Practice

Outline

- Chapter 1: Interface with Github
- Chapter 2: Other git tools

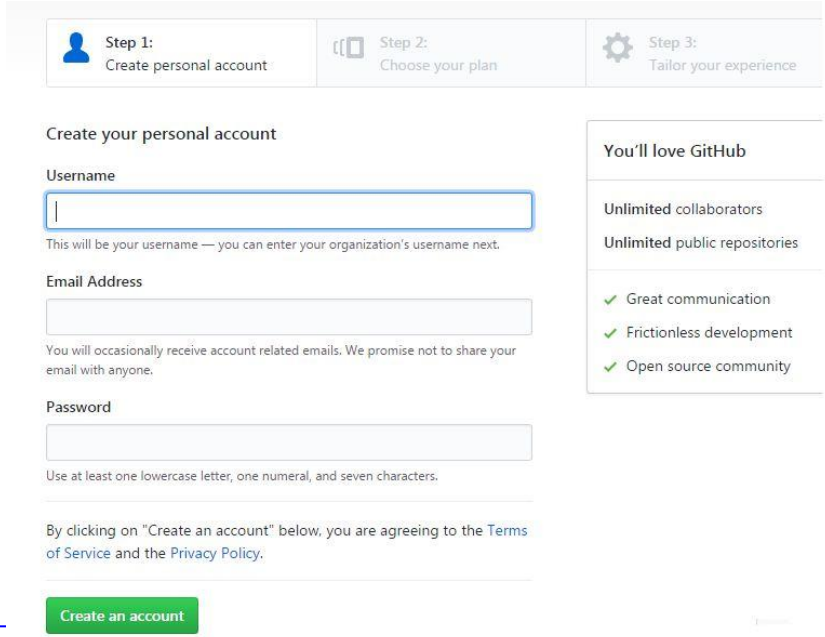
I. Interface with Github

Interface with Github

1. Create Github account
2. Create new repository
3. Clone
4. First commit
5. Push to Github
6. Create pull request
7. Review pull request

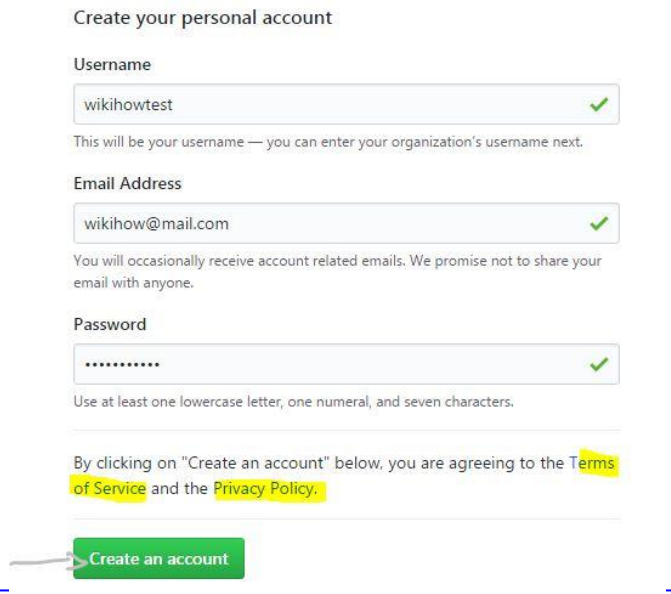
1. Create Github account

➢ Go to the [GitHub sign up page]



1. Create Github account

- Enter a username, valid email address, and password.
  Use at least one lowercase letter, one numeral, and seven characters.

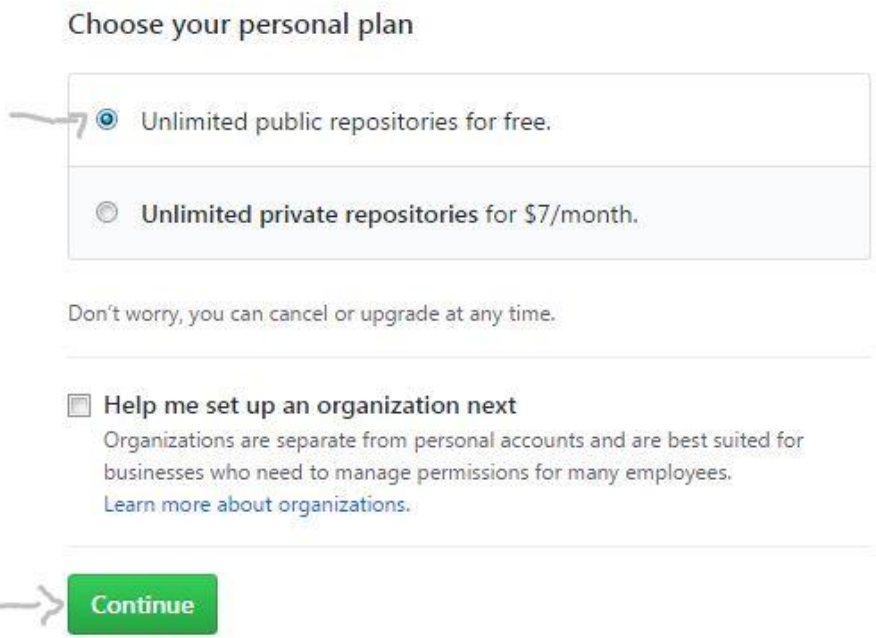


1. Create Github account

- Review carefully the GitHub Terms of Service and Privacy Policy before continuing.
  Upon clicking the “Create an account” button you will simultaneously be agreeing to these documents.

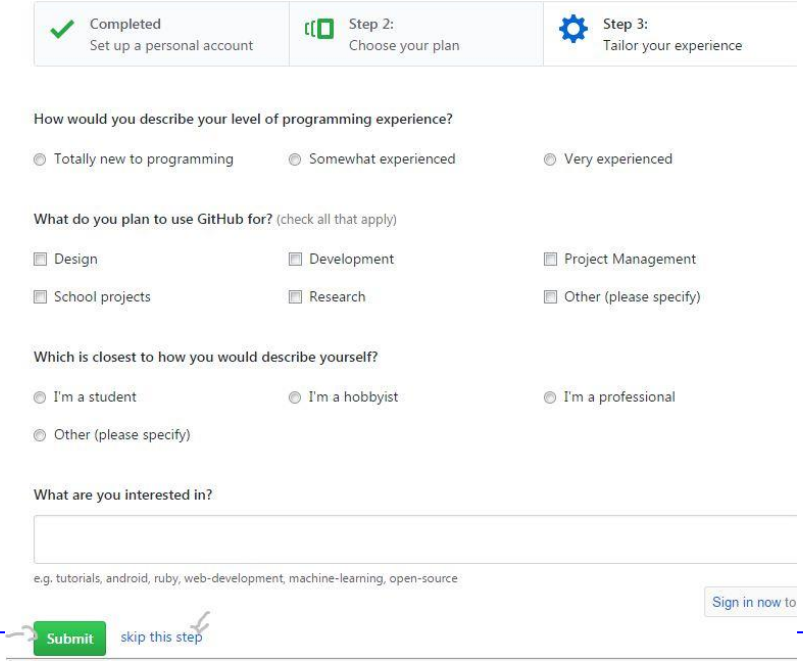
1. Create Github account

➢ Choose a plan.



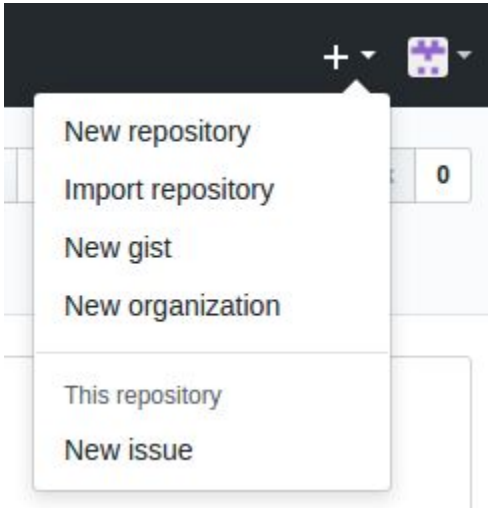
1. Create Github account

➢ Tailor experience.



2. Create new repository

- In the upper-right corner of any page, click + , and then click New repository.



2. Create new repository

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

framgit / training

Great repository names are short and memorable. Need inspiration? How about **refactored-octo-engine**.

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 let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

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

Create repository

2. Create new repository

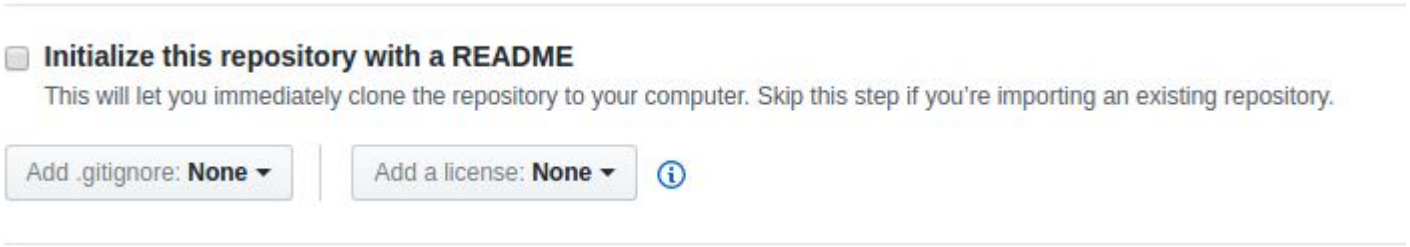
- In the page that create new repository, select the account you wish to create the repository on.
- Type a name for your repository, and an optional description.

2. Create new repository

- You can choose to make the repository either public or private. Public repositories are visible to the public, while private repositories are only accessible to you, and people you share them with. Your account must be on a paid plan to create a private repository.

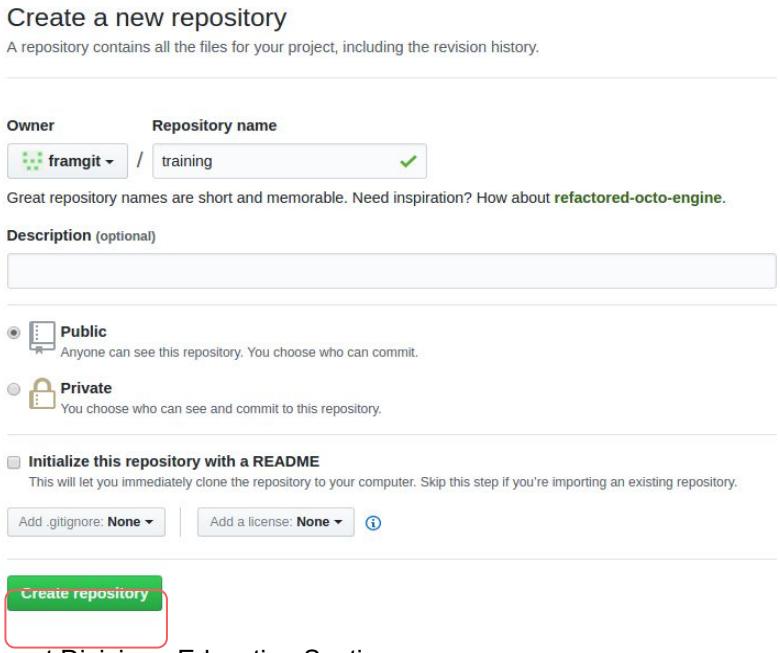
2. Create new repository

- There are a number of optional items you can pre-populate your repository with. If you're importing an existing repository to GitHub, don't choose any of these options, as you may introduce a merge conflict. You can choose to add these files using the command line later.



2. Create new repository

- When you're finished, click Create repository.



2. Create new repository

➢ **Create a new repository on the command line**

- Initialize the local directory as a Git repository.

```
$ git init
```

- Add the files in your new local repository. This stages them for the first commit.

```
$ git add .
```

Create new repository

- Commit the files that you've staged in your local repository.

```
$ git commit -m "First commit"
```

- In Terminal, add the URL for the remote repository where your local repository will be pushed.

```
$ git remote add origin remote repository URL
```

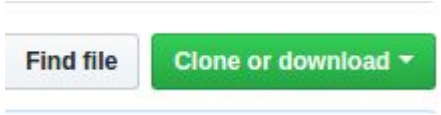
2. Create new repository

- Push the changes in your local repository to GitHub.

```
$ git push origin master
```

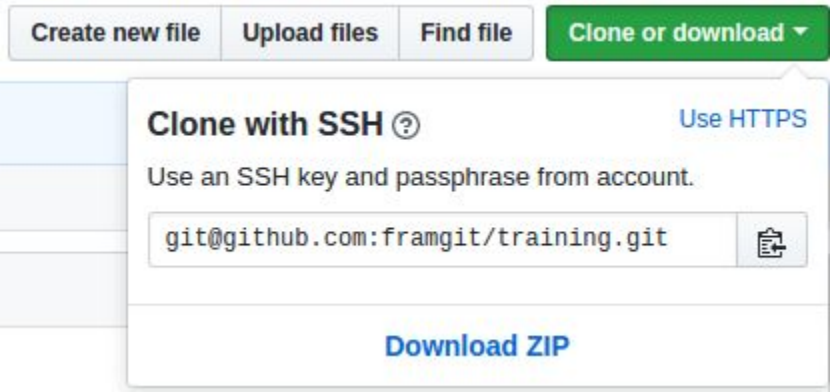
3. Clone

- On GitHub, navigate to the main page of the repository.
- Under the repository name, click Clone or download.



3. Clone

- In the Clone with HTTPs section, click to copy the clone URL for the repository.



3. Clone

- Open Terminal.
- Change the current working directory to the location where you want the cloned directory to be made.
- Type git clone, and then paste the URL you copied in Step 2.

```
$ git clone https://github.com/your-username/your-repository
```

3. Clone

- Press Enter. Your local clone will be created.

```
$ git clone https://github.com/your-username/your-repository
```

4. First commit

- Stage the file for commit to your local repository.

```
$ git add .
```

- Commit the file that you've staged in your local repository.

```
$ git commit -m "Add existing file"
```

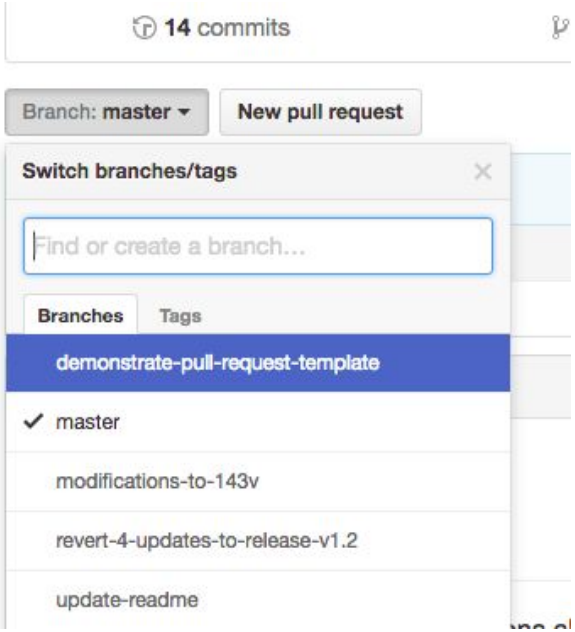
5. Push to Github

- Stage the file for commit to your local repository.

```
$ git push origin your-branch
```

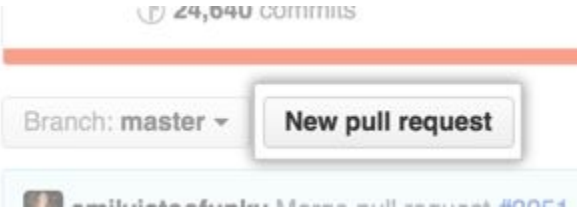
6. Create pull request

- On GitHub, navigate to the main page of the repository.
- In the "Branch" menu, choose the branch that contains your commits.



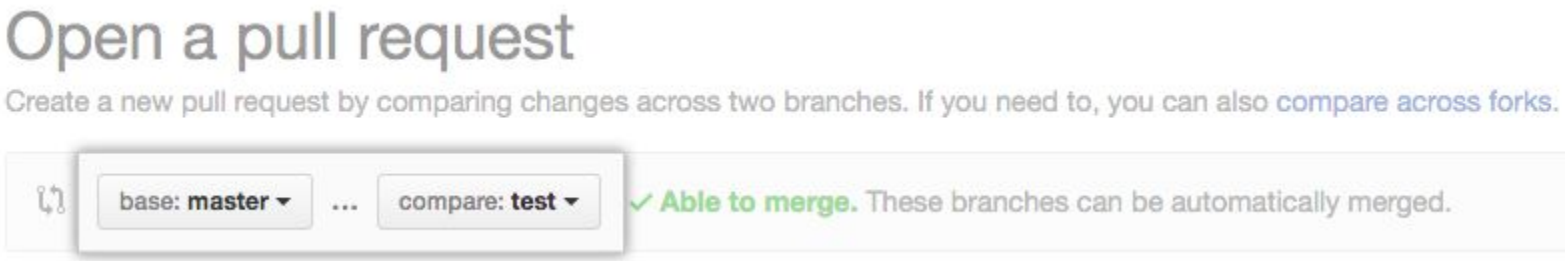
6. Create pull request

- To the right of the Branch menu, click New pull request.



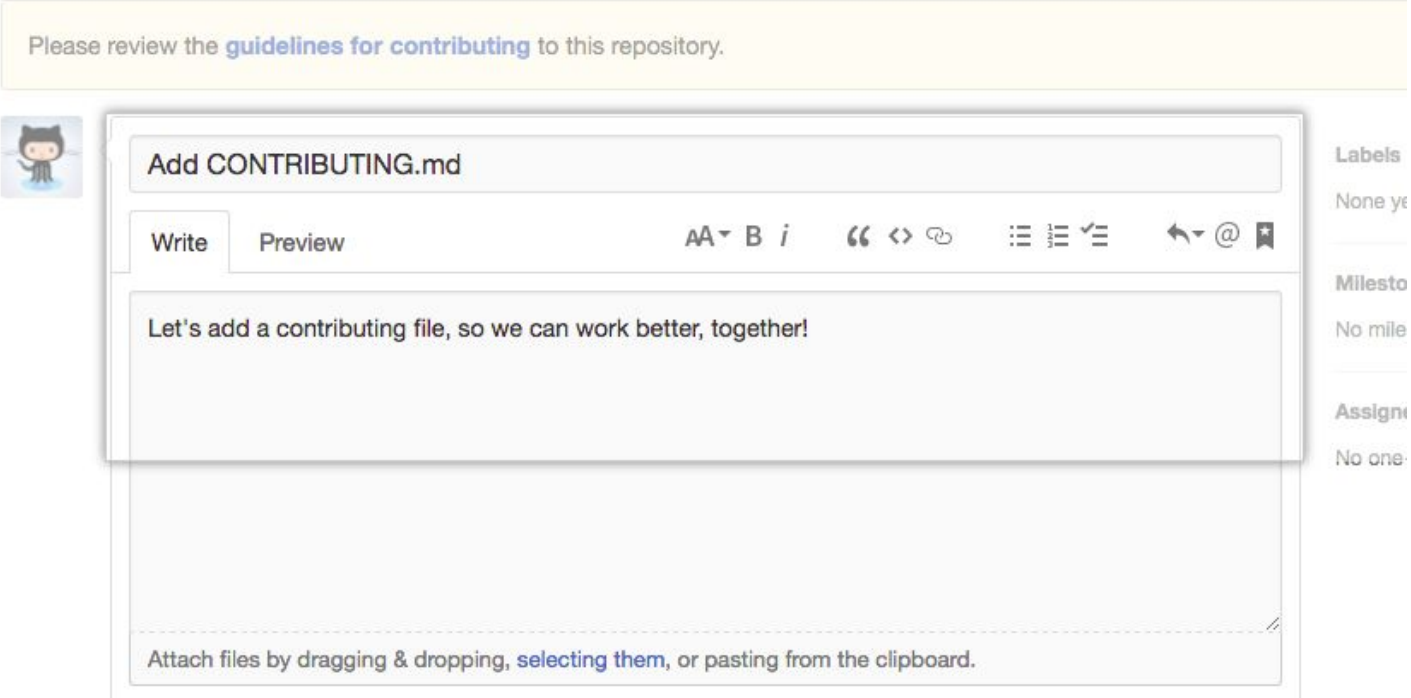
6. Create pull request

- Use the *base* branch dropdown menu to select the branch you'd like to merge your changes into, then use the *compare* branch drop-down menu to choose the topic branch you made your changes in.



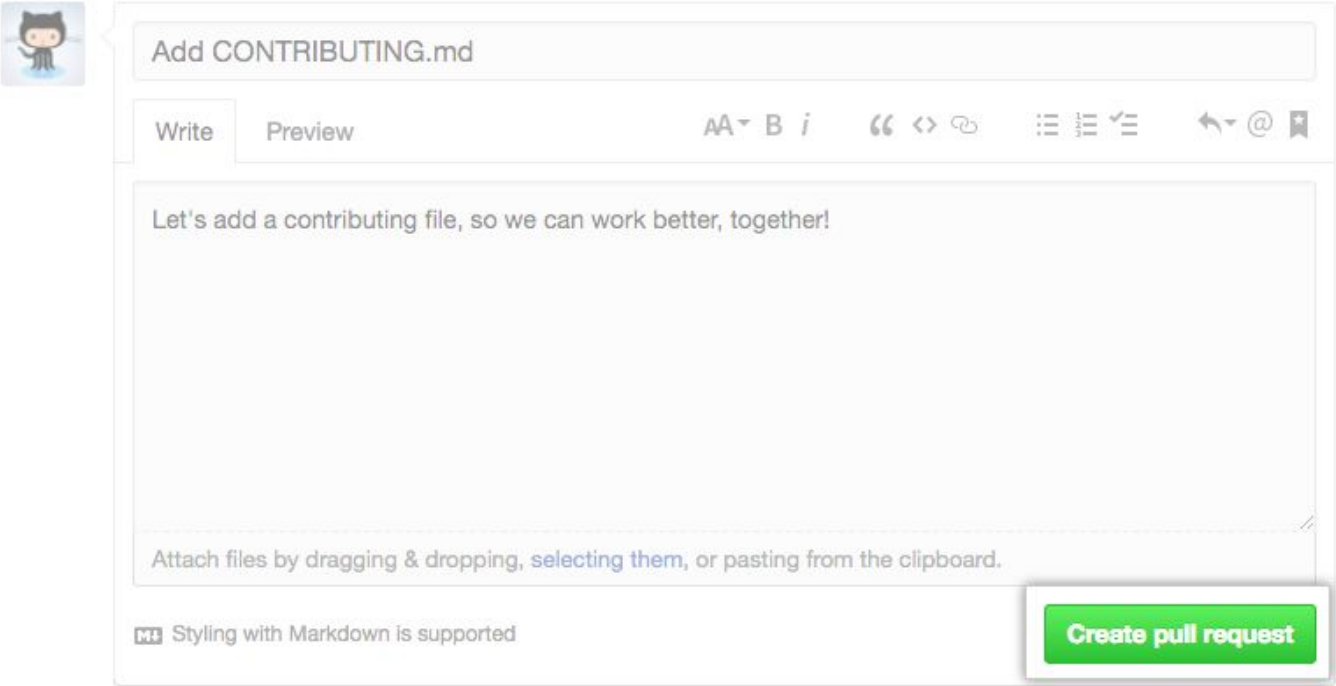
6. Create pull request

- Type a title and description for your pull request.

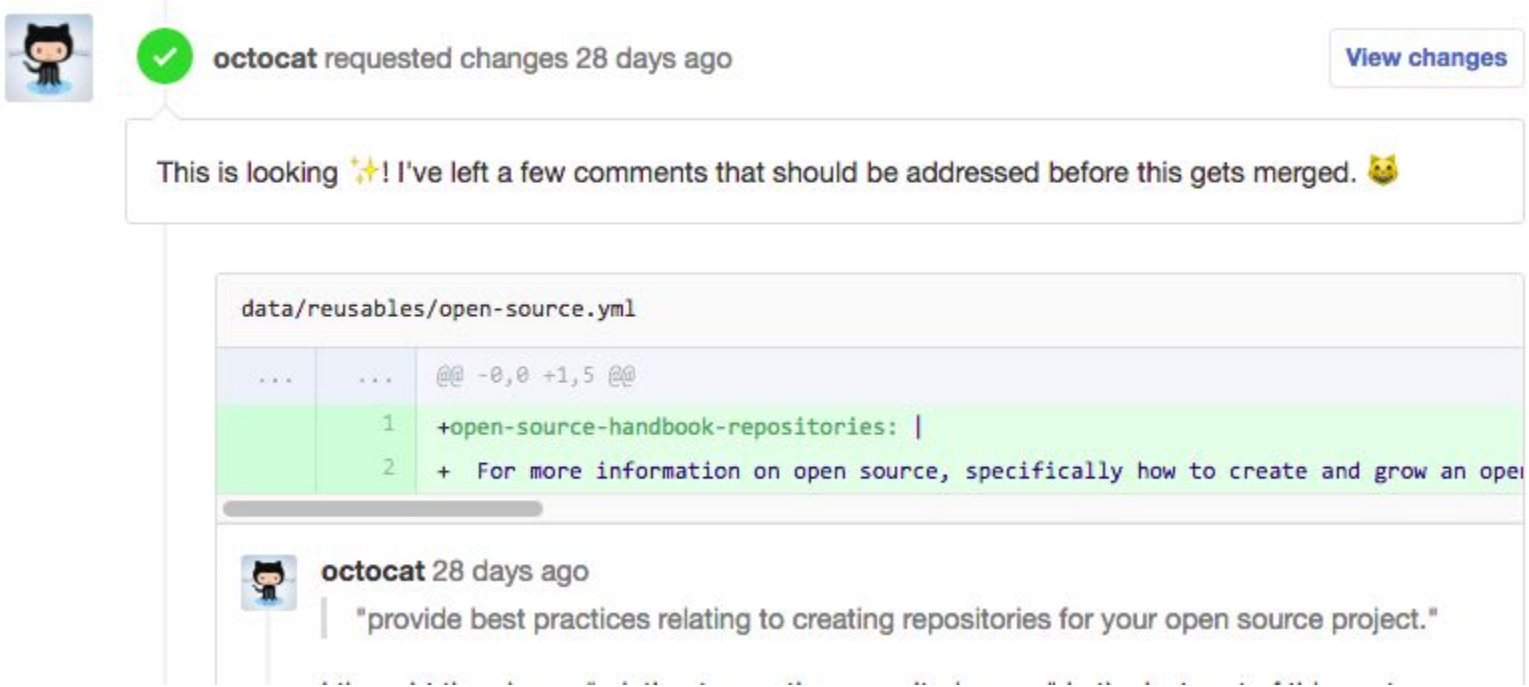


6. Create pull request

- Click Create pull request.



7. Review pull request

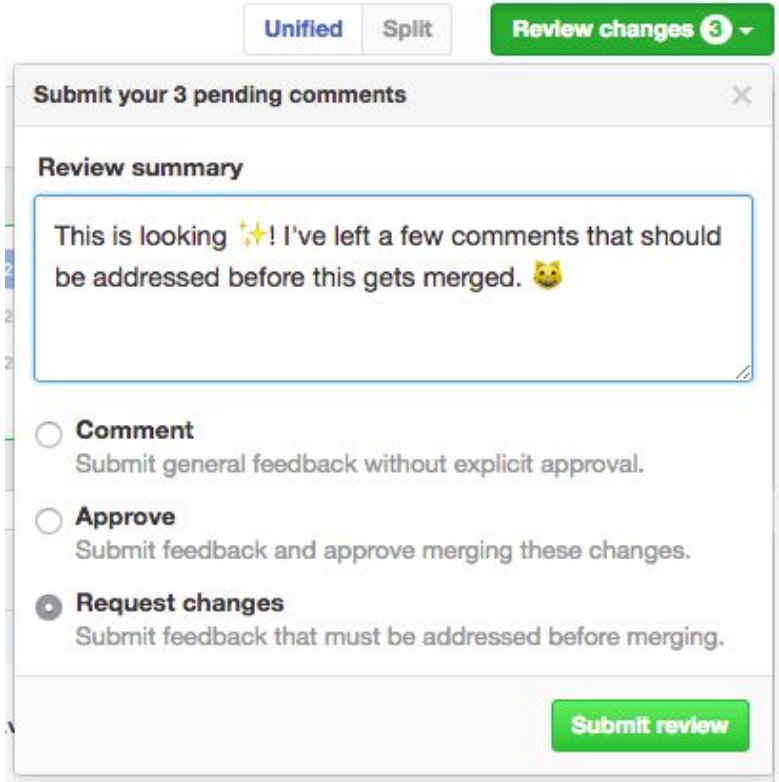


7. Review pull request

A review has three possible statuses:

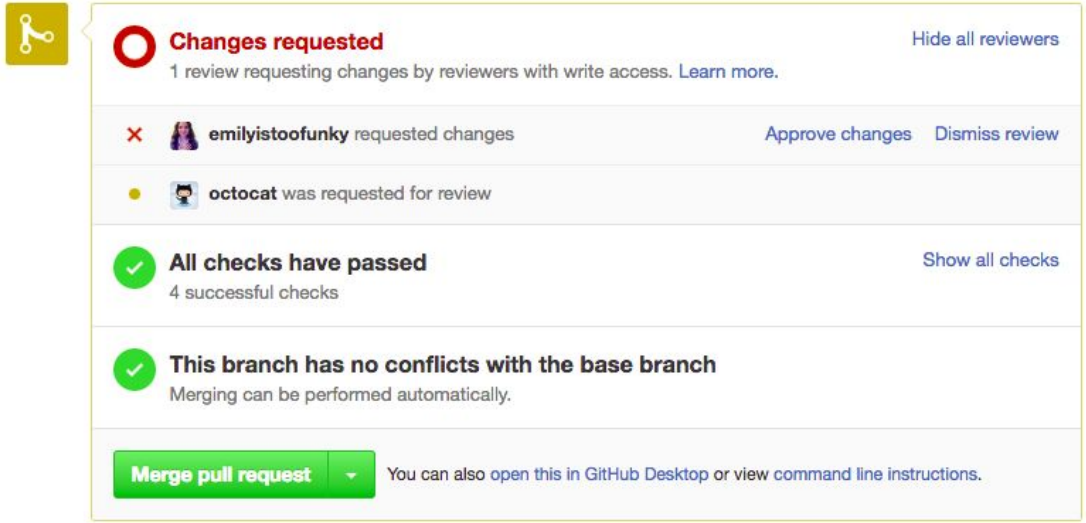
- Comment: Submit general feedback without explicitly approving the changes or requesting additional changes.
- Approve: Submit feedback and approve merging the changes proposed in the pull request.
- Request changes: Submit feedback that must be addressed before the pull request can be merged.

7. Review pull request



7. Review pull request

- You can view all of the reviews a pull request has received in the Conversation timeline, and you can see reviews by repository owners and collaborators in the pull request's merge box.



II. Other git tools

Outline

1. Gitk
2. Git gui

1. Gitk

- Name
- Synopsis
- Description
- Options

1. Gitk

- Name:
  - The Git repository browser
- Synopsis:

```
$ gitk [<options>] [<revision range>] [\--] [<path>…]
```

- Description:

  Displays changes in a repository or a selected set of commits:
  - The commit graph.
  - Showing information related to each commit.
  - The files in the trees of each revision.

1. Gitk

- Options:
  - rev-list options and arguments
  - gitk-specific options

2. Git gui

- Name
- Synopsis
- Description
- Commands

2. Git gui

- **Name:**
  - A portable graphical interface to Git.
- **Synopsis:**

```
$ git gui [<command>] [arguments]
```

- **Description:**

  Unlike *gitk, f*ocuses on commit generation and single file annotation and does not show project history.

2. Git gui

- **Commands:**
  - blame
  - browser
  - citool
  - version