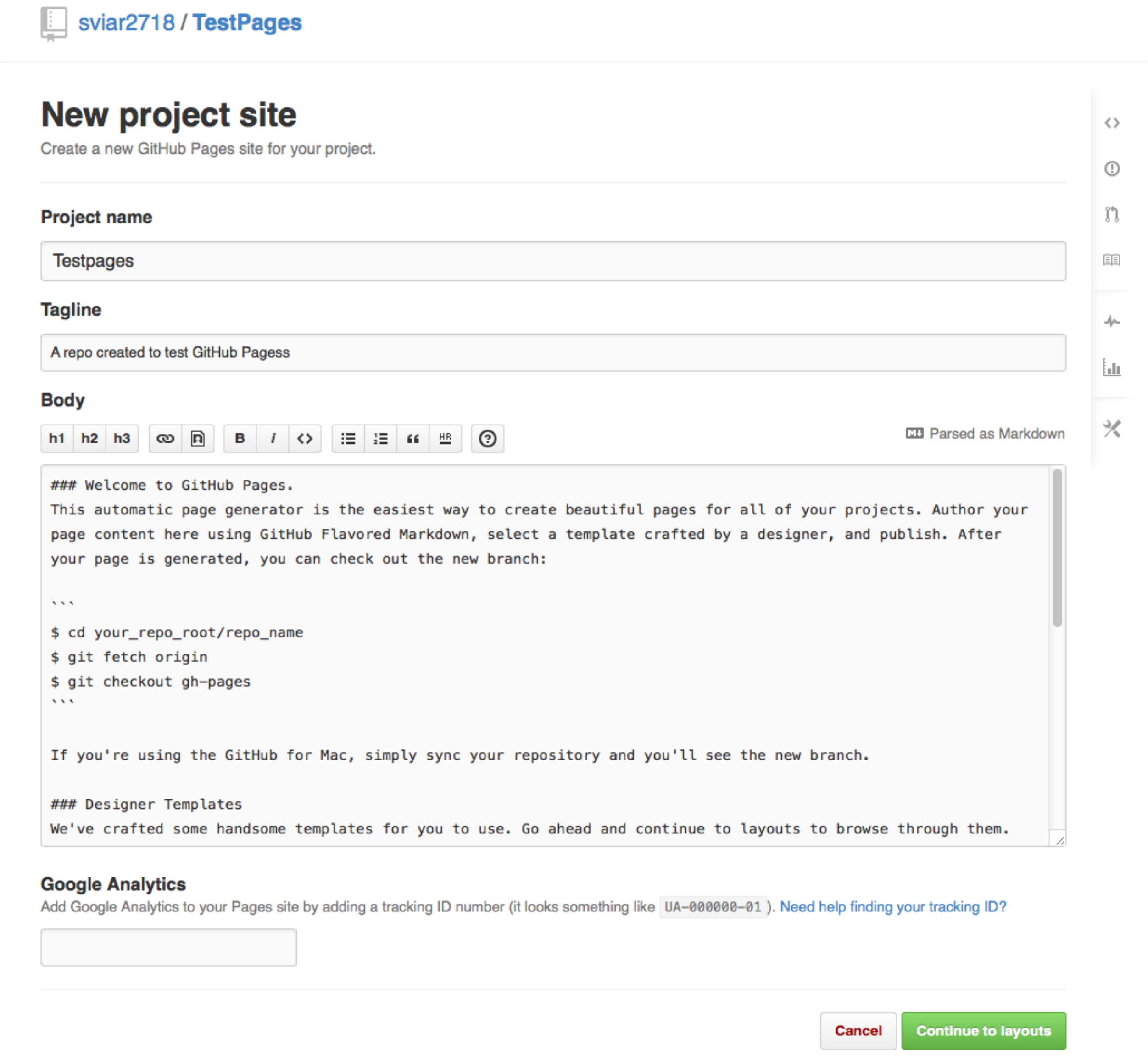
GitHub Pages

Create a beautiful site for your project with our GitHub Pages generator.

Author your content in our Markdown editor, select a theme, then publish.

To publish a page manually, push an HTML or Jekyll site to a branch named gh-pages. More info.

You will see:



sviar2718 / TestPages

New project site

Create a new GitHub Pages site for your project.

Project name

Testpages

Tagline

A repo created to test GitHub Pagess

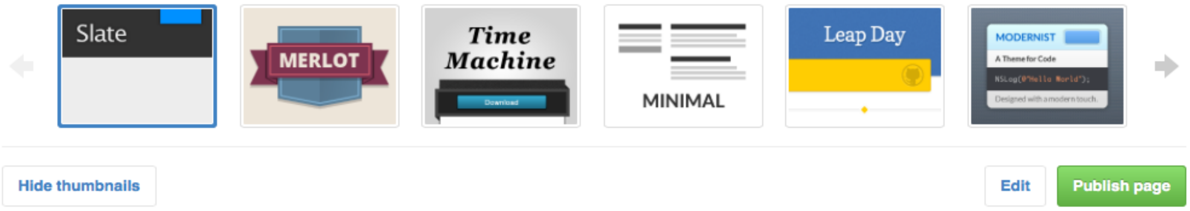
Body

```
### Welcome to GitHub Pages.
This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your
page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After
your page is generated, you can check out the new branch:

```
$ cd your_repo_root/repo_name
$ git fetch origin
$ git checkout gh-pages
```

If you're using the GitHub for Mac, simply sync your repository and you'll see the new branch.

Designer Templates
We've crafted some handsome templates for you to use. Go ahead and continue to layouts to browse through them.
```

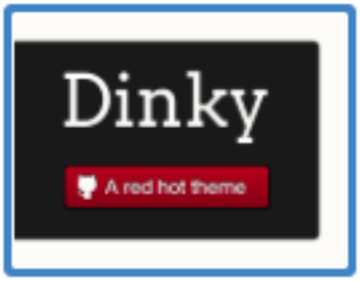
Google Analytics

Add Google Analytics to your Pages site by adding a tracking ID number (it looks something like `UA-000000-01`). Need help finding your tracking ID?

Now click on “Continue to layouts”:

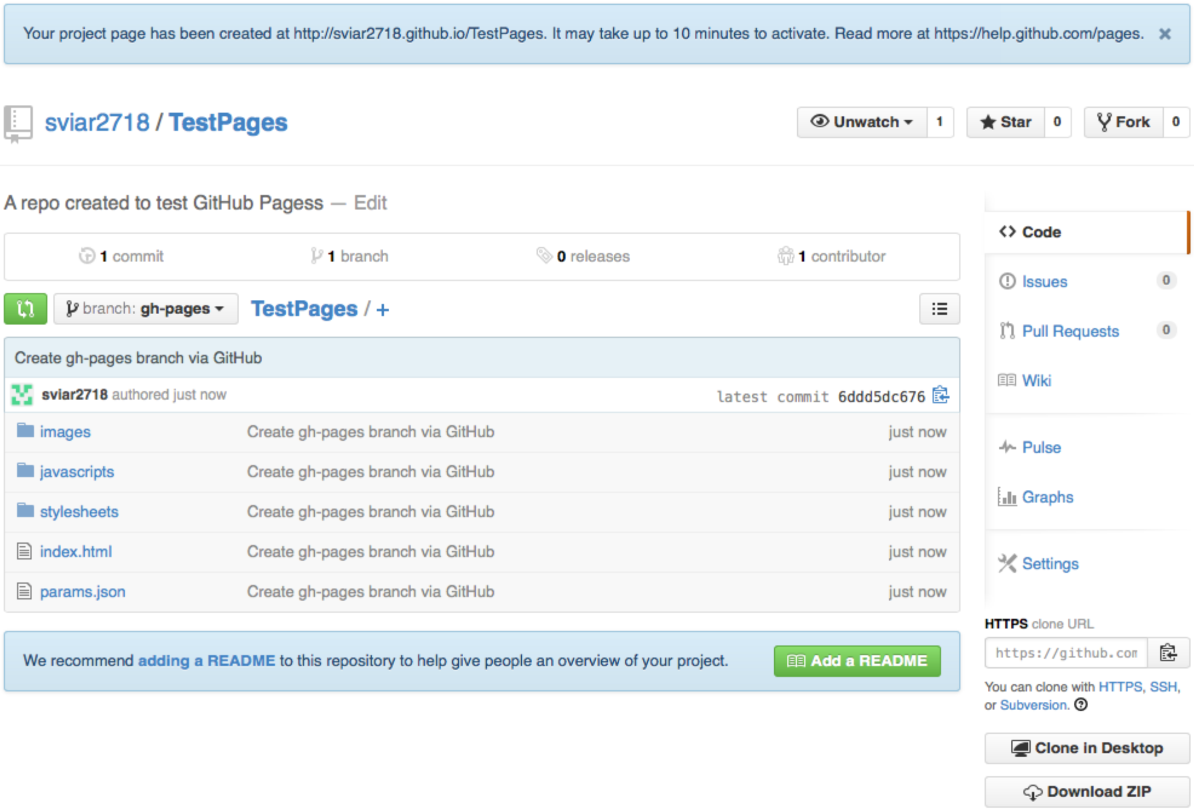


You need to pick a “theme” from one of the ones you see above. You can use the right arrow to go all the way to the right. If you want to do what I did to match “DanWeb”, you should choose “Dinky”:



Now click on “Publish page”

Now you will see:



Your project page has been created at http://sviar2718.github.io/TestPages. It may take up to 10 minutes to activate. Read more at https://help.github.com/pages.

sviar2718 / **TestPages**

A repo created to test GitHub Pagess — Edit

1 commit · **1** branch · **0** releases · **1** contributor

Create gh-pages branch via GitHub

sviar2718 authored just now · latest commit 6ddd5dc676

images	Create gh-pages branch via GitHub	just now
javascripts	Create gh-pages branch via GitHub	just now
stylesheets	Create gh-pages branch via GitHub	just now
index.html	Create gh-pages branch via GitHub	just now
params.json	Create gh-pages branch via GitHub	just now

We recommend adding a README to this repository to help give people an overview of your project.

HTTPS clone URL

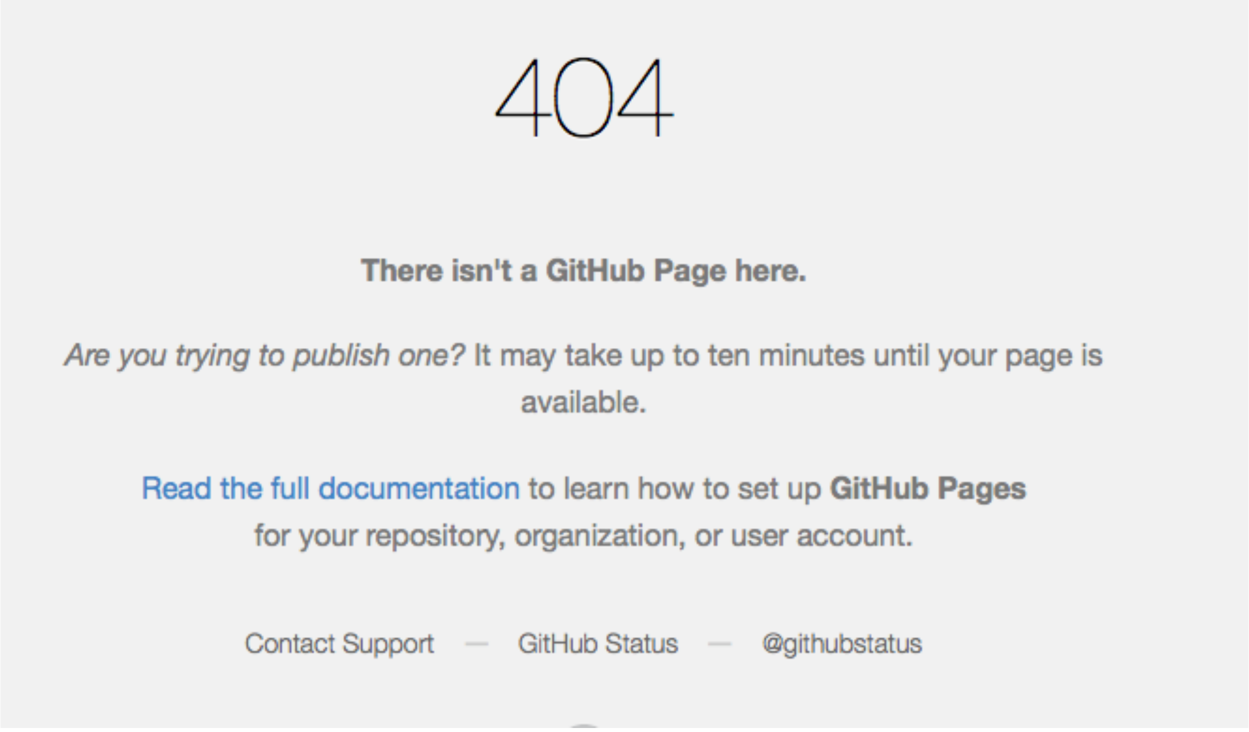
https://github.com

You can clone with HTTPS, SSH, or Subversion.

Notice that it says
“Your project page has been created at http://sviar2718.github.io/TestPages. It may take up to 10 minutes to activate. Read more at https://help.github.com/pages.”
at the top.

You can copy this to another tab in your browser to see what you’re website looks like.

It can take up to 10 minutes. While you wait, you may see the following at the url:



404

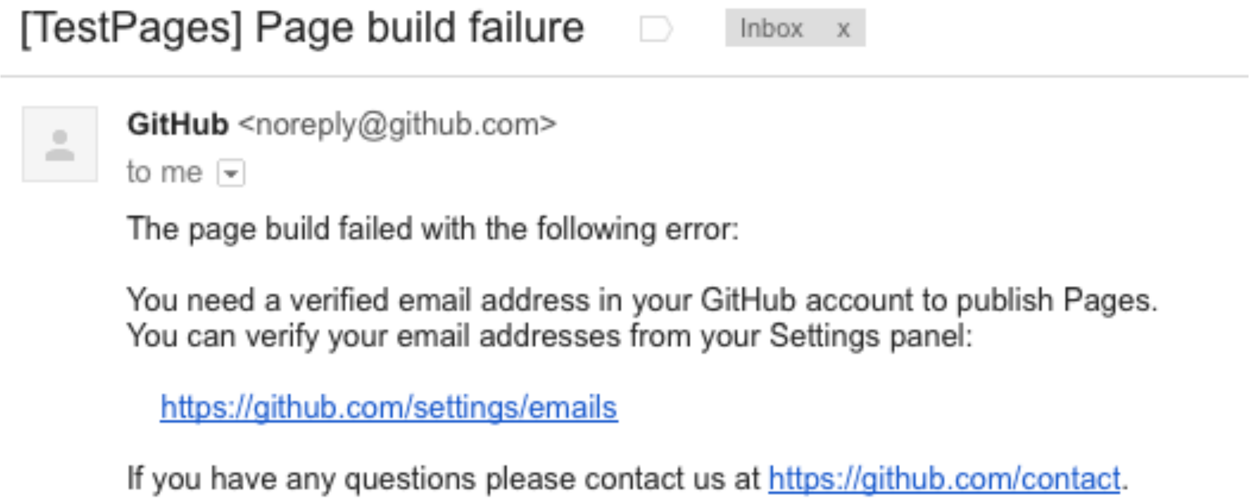
There isn't a GitHub Page here.

Are you trying to publish one? It may take up to ten minutes until your page is available.

Read the full documentation to learn how to set up **GitHub Pages** for your repository, organization, or user account.

Contact Support — GitHub Status — @githubstatus

IMPORTANT NOTE: You will need to have a verified email address for this to work. If you see that this is taking a long time to work, then check your email address to see if you have an email like the following:



[TestPages] Page build failure

GitHub <noreply@github.com>
to me

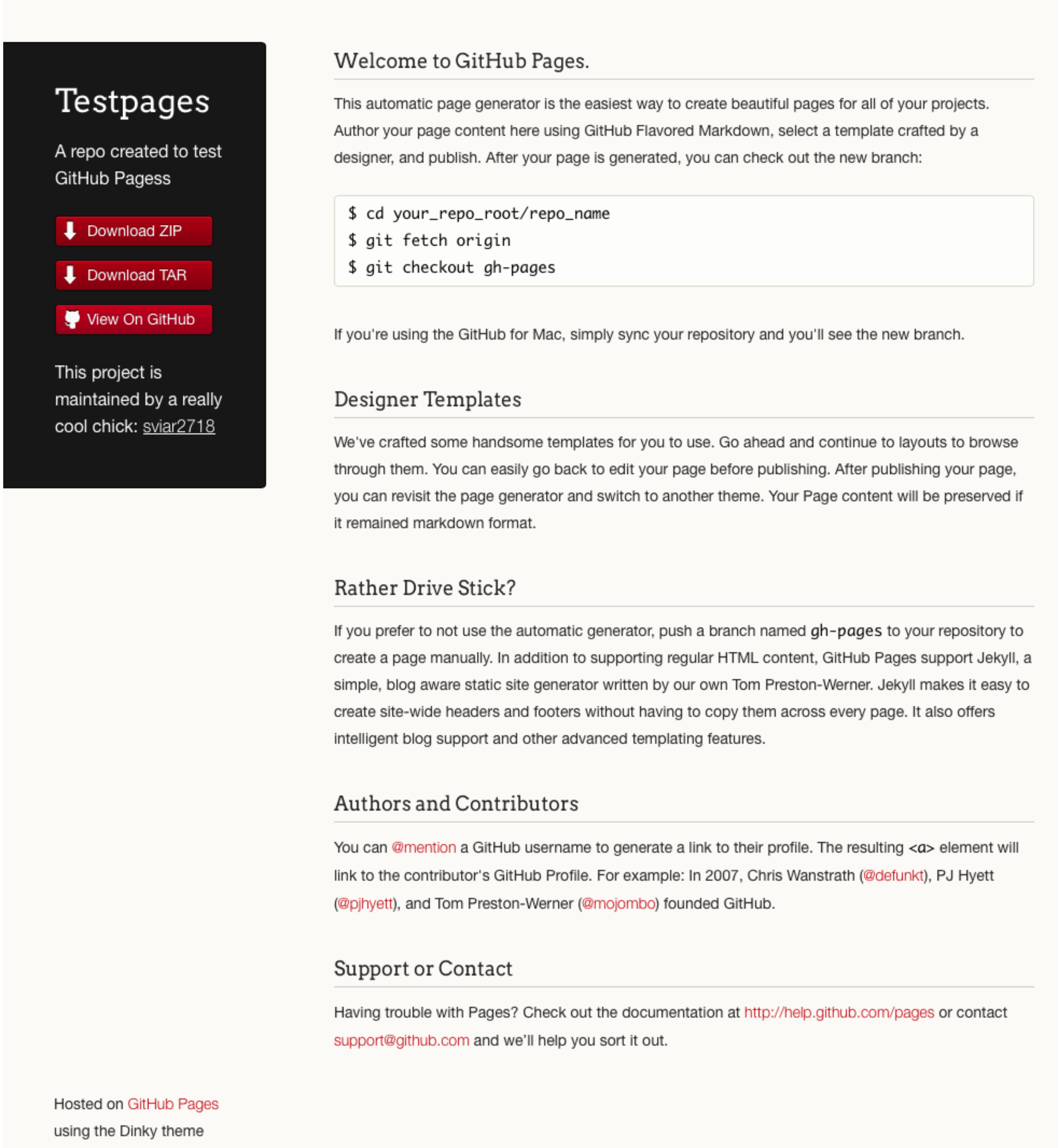
The page build failed with the following error:

You need a verified email address in your GitHub account to publish Pages.
You can verify your email addresses from your Settings panel:

https://github.com/settings/emails

If you have any questions please contact us at https://github.com/contact.

Eventually, you should see a screen like the following:



Testpages

A repo created to test GitHub Pagess

This project is maintained by a really cool chick: sviar2718

Welcome to GitHub Pages.

This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:

```
$ cd your_repo_root/repo_name
$ git fetch origin
$ git checkout gh-pages
```

If you're using the GitHub for Mac, simply sync your repository and you'll see the new branch.

Designer Templates

We've crafted some handsome templates for you to use. Go ahead and continue to layouts to browse through them. You can easily go back to edit your page before publishing. After publishing your page, you can revisit the page generator and switch to another theme. Your Page content will be preserved if it remained markdown format.

Rather Drive Stick?

If you prefer to not use the automatic generator, push a branch named `gh-pages` to your repository to create a page manually. In addition to supporting regular HTML content, GitHub Pages support Jekyll, a simple, blog aware static site generator written by our own Tom Preston-Werner. Jekyll makes it easy to create site-wide headers and footers without having to copy them across every page. It also offers intelligent blog support and other advanced templating features.

Authors and Contributors

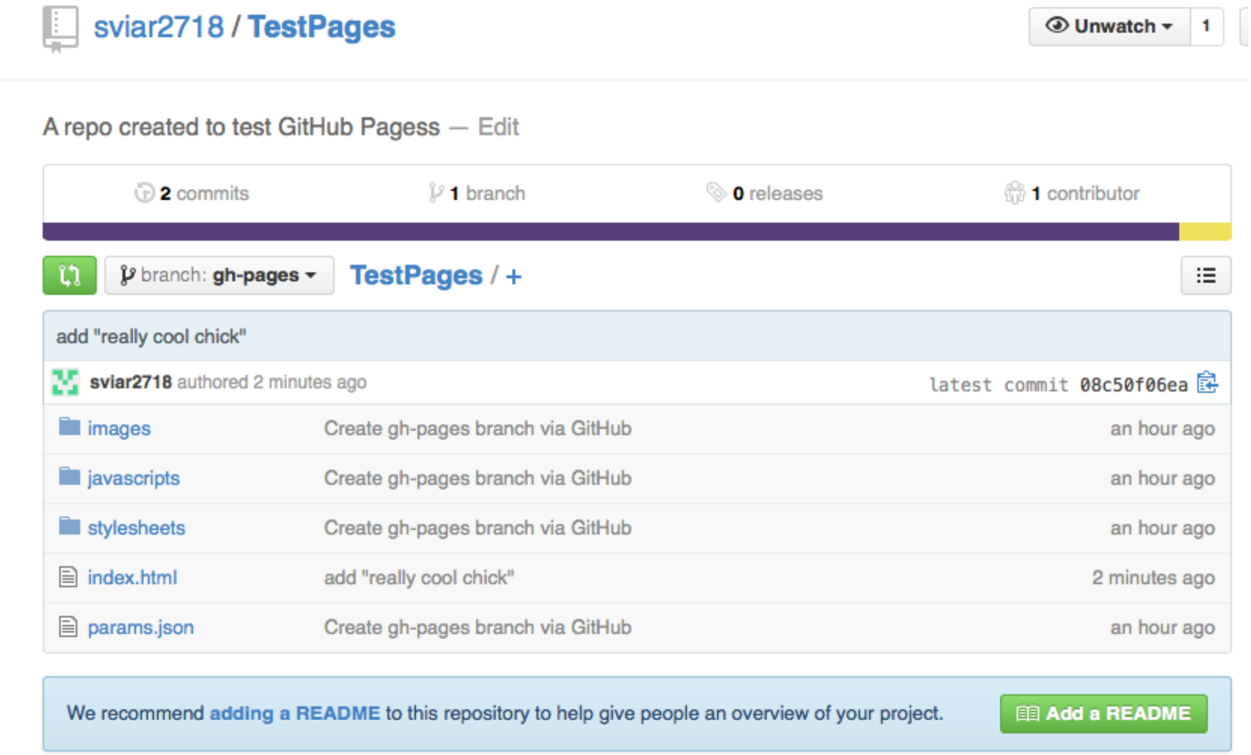
You can @mention a GitHub username to generate a link to their profile. The resulting `<a>` element will link to the contributor's GitHub Profile. For example: In 2007, Chris Wanstrath (@defunkt), PJ Hyett (@pjhyett), and Tom Preston-Werner (@mojombo) founded GitHub.

Support or Contact

Having trouble with Pages? Check out the documentation at http://help.github.com/pages or contact support@github.com and we'll help you sort it out.

Hosted on GitHub Pages using the Dinky theme

Let's go back to the GitHub website and edit some content:



sviar2718 / TestPages

A repo created to test GitHub Pagess — Edit

2 commits | 1 branch | 0 releases | 1 contributor

TestPages / +

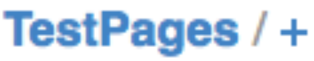
add "really cool chick"

sviar2718 authored 2 minutes ago latest commit 08c50f06ea

images	Create gh-pages branch via GitHub	an hour ago
javascripts	Create gh-pages branch via GitHub	an hour ago
stylesheets	Create gh-pages branch via GitHub	an hour ago
index.html	add "really cool chick"	2 minutes ago
params.json	Create gh-pages branch via GitHub	an hour ago

We recommend adding a README to this repository to help give people an overview of your project.

Click on the little “+” next to the name of your repo:

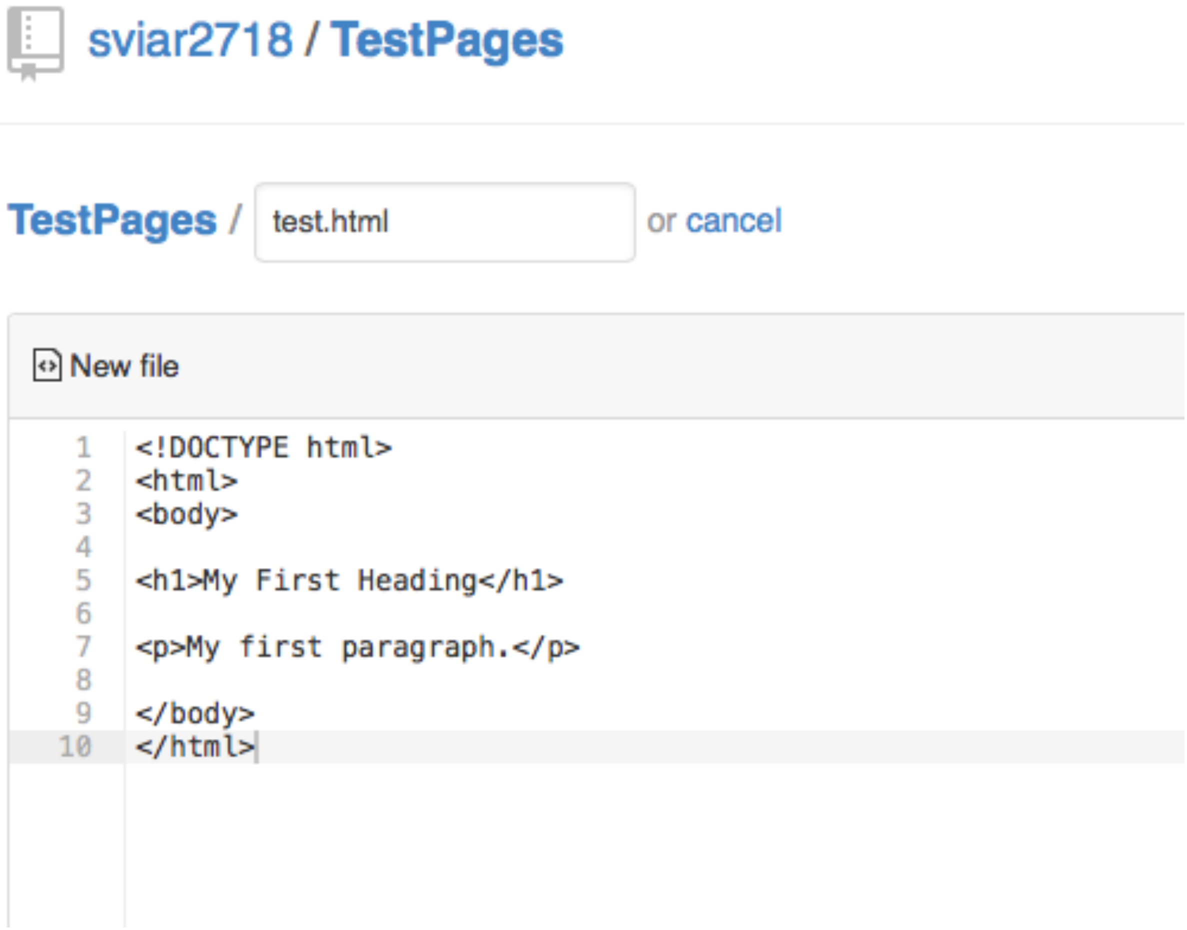


This will allow you to create a new file. Call the new file “test.html”. Now navigate to the following url (in another tab): http://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

You should see the following code:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

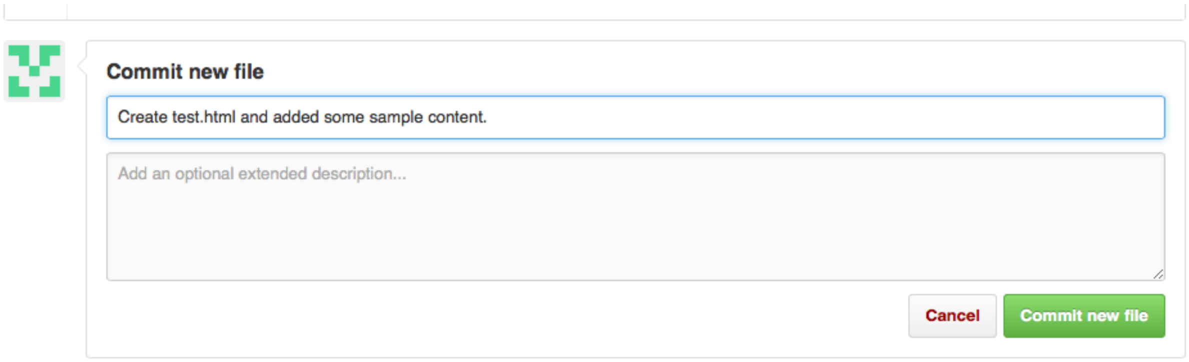
Copy and paste this into your new file. Your document should look like the following:



TestPages / test.html

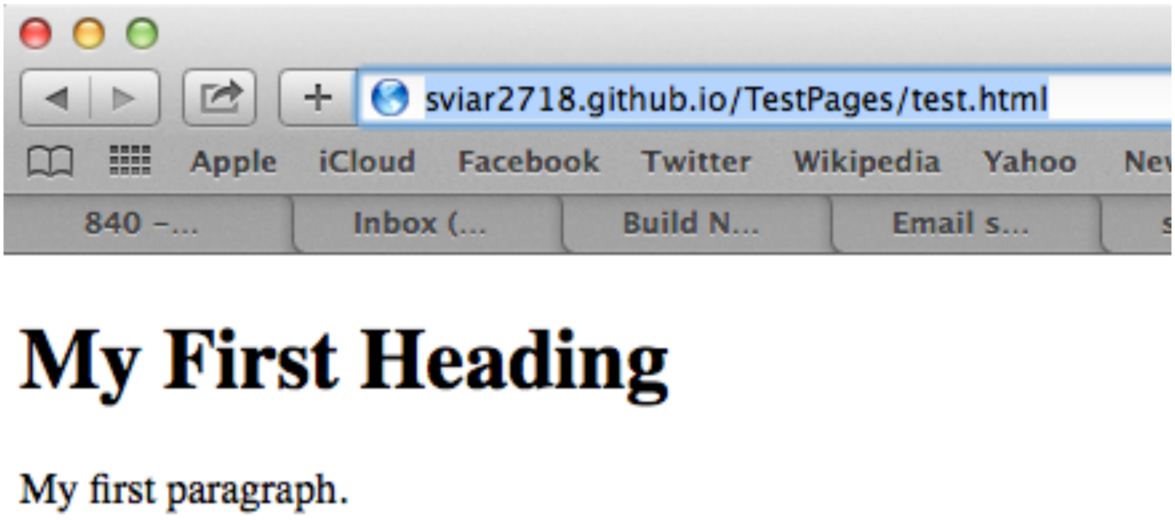
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Now add a comment for your new file and commit:



Create test.html and added some sample content.

Give it a little time and try to navigate to your test file:

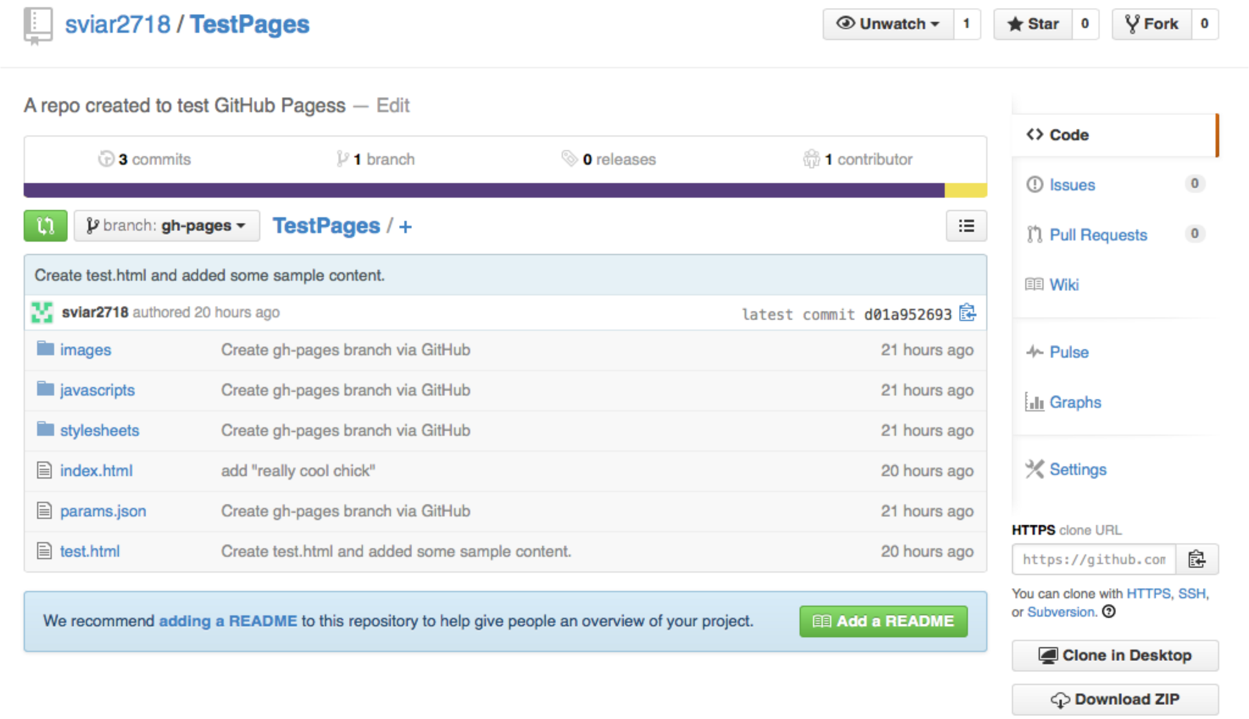


My First Heading

My first paragraph.

Congrats! You just edited your first webpage! Note that this is available to the entire internet! Now you can continue to edit your repo online, but an easier way is to clone your repo to your local computer, make changes, commit your changes and then push your changes back to GitHub. Let's do an example (this will only work if you have a PC or Mac, if you have Linux, you will need to use the command line).

On the GitHub page for your repo:



A repo created to test GitHub Pagess — Edit

Create test.html and added some sample content.

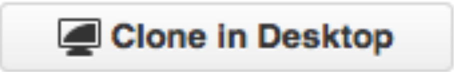
sviar2718 authored 20 hours ago latest commit d01a952693

images	Create gh-pages branch via GitHub	21 hours ago
javascripts	Create gh-pages branch via GitHub	21 hours ago
stylesheets	Create gh-pages branch via GitHub	21 hours ago
index.html	add "really cool chick"	20 hours ago
params.json	Create gh-pages branch via GitHub	21 hours ago
test.html	Create test.html and added some sample content.	20 hours ago

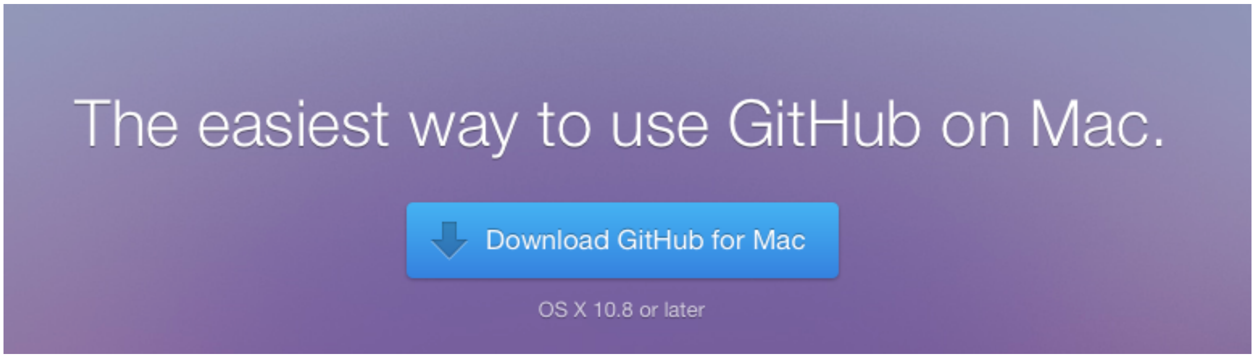
We recommend adding a README to this repository to help give people an overview of your project.

You can clone with HTTPS, SSH, or Subversion.

Click on “Clone in Desktop”:

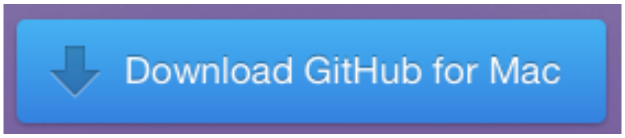


If you haven’t installed the GitHub client you should see:

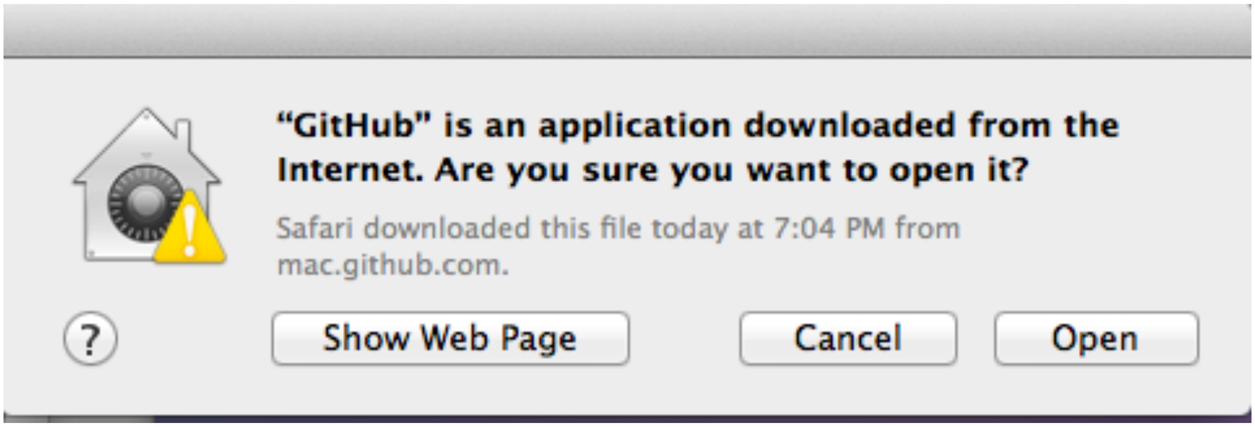


(Windows users should see something similar)

Now click on the big blue button and install the GitHub client for Mac.

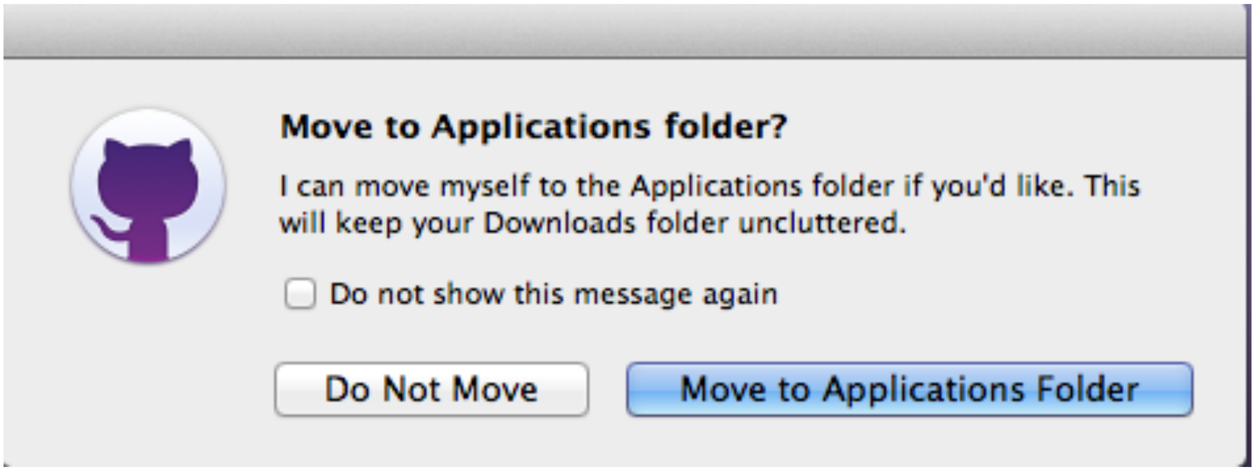


On a Mac, you will see:


“GitHub” is an application downloaded from the Internet. Are you sure you want to open it?

Safari downloaded this file today at 7:04 PM from mac.github.com.

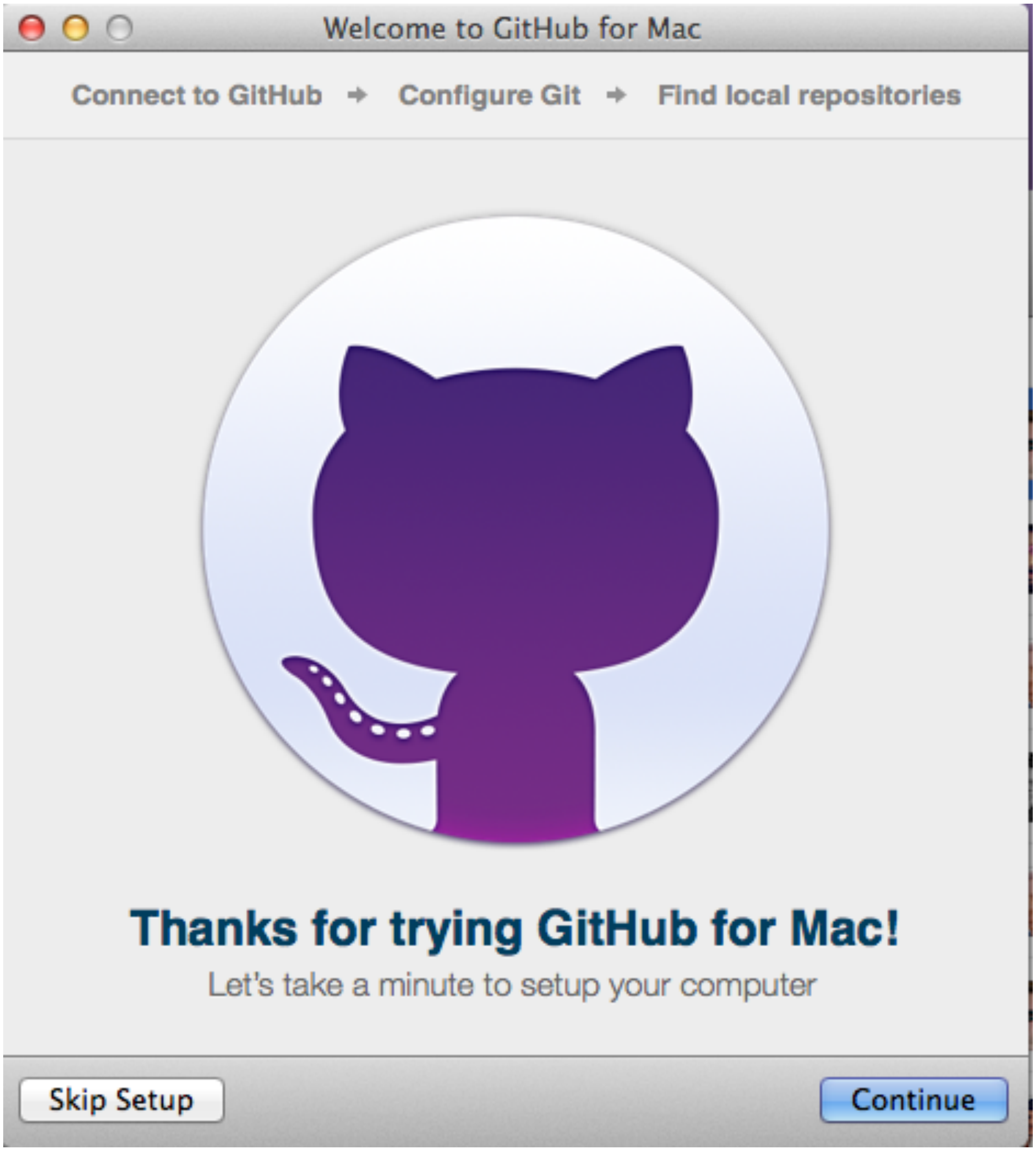
Click on “Open”


Move to Applications folder?

I can move myself to the Applications folder if you'd like. This will keep your Downloads folder uncluttered.

□ Do not show this message again

Now click on “Move to Applications Folder”

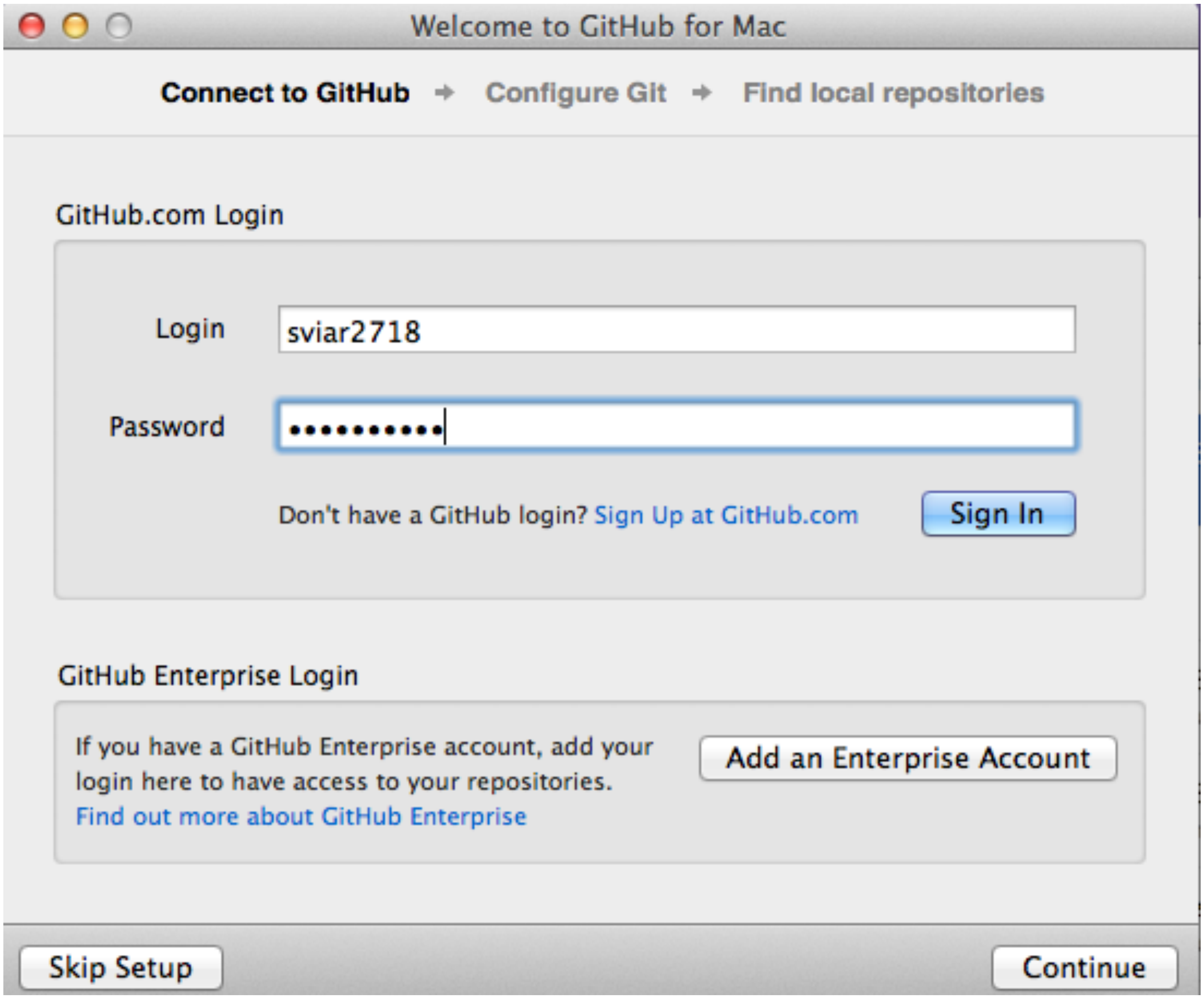


Welcome to GitHub for Mac

Thanks for trying GitHub for Mac!

Let's take a minute to setup your computer

Now click on “Continue” and sign in with your user name and password:



Welcome to GitHub for Mac
Connect to GitHub → Configure Git → Find local repositories
Sign In
Add an Enterprise Account
Skip Setup
Continue

GitHub.com Login

Login: sviar2718

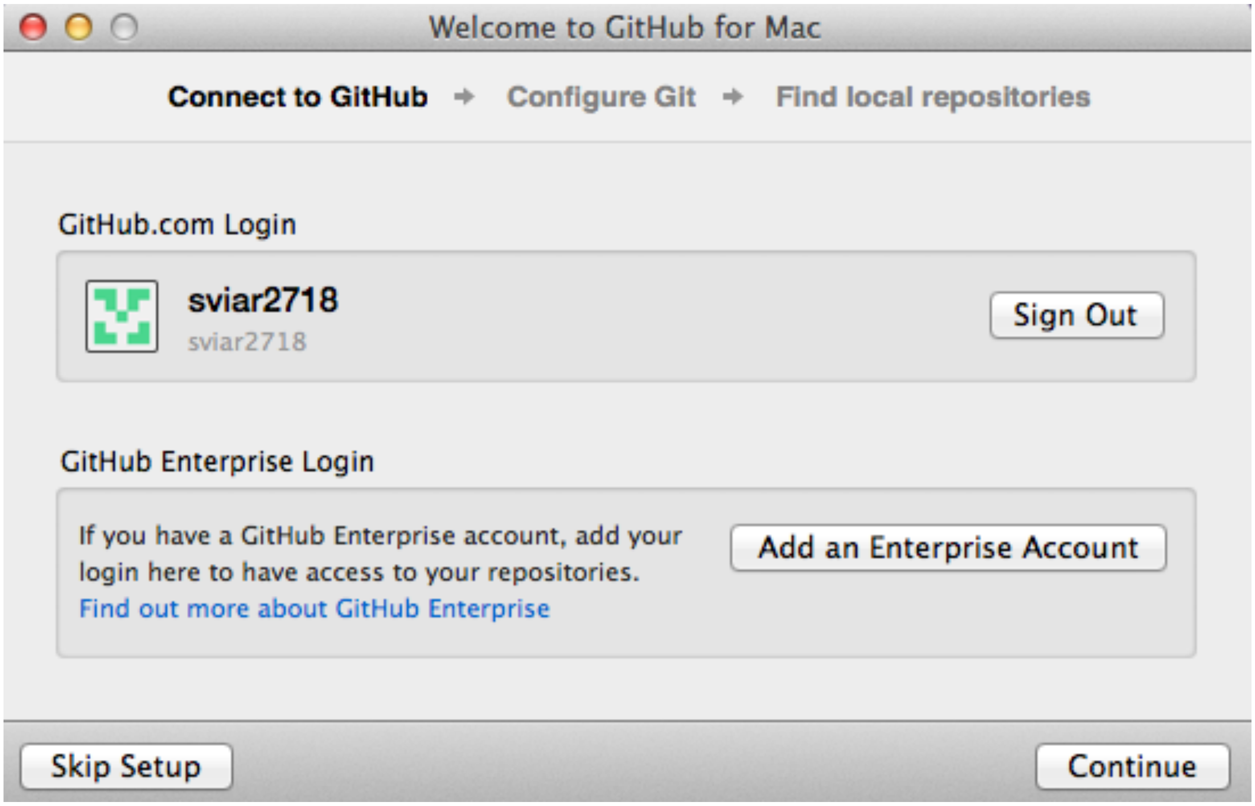
Password: ••••••••••

Don't have a GitHub login? Sign Up at GitHub.com

GitHub Enterprise Login

If you have a GitHub Enterprise account, add your login here to have access to your repositories.
Find out more about GitHub Enterprise

Click on “Continue”


Welcome to GitHub for Mac
Connect to GitHub → Configure Git → Find local repositories

GitHub.com Login

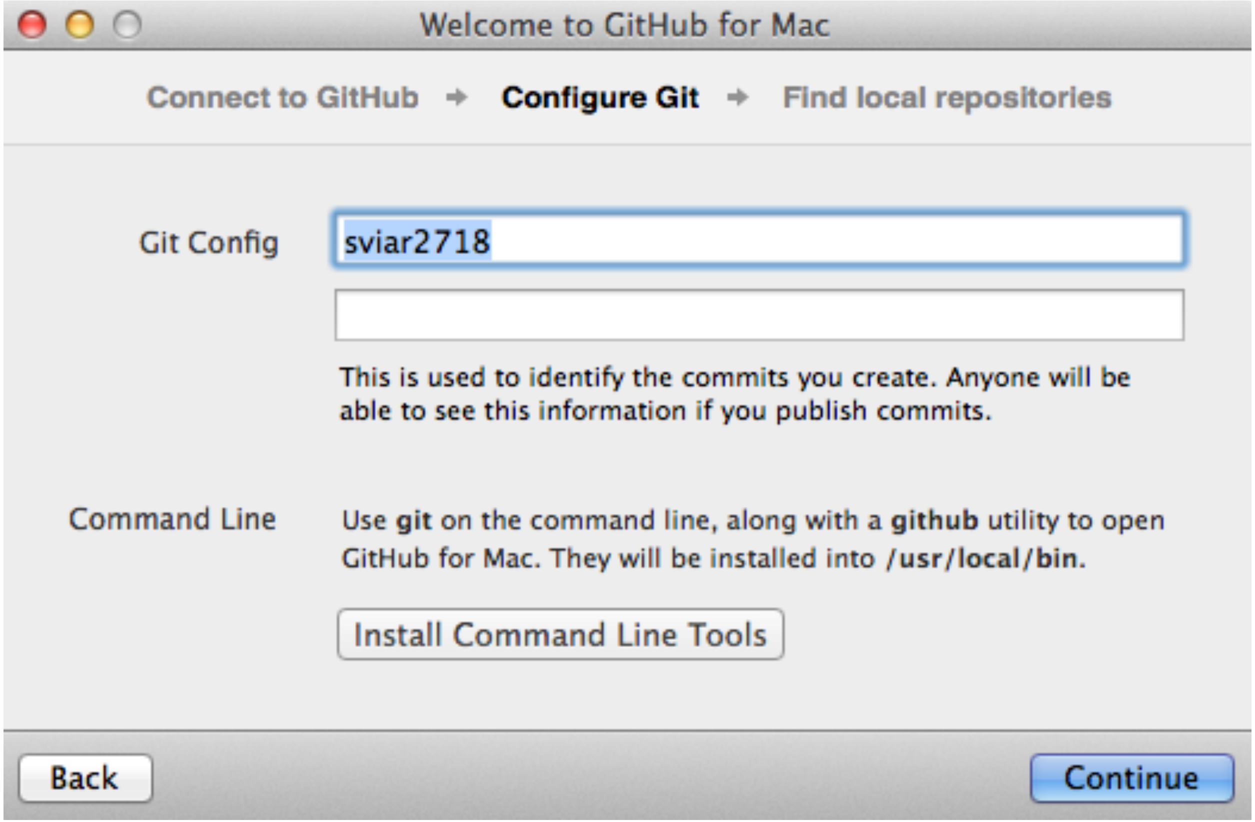
sviar2718
sviar2718

GitHub Enterprise Login

If you have a GitHub Enterprise account, add your login here to have access to your repositories.
Find out more about GitHub Enterprise

Add an Enterprise Account
Skip Setup
Continue

Given the option, you should install the command line tools (click the button).


Welcome to GitHub for Mac
Connect to GitHub → Configure Git → Find local repositories

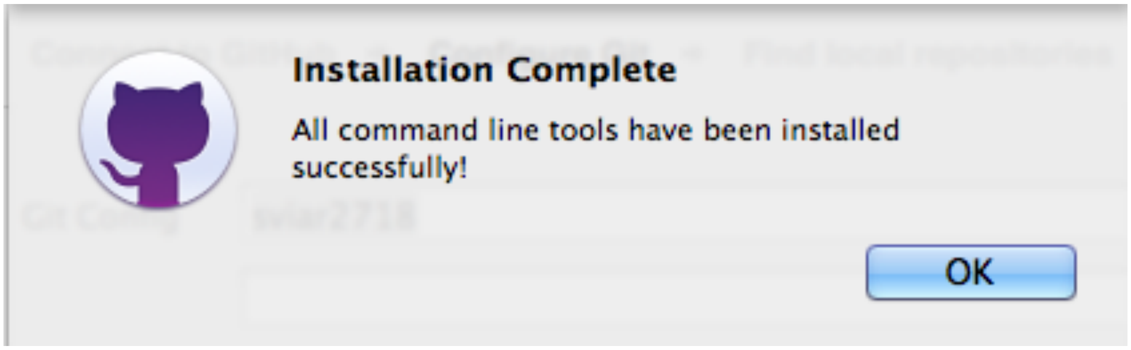
Git Config: sviar2718

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Command Line

Use **git** on the command line, along with a **github** utility to open GitHub for Mac. They will be installed into **/usr/local/bin**.

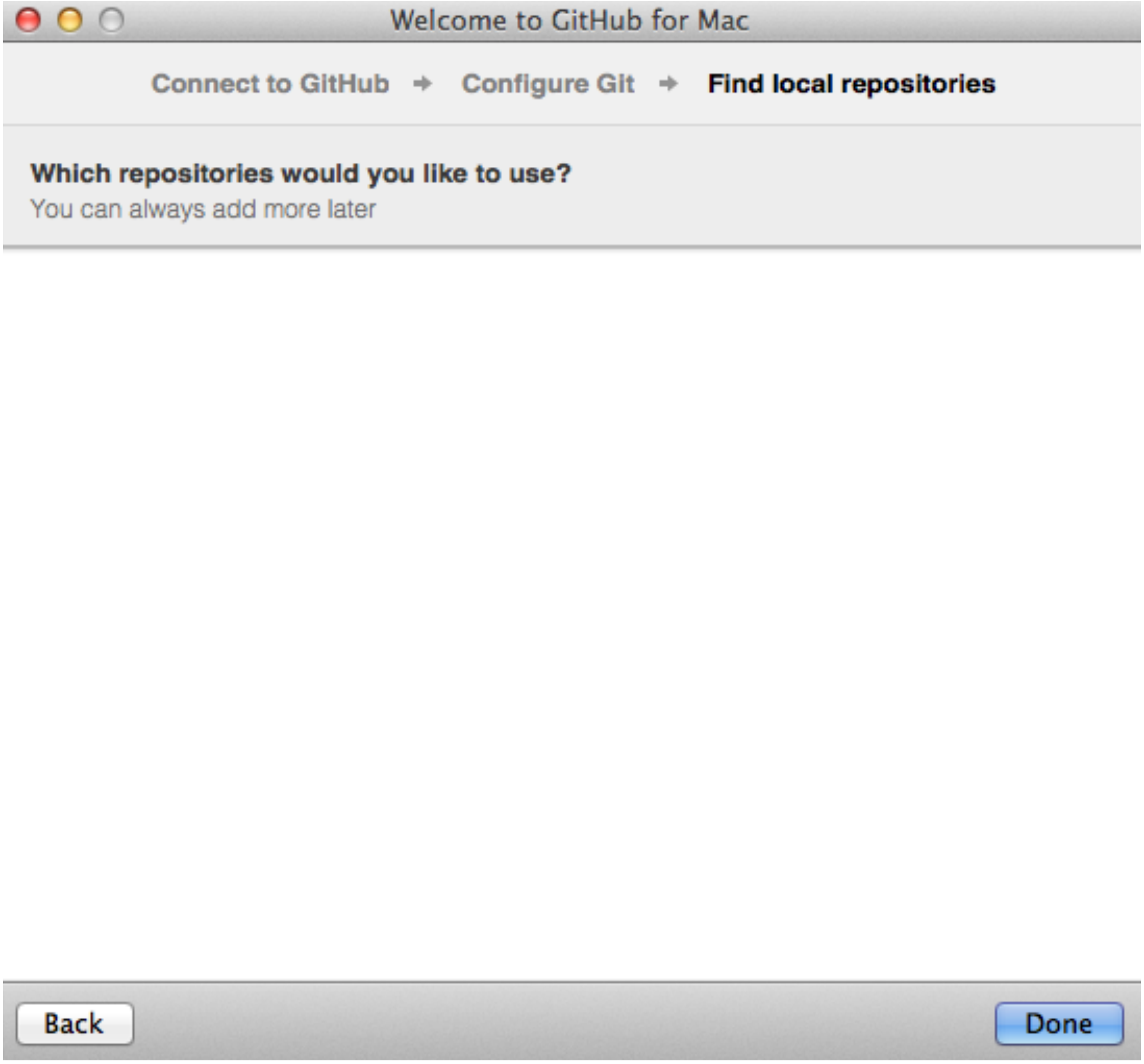
Install Command Line Tools
Back
Continue



Installation Complete

All command line tools have been installed successfully!

Continue to click “Continue” and then “Done”



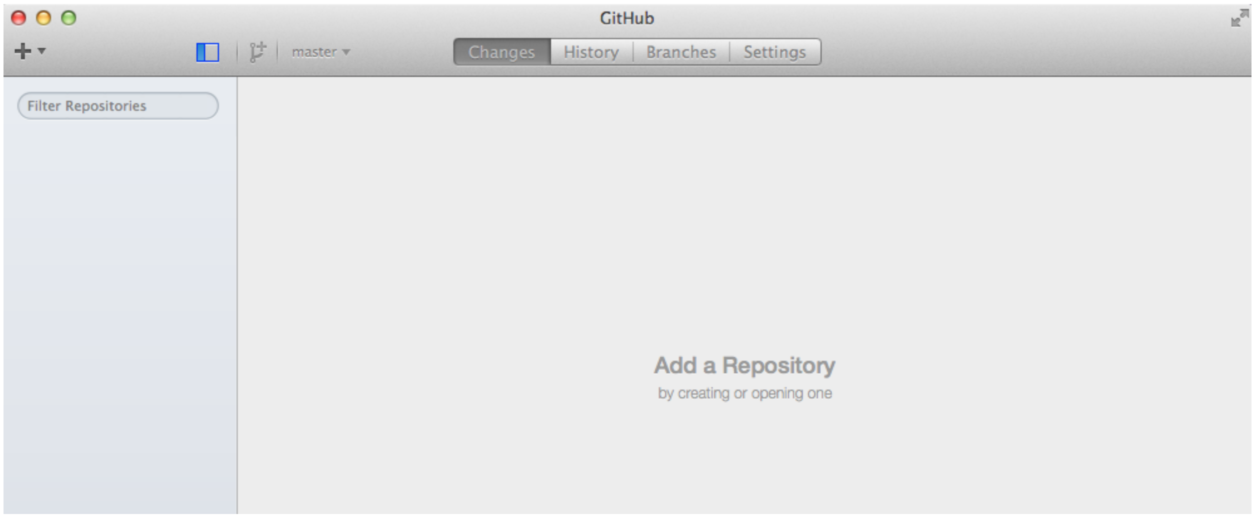
Welcome to GitHub for Mac

Connect to GitHub → Configure Git → **Find local repositories**

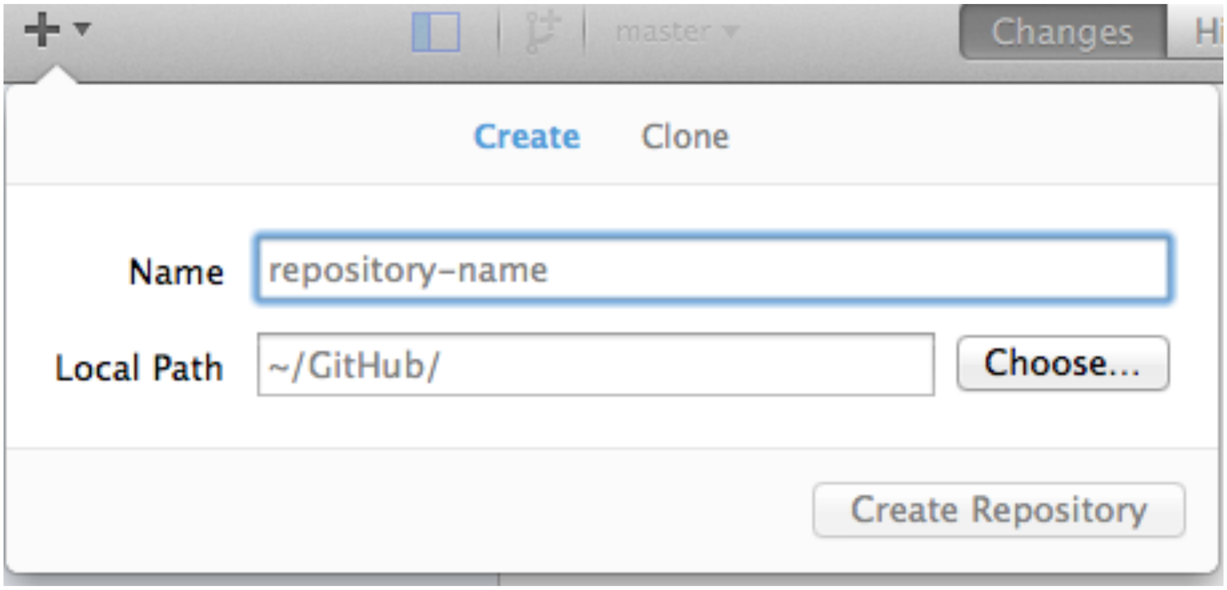
Which repositories would you like to use?

You can always add more later

Now you should see something like:



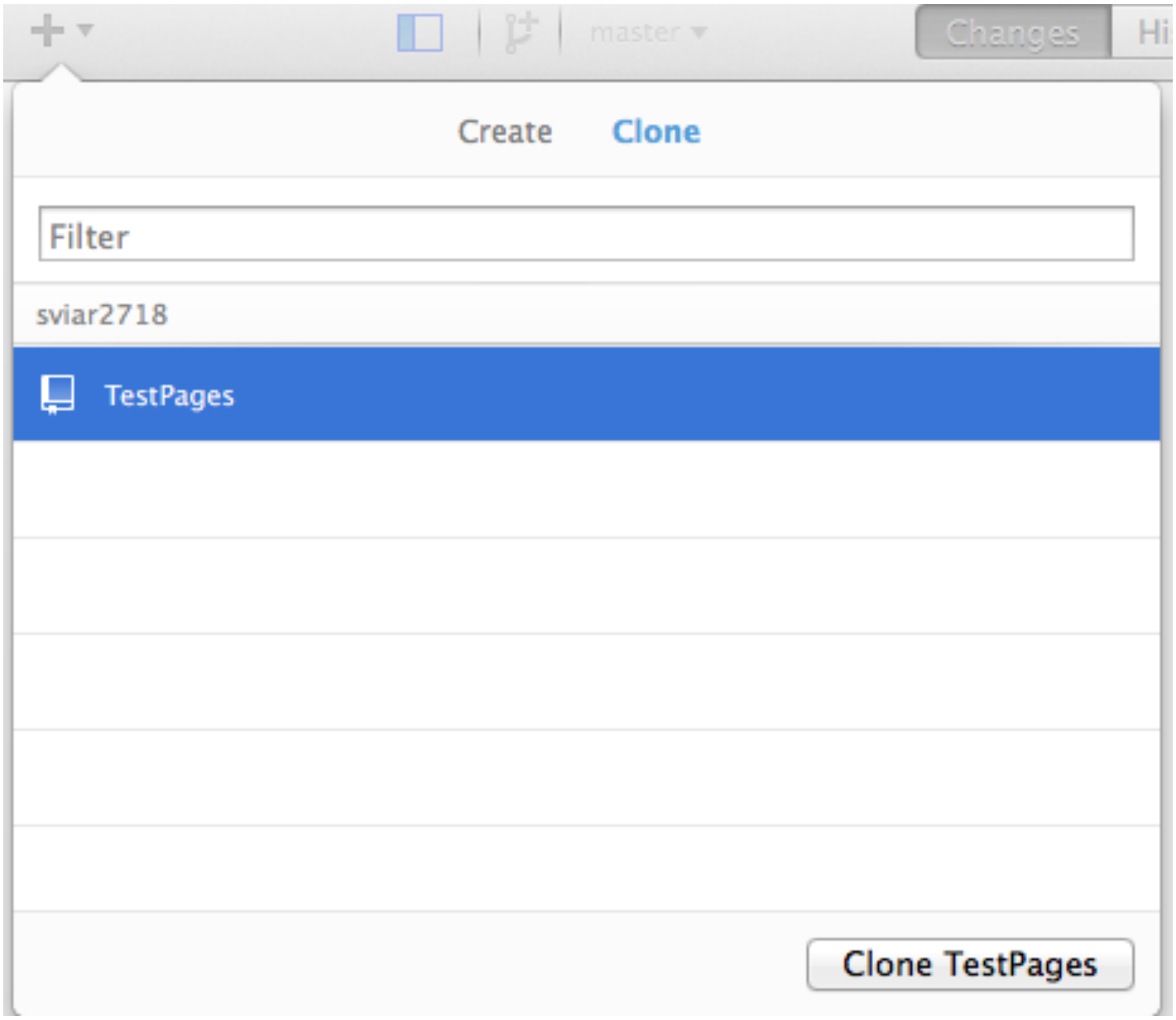
We want to add the repository that we created earlier. To do this, click on the "+"



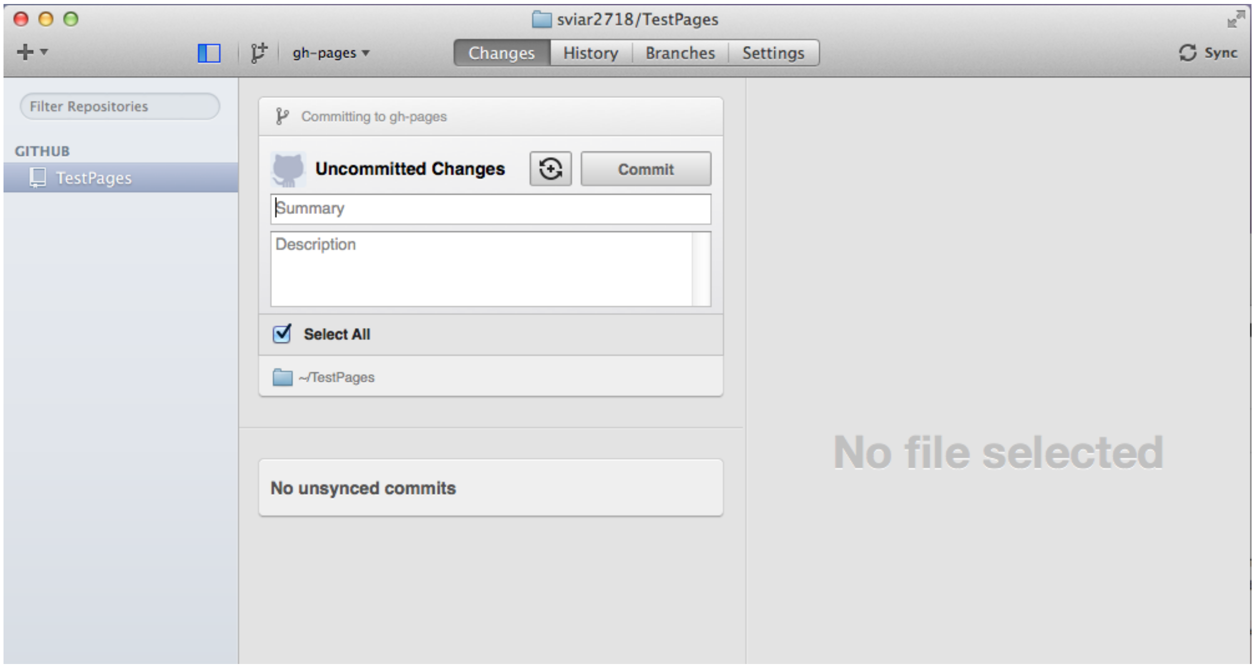
Name: repository-name

Local Path: ~/GitHub/

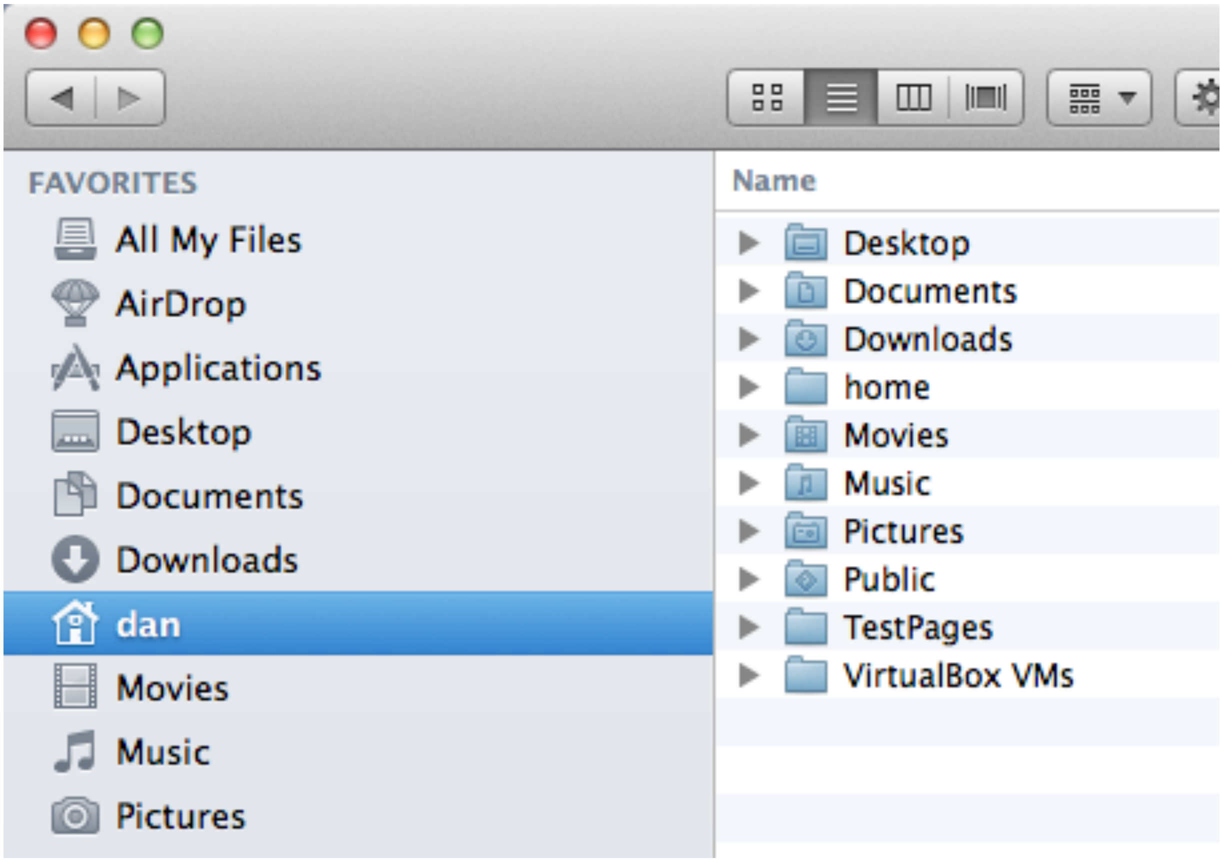
And then click on "Clone" to see the repo that you created earlier. Click on it and then then select "Clone"



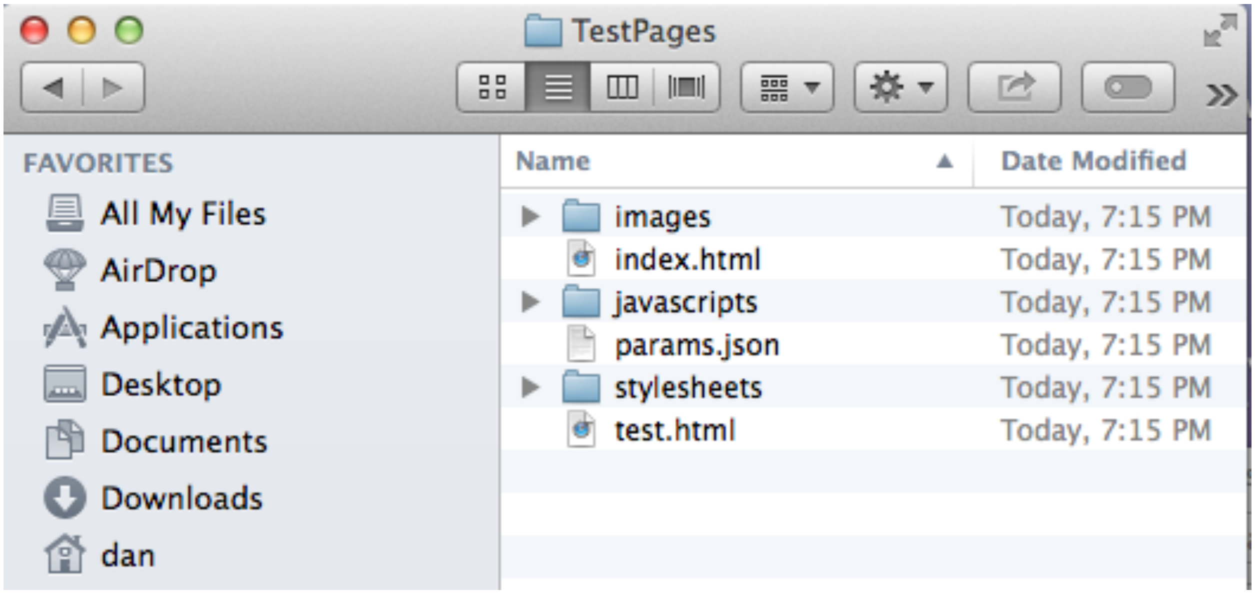
You should see something like this:



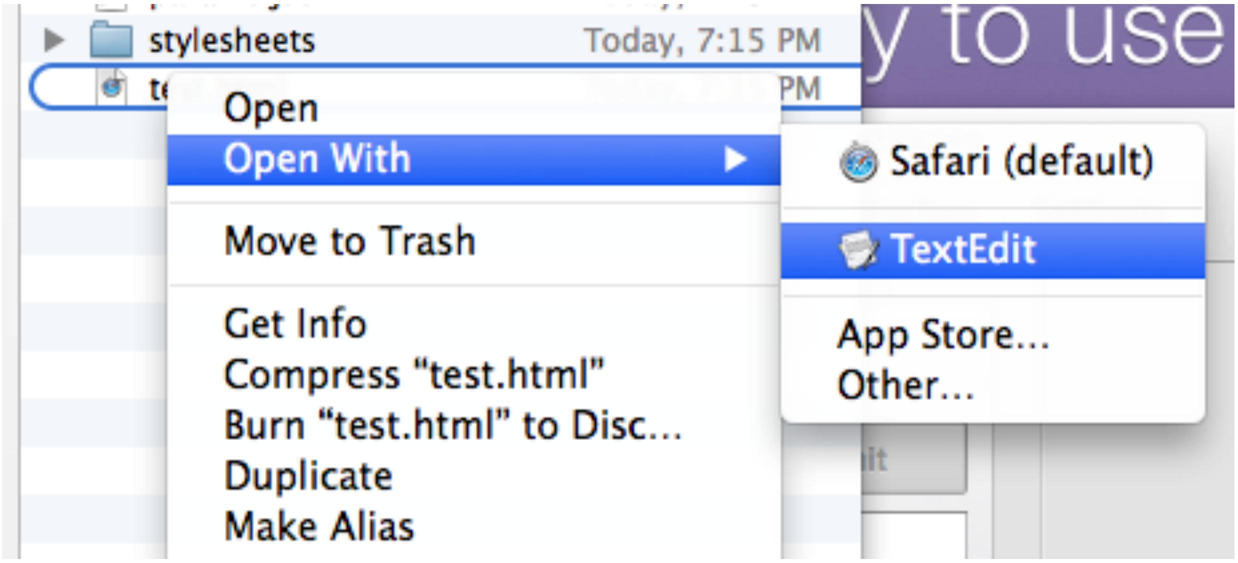
Now you need to find the folder that GitHub downloaded your repo to. On my Mac that was dan/TestPages:



Let’s go into that folder:

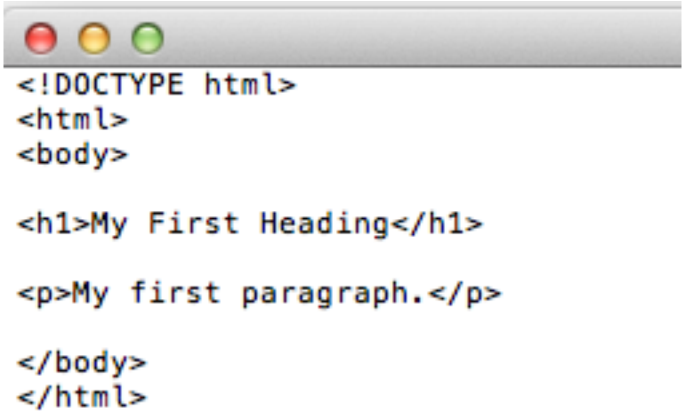


Using any text editor on your system, let’s make a change to test.html:



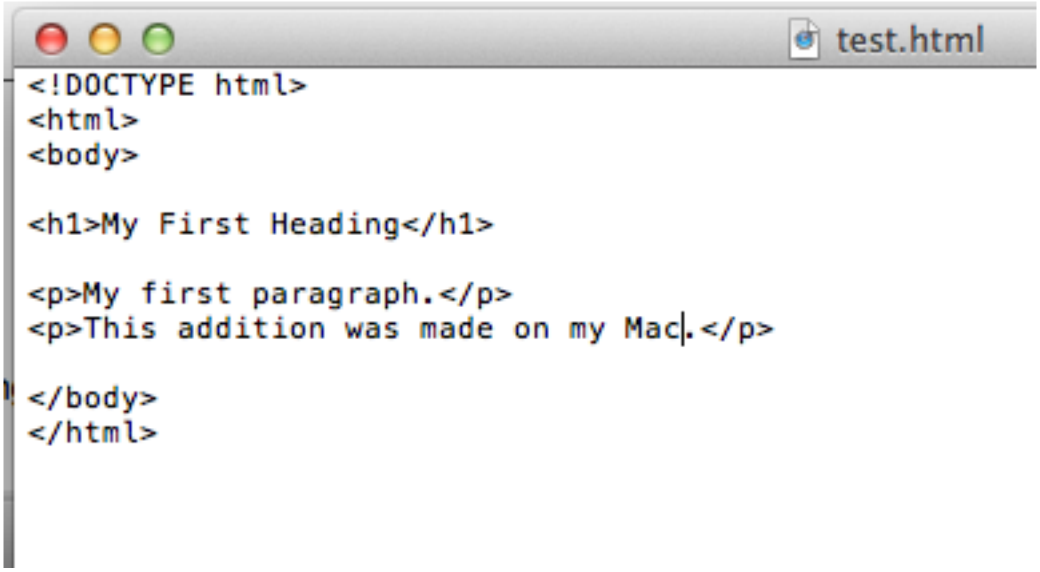
NOTE: On a Mac, if you want to use TextEdit, you may have to change preferences for TextEdit (see for example http://support.apple.com/kb/TA20406?viewlocale=en_US&locale=en_US)

You should see something like the following:



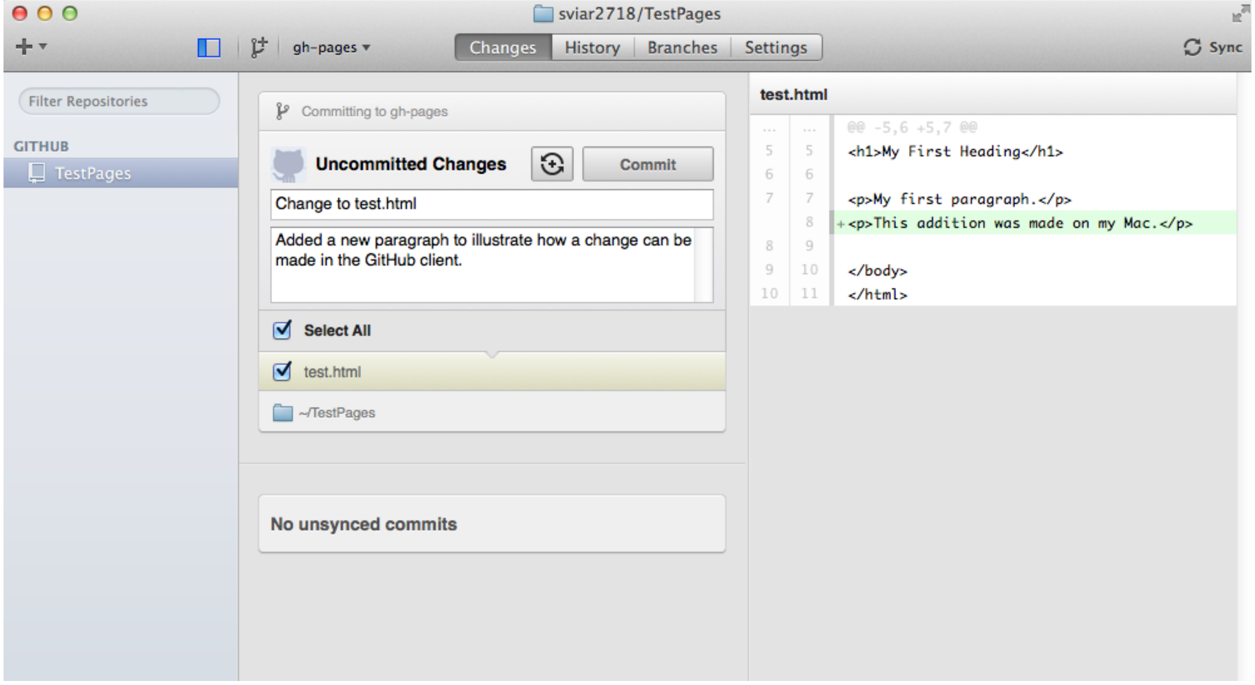
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Now make some changes and save the file



```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>
<p>This addition was made on my Mac.</p>

</body>
</html>
```

Close the file and go back to the GitHub client:



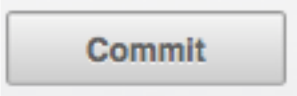
Change to test.html

Added a new paragraph to illustrate how a change can be made in the GitHub client.

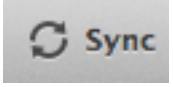
```
@@ -5,6 +5,7 @@
 <h1>My First Heading</h1>

 <p>My first paragraph.</p>
+<p>This addition was made on my Mac.</p>

 </body>
 </html>
```

Notice that it will highlight the change that was made. If you added something it will be highlighted in green. Anything you deleted will be highlighted in red. You should a comment to your commit and an (optional) description.

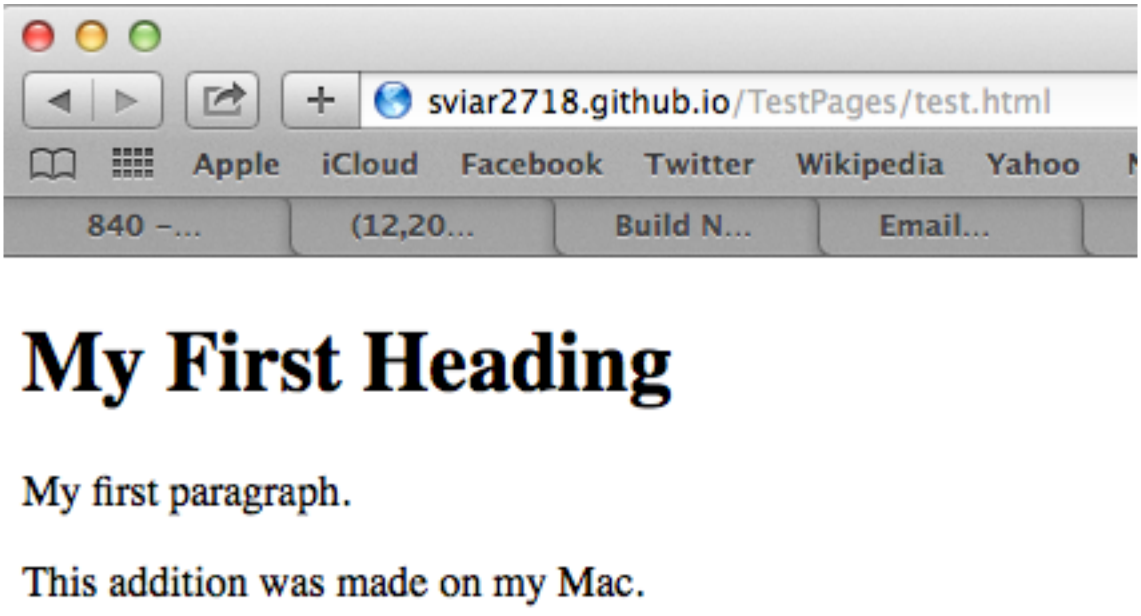
Click the Commit button



Now click the Sync button



Now go back to your webpage to see if you see your changes:



My First Heading

My first paragraph.

This addition was made on my Mac.

Next Steps:

If you would like to start learning HTML to be able to change your website, any Google search can find dozens of online tutorials. Here's a sample:

http://www.w3schools.com/html/
http://www.codecademy.com/tracks/web
http://www.austincc.edu/hr/profdev/eworkshops/docs/HTML_Basics.pdf

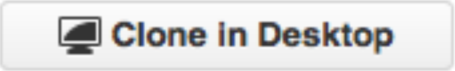
Take a look at Mr. Viar's GitHub site:

http://dviar2718.github.io/DanWeb/

If you would like make a site like this, the easiest way to get started would be to go to the GitHub repository for the site:

https://github.com/dviar2718/DanWeb

Click on



Once you have the entire repo on your home computer, you can copy all of the folder contents into your repo and make the changes that you want. Once you've done that you can commit the changes to your repo and then sync back. NOTE: If you do this, you will replace everything that you've done, so be careful!