4.5.9 Deploy the Application

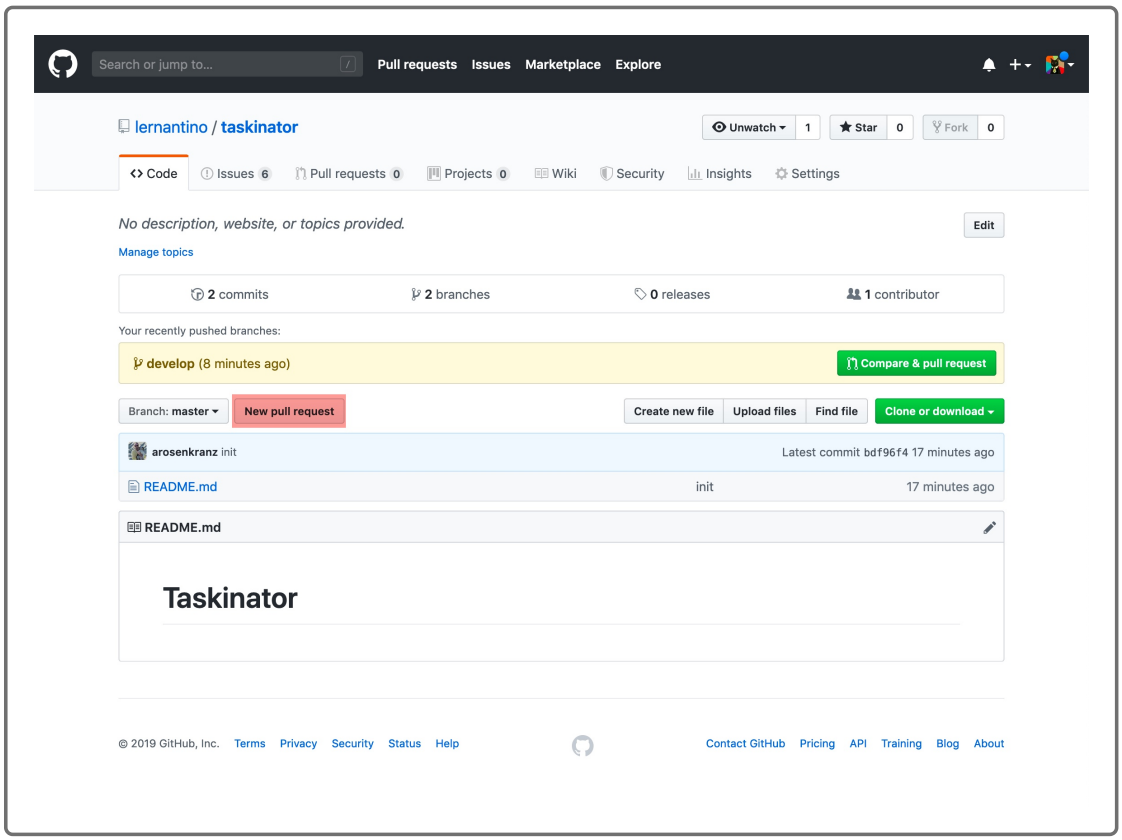
The rest of the work will all happen in the GitHub repositories. With all the finished code in the `develop` branch, we just need to get it to the `master` branch and deploy it.

We could use a similar strategy and locally merge the code into `master`, then push up to `master`, but if we were on a team that strategy wouldn't work. We want to use GitHub to merge any code into `master`, giving potential team members one more chance to review and test the code.

REWIND

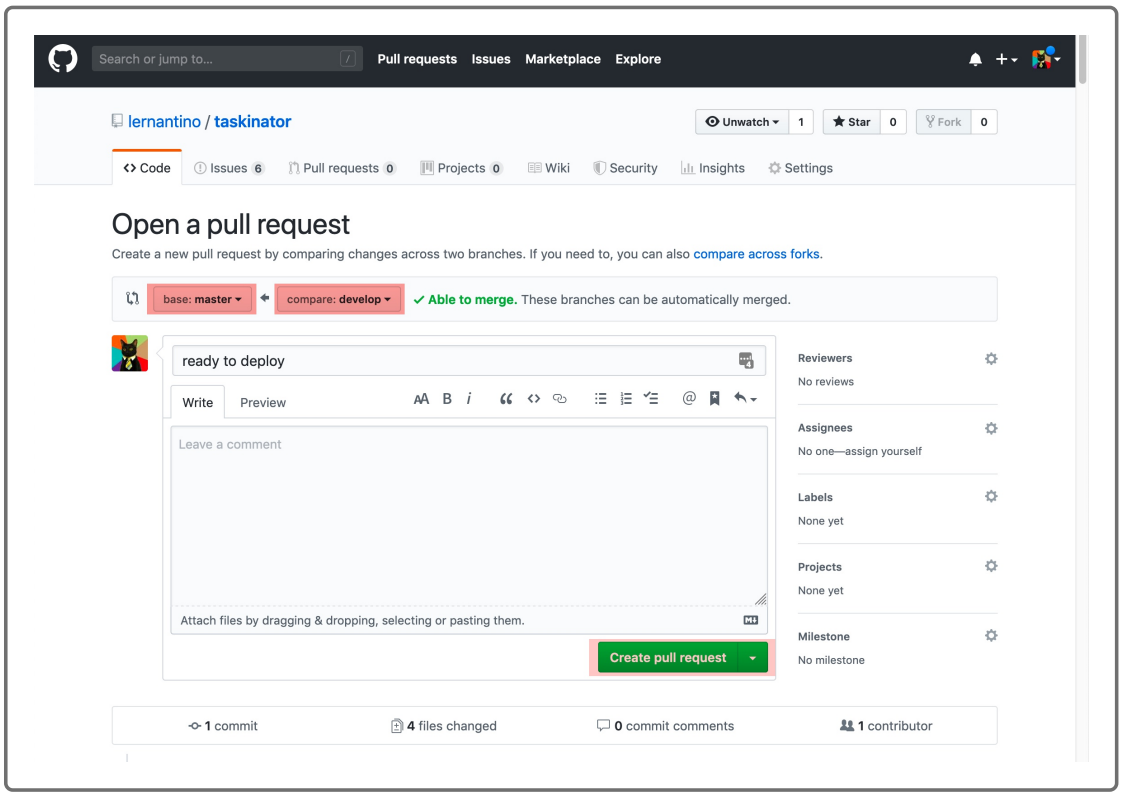
This is called **code review.**

First we need to open a GitHub pull request. We can do so by navigating to the main page of the repository (the Code tab) and clicking the New Pull Request button. See this image for reference:



Once we click that button, we have to chose the branches we want to merge code into and the branch the code is coming from. Under Compare Changes, a gray box should show two dropdowns: a `Base` branch and a `Compare` branch. For the `Base` branch, select `master`. For the `Compare` branch, select `develop`.

With the two branches picked now, let's open the pull request by clicking the green button that says Create Pull Request. This image shows how that page looks in GitHub:



Once we click that button, we'll be taken to a page for the pull request, under the Pull Requests tab in the repository. This page has all the options pertaining to a pull request, including who the pull request is assigned to and who we want to review the code before merging the pull request.

While you don't have any teammates to review the code, you've tested it so much throughout the build that you should feel confident it's working correctly. Plus because this is a personal project, you can always go back and fix things!

ON THE JOB

Because personal projects carry low stakes, we can always fix broken or bug-filled code after deploying. We should never do that for applications for paying clients or businesses; that's what code reviews are for.

Towards the bottom of this page, we should see a box with a green button that says Merge Pull Request. Click that button.

HIDE HINT

If the button isn't green and has a different message, that means work has been done in the `master` branch that isn't reflected in the `develop` branch.

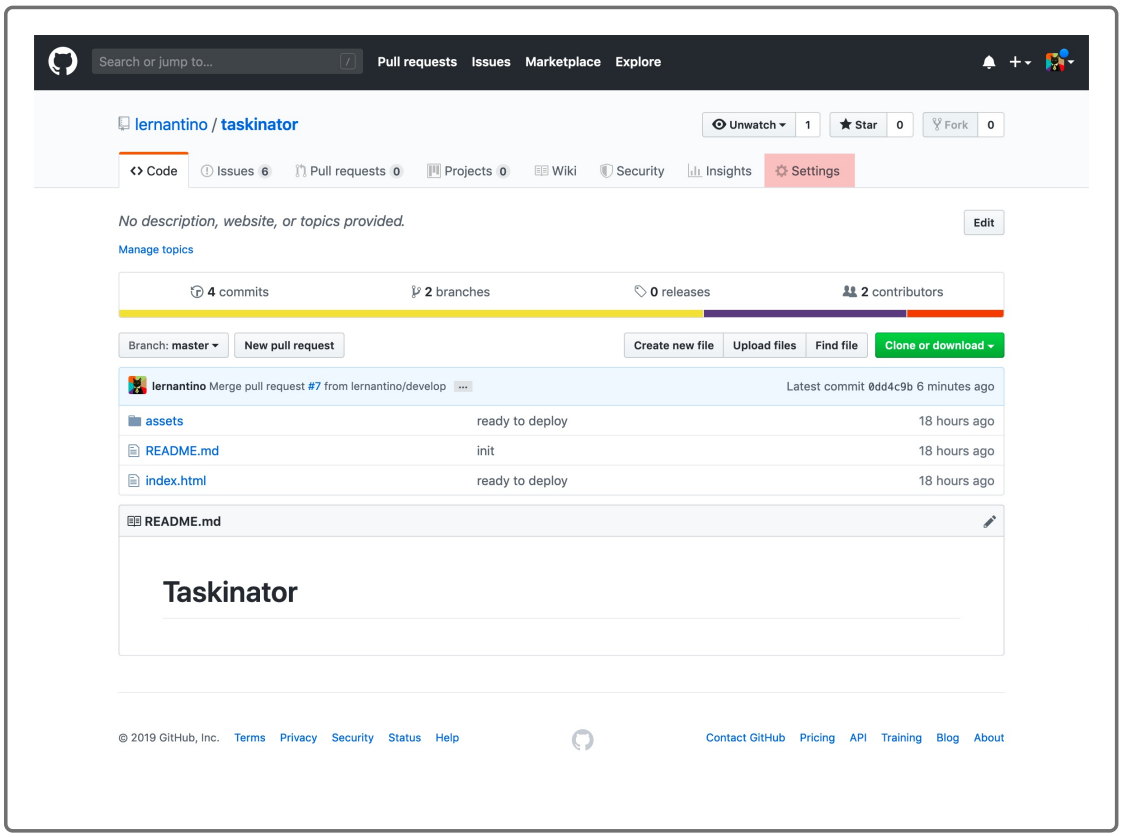
Now we have to decide whether or not to use `git pull origin master` in the local `develop` branch to integrate that code, then push it back up to `develop` on GitHub. Or we could ignore it and merge the pull request anyway, hoping for the best.

It's much safer to sync all code between branches before merging.

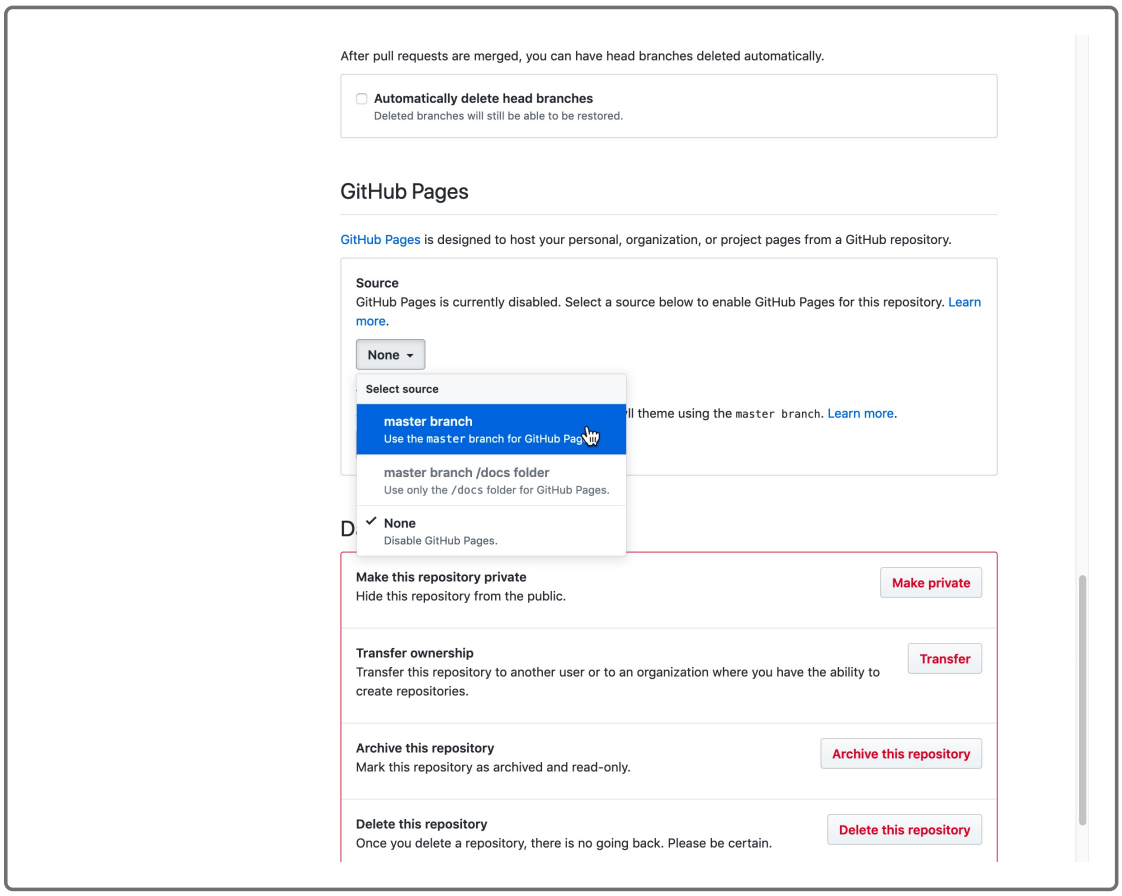
If the button has turned purple, the code is in the `master` branch! Let's go ahead and deploy it now.

Deploy to GitHub Pages

To deploy an application to GitHub pages, navigate to the Settings tab in the GitHub repository. This image outlines that tab:



Once there, we can scroll down the page until we see a section called GitHub Pages. In that section, a dropdown menu lets us pick which branch we want to deploy. Select master branch from those options.



After selection, the page will refresh itself. If we navigate back to the GitHub Pages section, we'll see a note letting us know the application is deployed at `https://<github-username>.github.io/taskinator`. Clicking that link will take us to our application on the internet!

Sometimes it takes GitHub Pages a minute or two to deploy, so don't panic if it's not immediately available.

Now we can use Taskinator anywhere! Remember, though, `localStorage` is unique to each device's browser, so tasks won't carry across devices. To achieve that, we'd need a remote database—which we'll learn about in the coming weeks.

Let's wrap up and reflect on all the amazing work we've completed.

If you wanted to access the `age` of the second animal in the following array, which one of these methods would NOT work?

```
var animalArr = [
  {
    name: "Fido the Field Mouse",
    age: 6
  },
  {
    name: "Lucy the Llama",
    age: 8
  },
  {
    name: "Dottie the Dog",
    age: 13
  }
];
```

- ◯ `animalArr.1.age`
- ◯ `animalArr[1].age`
- ◯ `animalArr[1]["age"]`

Check Answer

If you save your array of objects to the browser's local storage and it looks like `[Object object]` when you visit it in Chrome's DevTools, what's wrong?

- ◯ The array wasn't stringified with `JSON.stringify()` before saving it in Local Storage.
- ◯ The array wasn't parsed with `JSON.parse()` before saving it to Local Storage.

Check Answer

Finish ▸