Deploy using GitHub Desktop

Preface

My workflow is a bit different than what will be demonstrated in this lecture. I do not use GitHub desktop professionally. However, that doesn't mean it is not an acceptable tool to use in professional or personal environment. Tools are meant to get a job done and at the end of the day that's all that matters is the job gets done.

Because of this you may find differing material and resources online. All that is completely acceptable to follow if it works for you or others.

Good YouTube Series:

1. Getting Started With GitHub Part 1
2. Getting Started With GitHub Part 2
3. Getting Started With GitHub Part 3

What is GitHub Desktop?

GitHub Desktop is a Graphical User Interface for interacting with GitHub

Available on MacOS and Windows

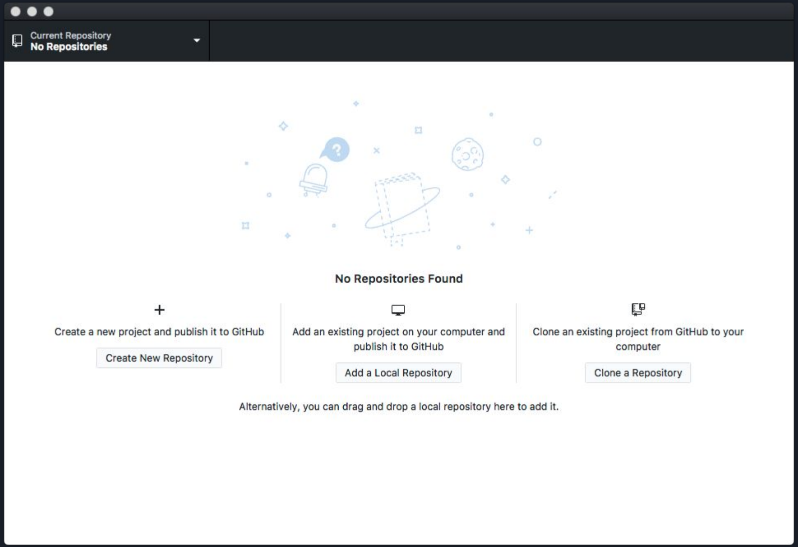
Designed by the folks at GitHub

Free

Installing GitHub Desktop

1. Download GitHub Desktop
2. Install GitHub Desktop (locate GitHubDesktop.zip on your file system)
3. Once installed you will be prompted to enter your GitHub.com credentials
4. Step through the rest of the steps, you may receive an e-mail a 3rd party app has authorized your account

Setting up your first Repo



No Repositories Found

Create a new project and publish it to GitHub

Add an existing project on your computer and publish it to GitHub

Clone an existing project from GitHub to your computer

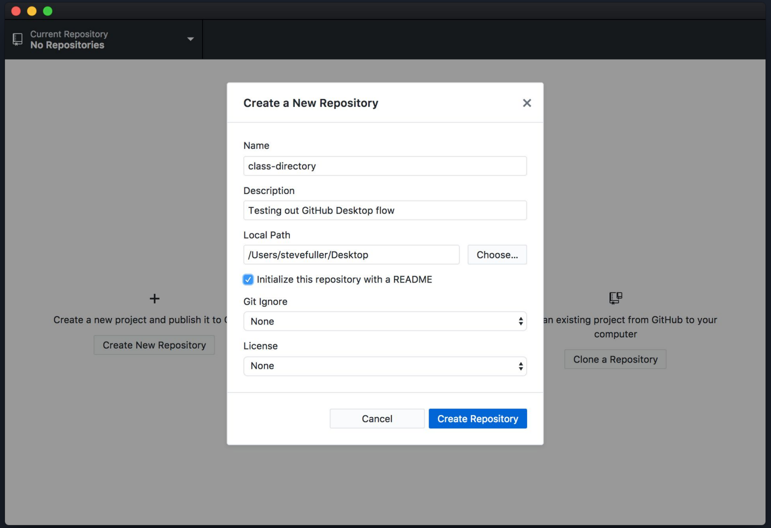
Alternatively, you can drag and drop a local repository here to add it.

- Create a new project and publish?
  - You don’t have Git files in your project
- Add an existing project and publish?
  - You have Git files in your project
- Clone an existing project from GitHub?
  - There’s code already on GitHub you want to download and work on

Create a new project

- Name your project
- Write Description
- Where do you want your Repo to reside on your filesystem?
- Initialize with README?
  - A README is information about your project like installation instruction(s), how to use, FAQs, etc.
- Git Ignore
  - Files you don’t want in GitHub
- License
  - Copyright protections

Creating this will create a new folder with your essential Git files



Current Repository
No Repositories

Create a New Repository

Name: class-directory

Description: Testing out GitHub Desktop flow

Local Path: /Users/stevefuller/Desktop

☑ Initialize this repository with a README

Git Ignore: None

License: None

Create a new project and publish it to (

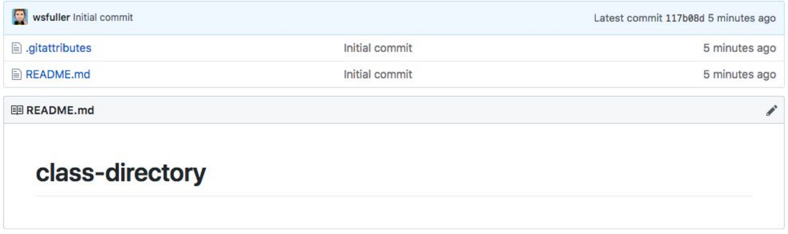
an existing project from GitHub to your computer

Publish Repository

In the top right corner press "Publish Repository"

Now go to:
github.com/:username/:repo-name

And you should see something like:

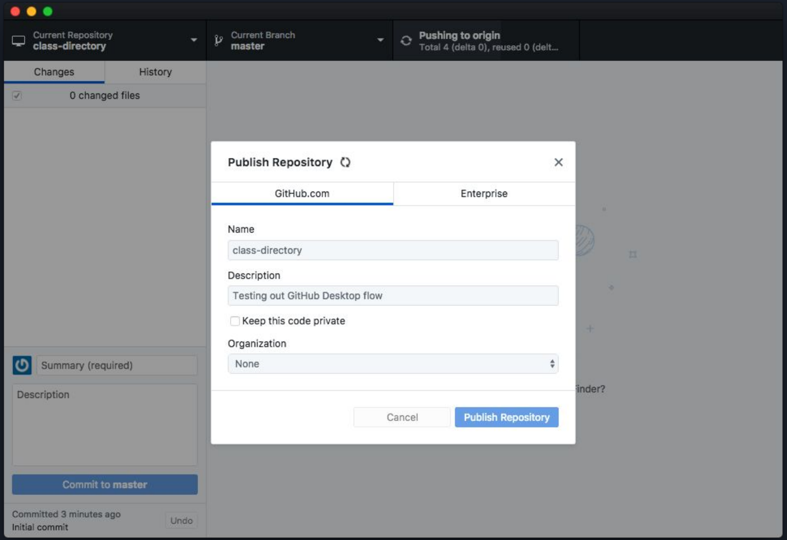


wsfuller Initial commit — Latest commit 117b08d 5 minutes ago

.gitattributes	Initial commit	5 minutes ago
README.md	Initial commit	5 minutes ago

README.md

class-directory



Current Repository: class-directory
Current Branch: master
Pushing to origin
Total 4 (delta 0), reused 0 (delt...

Changes | History

☑ 0 changed files

Publish Repository

GitHub.com | Enterprise

Name: class-directory
Description: Testing out GitHub Desktop flow
□ Keep this code private
Organization: None

Cancel | Publish Repository

Summary (required)
Description
Commit to master

Committed 3 minutes ago
Initial commit

Undo

Congratulations you’ve just published to GitHub!!!

But like, where’s my code and stuff?!?!

One more thing... the .gitignore file

.gitignore is a file we use to omit certain files from being submitted to GitHub

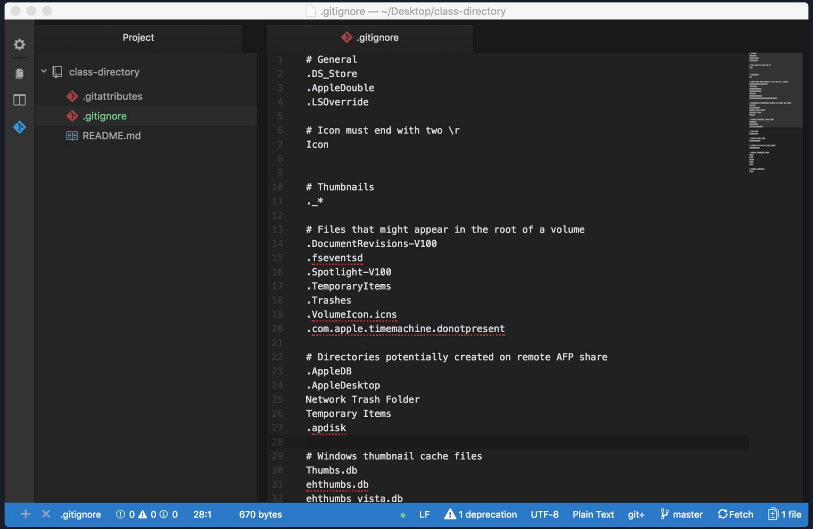
macOS gitignore

Windows gitignore

Here are a couple basic files that should cover our needs. For example though you don’t want to be committing things like .psd, .sketch, etc files into GitHub. The .gitignore should always be updated before for code commits to avoid any unnecessary files from accidentally being committed.

Adding your .gitignore

1. Open up your GitHub folder that was created in Atom
2. Make a new file called “.gitignore”
3. Paste in the sample code from the previous slide links
4. Commit this new file to your GitHub repo

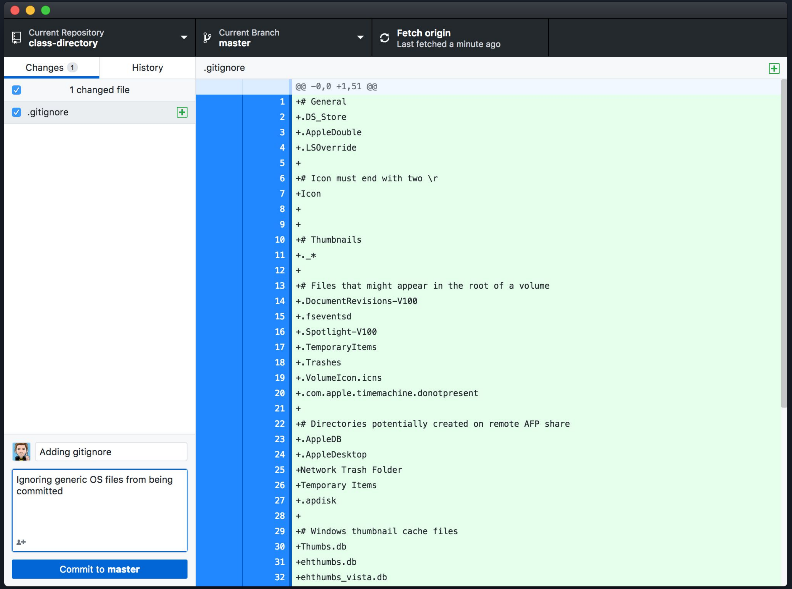


```
# General
.DS_Store
.AppleDouble
.LSOverride

# Icon must end with two \r
Icon


# Thumbnails
._*

# Files that might appear in the root of a volume
.DocumentRevisions-V100
.fseventsd
.Spotlight-V100
.TemporaryItems
.Trashes
.VolumeIcon.icns
.com.apple.timemachine.donotpresent

# Directories potentially created on remote AFP share
.AppleDB
.AppleDesktop
Network Trash Folder
Temporary Items
.apdisk

# Windows thumbnail cache files
Thumbs.db
ehthumbs.db
ehthumbs vista.db
```

+ × .gitignore ⓘ 0 ⚠ 0 ⓘ 0 28:1 670 bytes
● LF ⚠ 1 deprecation UTF-8 Plain Text git+ master Fetch 1 file

Commit your .gitignore

1. Open GitHub desktop
2. Fill out Summary
3. Fill out Desktoption (optional)
4. Commit to **master**
5. Click Push origin
6. Visit your repo



Current Repository: class-directory

Current Branch: master

1 changed file

☑ .gitignore

Summary: Adding gitignore

Description: Ignoring generic OS files from being committed

.gitignore

```
@@ -0,0 +1,51 @@
+# General
+.DS_Store
+.AppleDouble
+.LSOverride
+
+# Icon must end with two \r
+Icon
+
+
+# Thumbnails
+._*
+
+# Files that might appear in the root of a volume
+.DocumentRevisions-V100
+.fseventsd
+.Spotlight-V100
+.TemporaryItems
+.Trashes
+.VolumeIcon.icns
+.com.apple.timemachine.donotpresent
+
+# Directories potentially created on remote AFP share
+.AppleDB
+.AppleDesktop
+Network Trash Folder
+Temporary Items
+.apdisk
+
+# Windows thumbnail cache files
+Thumbs.db
+ehthumbs.db
+ehthumbs_vista.db
```

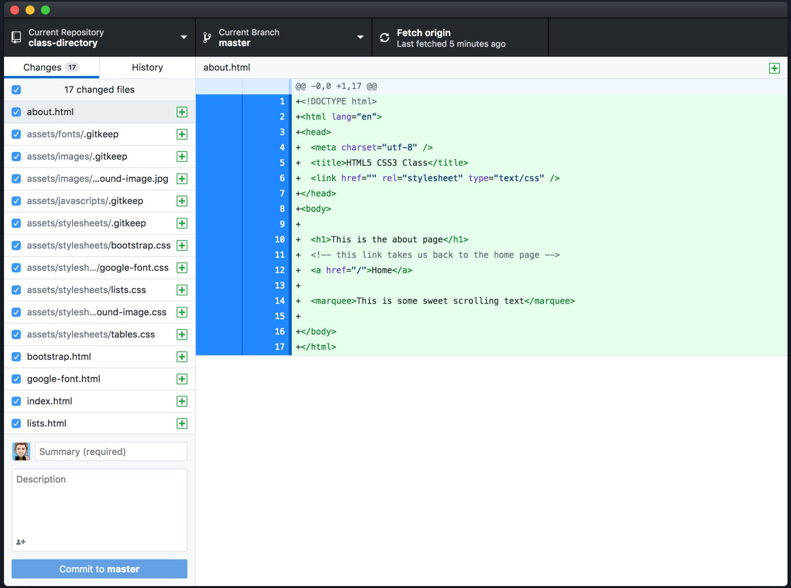
You’ve just updated your repo!!!

But like, where's my code and stuff?!?!

Let’s get our actual code up

All we’ve done is prepare a folder that’s linked to GitHub, there’s a couple more steps to get this all working

1. Copy your working code into the one you just created for GitHub
2. Now you should see your files showing up
3. Summary
4. Description
5. Commit to **master**
6. Push to origin
7. Visit your GitHub Repo
8. Pat self on back



Current Repository **class-directory**

Current Branch **master**

☑ 17 changed files

- ☑ about.html
- ☑ assets/fonts/.gitkeep
- ☑ assets/images/.gitkeep
- ☑ assets/images/...ound-image.jpg
- ☑ assets/javascripts/.gitkeep
- ☑ assets/stylesheets/.gitkeep
- ☑ assets/stylesheets/bootstrap.css
- ☑ assets/stylesh.../google-font.css
- ☑ assets/stylesheets/lists.css
- ☑ assets/stylesh...ound-image.css
- ☑ assets/stylesheets/tables.css
- ☑ bootstrap.html
- ☑ google-font.html
- ☑ index.html
- ☑ lists.html

about.html

```
@@ -0,0 +1,17 @@
+<!DOCTYPE html>
+<html lang=“en”>
+<head>
+  <meta charset="utf-8" />
+  <title>HTML5 CSS3 Class</title>
+  <link href="" rel="stylesheet" type="text/css" />
+</head>
+<body>
+
+  <h1>This is the about page</h1>
+  <!-- this link takes us back to the home page -->
+  <a href="/">Home</a>
+
+  <marquee>This is some sweet scrolling text</marquee>
+
+</body>
+</html>
```

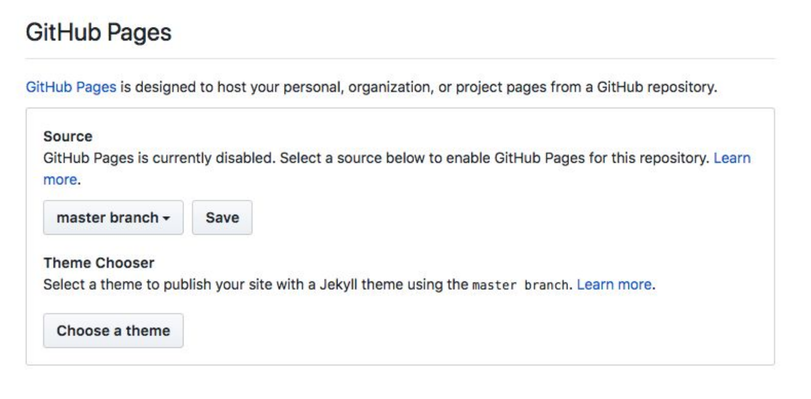
Oh yeah

But we aren’t quite done yet…

Publish to GitHub Pages

GitHub pages will allow GitHub to take our code and publish a working website for everyone to view

1. Go to your GitHub Repo page
2. Click the Settings tab
3. Scroll down till you see “GitHub Pages”
4. Source should be your master branch
5. Click save
6. You should see a link: “Your site is ready to be published at :URL
7. Go grab a coffee and should be live in ~30 minutes



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

Theme Chooser

Select a theme to publish your site with a Jekyll theme using the `master branch`. Learn more.

Congrats on using GitHub and Deploying your site!!!