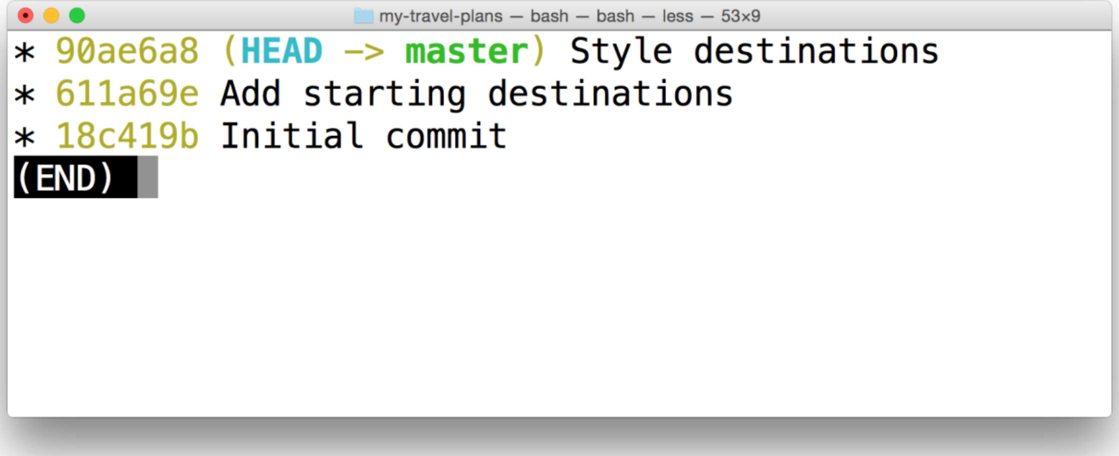
Push Changes To A Remote

Reviewing Commits

Let's take a look at the commits that I have in my Repository.



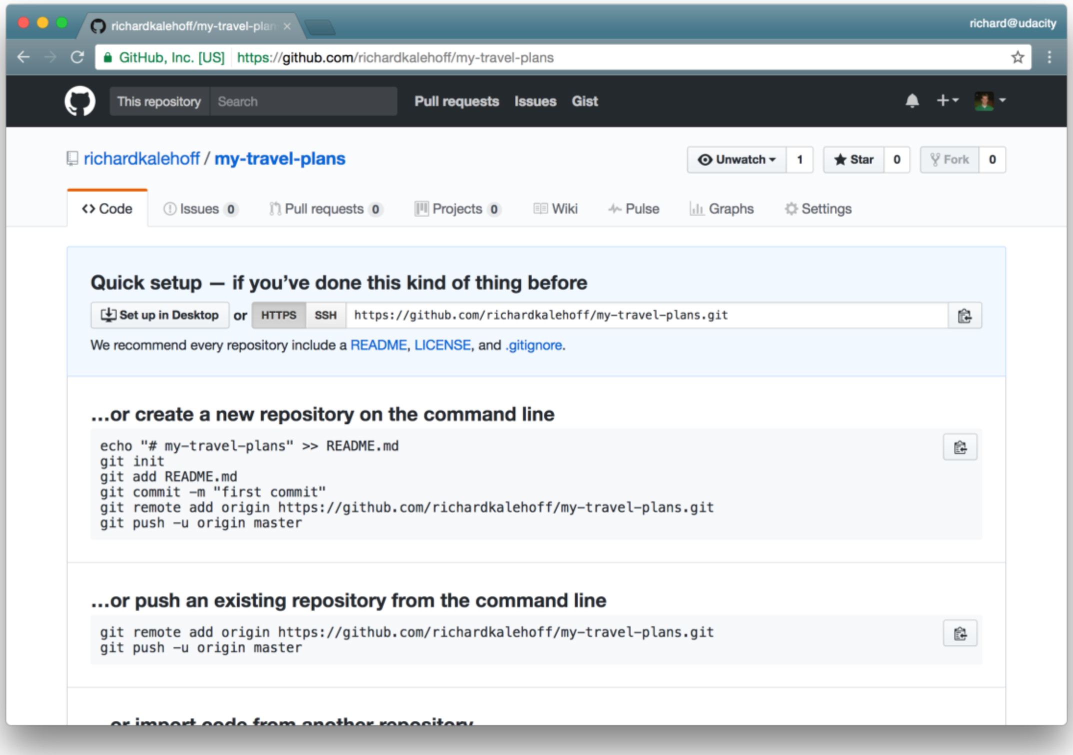
```
* 90ae6a8 (HEAD -> master) Style destinations
* 611a69e Add starting destinations
* 18c419b Initial commit
(END)
```

The terminal application displaying the log of the project's commits.

I used the following log command to display these commits

```
$ git log --oneline --graph --decorate --all
```

These commits are only in the local repository, though. They haven't been sent up to the remote repository yet. When commits are sent to the remote a remote branch indicator will appear in the log. Since there aren't any remote branch indicators we can tell that there are no commits on the remote repository. But just to be 100% certain let's look at the remote repository on GitHub to see if there any commits.


richardkalehoff/my-travel-plan ×
richard@udacity
GitHub, Inc. [US] https://github.com/richardkalehoff/my-travel-plans
This repository
Search
Pull requests
Issues
Gist
richardkalehoff / my-travel-plans
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
Set up in Desktop
or
HTTPS
SSH
https://github.com/richardkalehoff/my-travel-plans.git

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

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

…or create a new repository on the command line

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

…or push an existing repository from the command line

```
git remote add origin https://github.com/richardkalehoff/my-travel-plans.git
git push -u origin master
```

…or import code from another repository

The remote repository doesn't contain any commits, so GitHub displays the repository's setup screen.

Since we haven't sent any commits update to GitHub yet it's still showing us the setup screen to tell us how we can connect our local repository to the remote repository and send some commits. Since this is still the setup screen we can know that there are no commits in the remote repository.

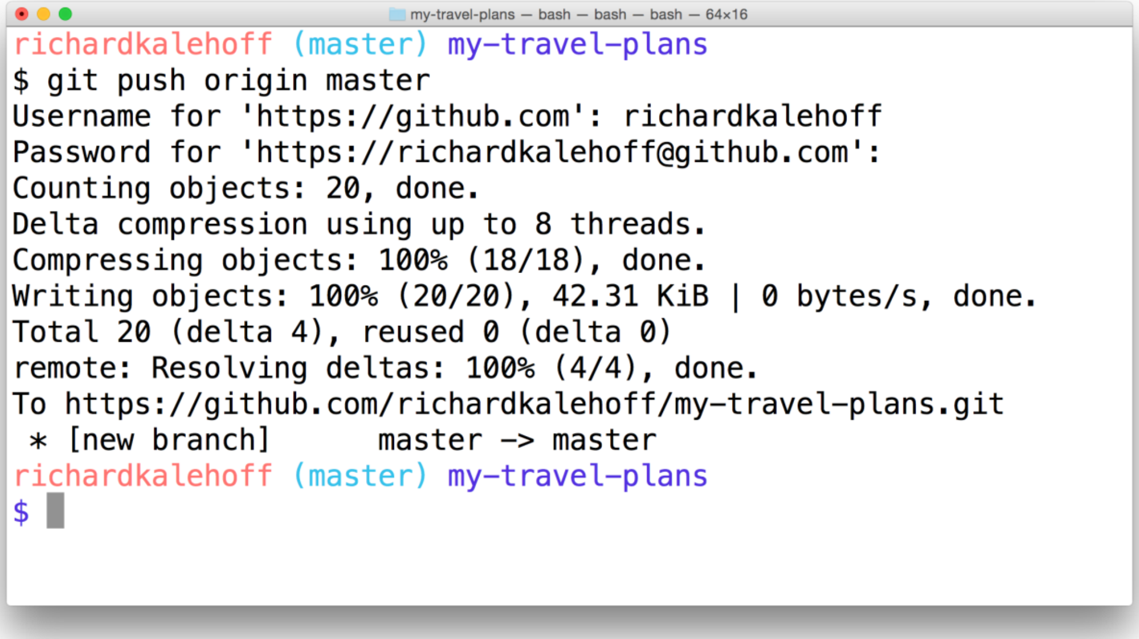
Sending Commits

To send local commits to a remote repository you need to use the git push command. You provide the remote short name and then you supply the name of the branch that contains the commits you want to push:

```
$ git push <remote-shortname> <branch>
```

My remote's shortname is origin and the commits that I want to push are on the master branch. So I'll use the following command to send my commits to the remote repository on GitHub:

```
$ git push origin master
```



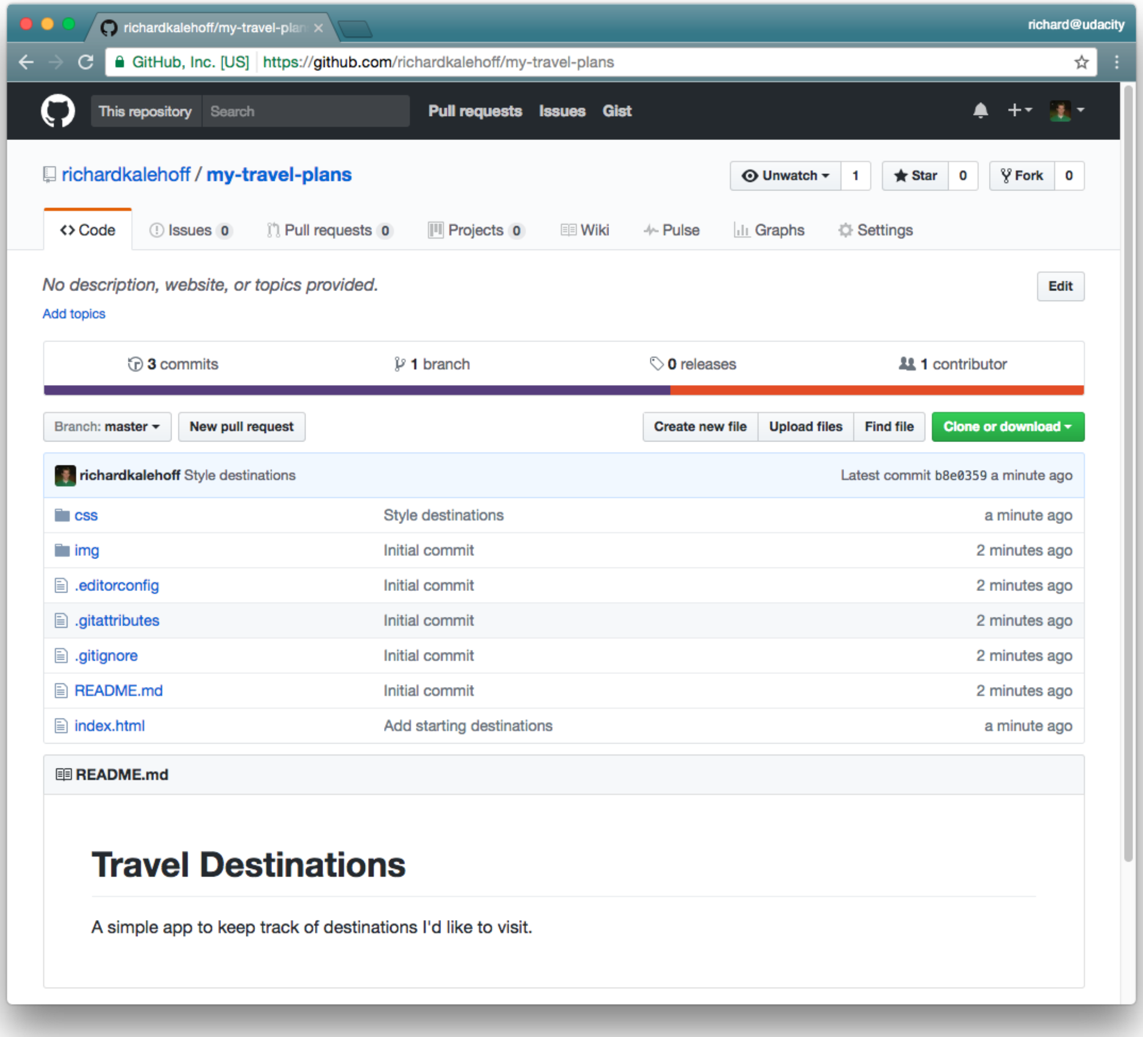
```
richardkalehoff (master) my-travel-plans
$ git push origin master
Username for 'https://github.com': richardkalehoff
Password for 'https://richardkalehoff@github.com':
Counting objects: 20, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (20/20), 42.31 KiB | 0 bytes/s, done.
Total 20 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/richardkalehoff/my-travel-plans.git
 * [new branch]      master -> master
richardkalehoff (master) my-travel-plans
$ 
```

The terminal application showing the git push command. It asks for my username, password (which isn't displayed), and then displays information about what it's doing to send the commits.

There a couple of things to notice:

- Depending on how you have configured GitHub and the remote URL that's being used, you might have to enter your username and password.
    - this will happen if you use the HTTP version of the remote (rather than the ssh version)
    - If you have configured GitHub to use the SSH protocol and have supplied it with your SSH key then you don't need to worry about doing this step. Check the Connecting to GitHub with SSH documentation page if you're interested in using SSH with GitHub.
- If you have to enter your username and password your username will show up after typing but your password will not. So just keep typing your password and press enter when you're done.
    - If you encounter any errors with your password don't worry it'll just ask you to type it in again
- Git does some compressing of things to make it smaller and then sends those off to the remote
- A new branch is created - at the very bottom it says [new branch] and then master -> master

Now let's look at GitHub:



No description, website, or topics provided.

richardkalehoff Style destinations — Latest commit b8e0359 a minute ago

css	Style destinations	a minute ago
img	Initial commit	2 minutes ago
.editorconfig	Initial commit	2 minutes ago
.gitattributes	Initial commit	2 minutes ago
.gitignore	Initial commit	2 minutes ago
README.md	Initial commit	2 minutes ago
index.html	Add starting destinations	a minute ago

README.md

Travel Destinations

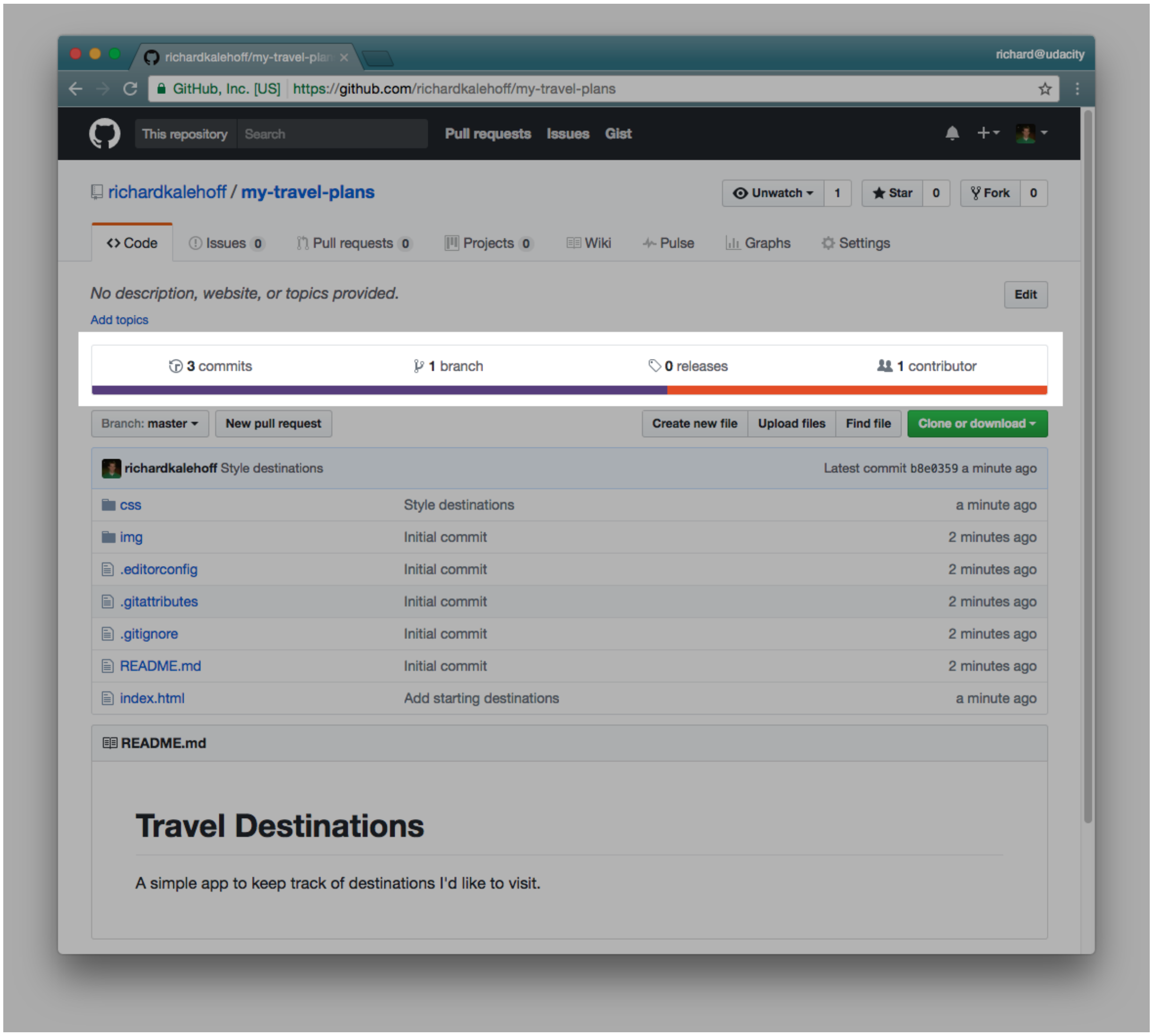
A simple app to keep track of destinations I'd like to visit.

My entire project is up on GitHub!

Our project is up on GitHub - how awesome and easy was that! One cool feature that GitHub does is that it automatically shows the content of the README file which can be extremely helpful.

GitHub also displays a lot of details about our Repository. Right now it's showing that there are:

- three commits
- one branch
- one contributor



No description, website, or topics provided.

richardkalehoff Style destinations — Latest commit b8e0359 a minute ago

css	Style destinations	a minute ago
img	Initial commit	2 minutes ago
.editorconfig	Initial commit	2 minutes ago
.gitattributes	Initial commit	2 minutes ago
.gitignore	Initial commit	2 minutes ago
README.md	Initial commit	2 minutes ago
index.html	Add starting destinations	a minute ago

Travel Destinations

A simple app to keep track of destinations I'd like to visit.

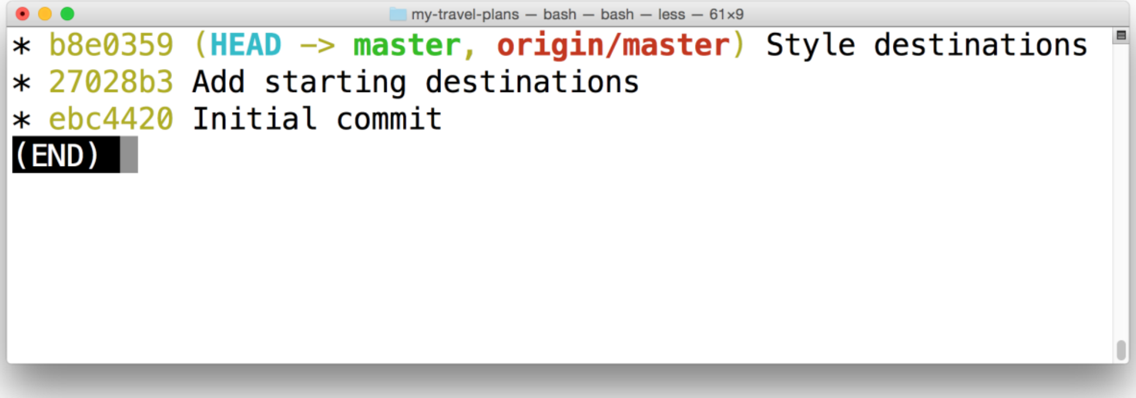
The project's main page on GitHub displays information about the repository.

Now before we move on, let's just check the local repository to see how it changed after pushing.

Run the following command:

```
$ git log --oneline --graph --decorate --all
```

Important: make sure to include the --decorate and --all flags



```
* b8e0359 (HEAD -> master, origin/master) Style destinations
* 27028b3 Add starting destinations
* ebc4420 Initial commit
(END)
```

The terminal application showing the results of running git log --oneline --graph --decorate --all. The new origin/master tracking branch now exists.

We now have a new marker in the output! This marker is origin/master and is called a **tracking branch**. A tracking branch's name includes the shortname of the remote repository as well as the name of the branch. So the tracking branch origin/master is telling us that the remote origin has a master branch that points to commit 9b7d28f (and includes all of the commits before 9b7d28f). This is really helpful because this means we can track the information of the remote Repository right here in our local one!

One *very important thing to know* is that this origin/master tracking branch is not a live representation of where the branch exists on the remote repository. If a change is made to the remote repository not by us but by someone else, the origin/master tracking branch in our local repository will not move. We have to tell it to go check for any updates and *then* it will move. We'll look at how to do this in the next section.

Recap

The git push command is used to send commits from a local repository to a remote repository.

```
$ git push origin master
```

The git push command takes:

- the shortname of the remote repository you want to send commits to
- the name of the branch that has the commits you want to send