Share Your Techdegree Projects With GitHub Desktop

Instruction

GitHub Desktop Reference

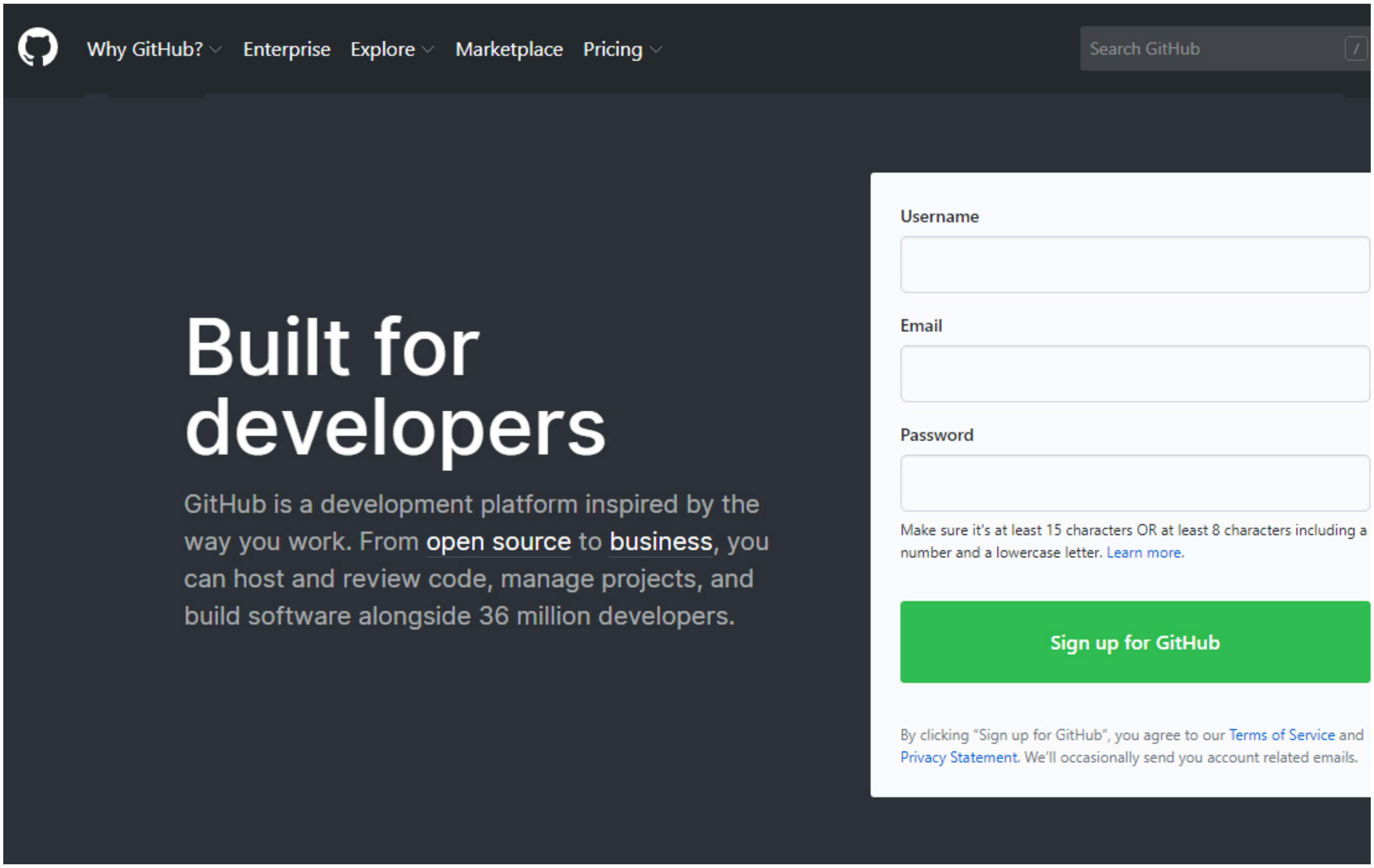
To help make using the GitHub Desktop application easier, we have created this handy reference document as a reminder of what you have learned in this Workshop. Below you will find step by step instructions for creating a GitHub account, downloading and installing the Desktop application, and using the Desktop application to create and manage repositories for your projects.

Table of Contents

Setting Up a GitHub Account

Note: If you already have a GitHub account you do not need to create a new account to store and manage your Techdegree projects. You can simply click the "Sign In" button, enter your username and password, and then proceed to the **Downloading and Installing the GitHub Desktop Application** step below.

- Start by going to github.com in your browser.
- You should see a form on the main page with a "Sign up for GitHub" button. Fill out that form and click the button. If you don’t see that form on the main page, you can also access it by clicking the "Sign Up" button located in the top-right corner of the page.



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 36 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

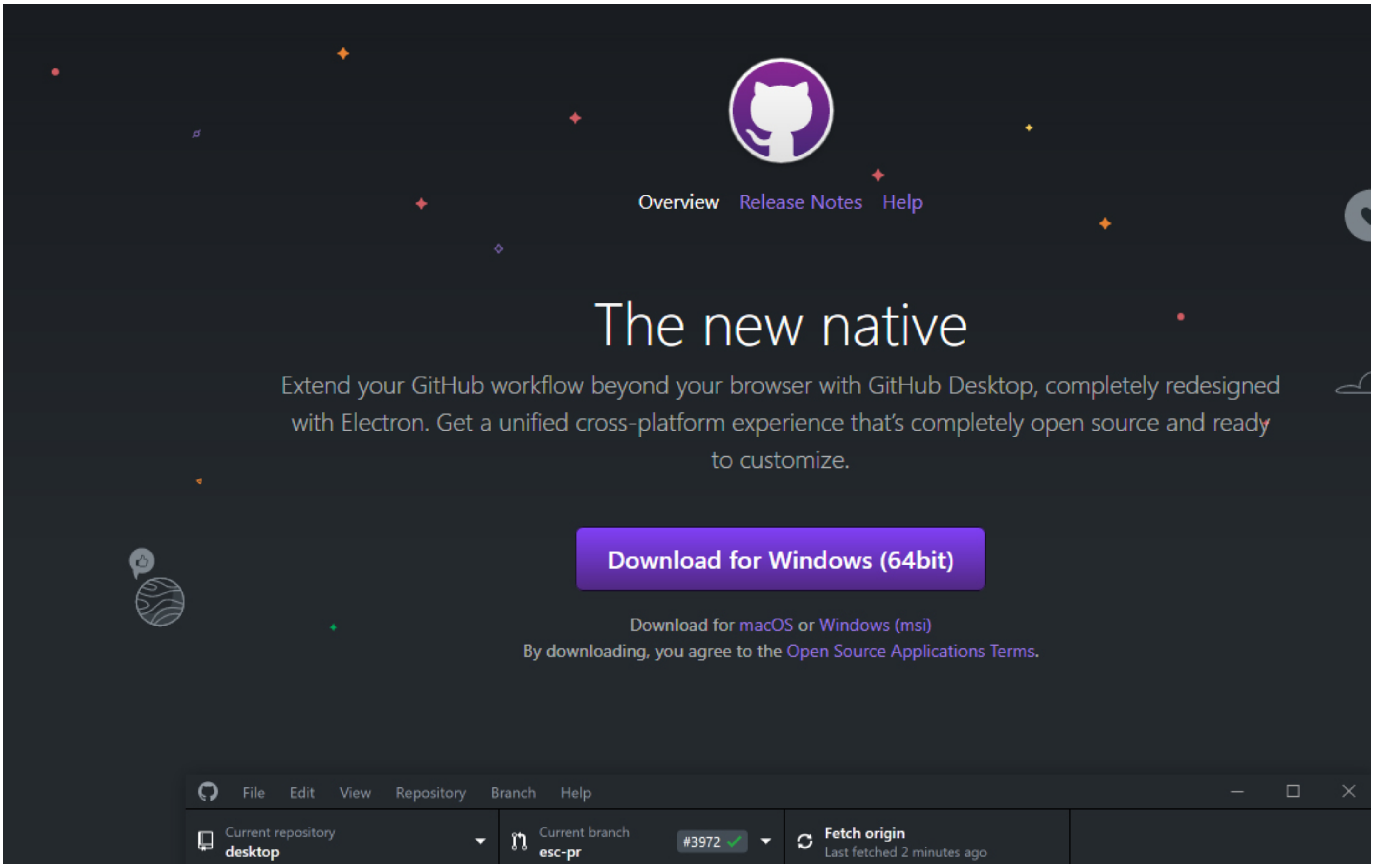
- Next you will need to choose your subscription plan. Select the Free option as it contains all of the features and tools you will need for your work in the Techdegree and click the "Continue" button.
- From there, you'll tailor your GitHub experience. You will be asked questions about your experience level and what you will be using your GitHub account for. Choose whichever options are most appropriate for you.

Congratulations! You now have your very own GitHub account.

To learn more about creating a GitHub account, check out this video: Creating a GitHub Account

Downloading and Installing the GitHub Desktop Application

- Start by going to desktop.github.com in your browser.
- Choose the appropriate option for your operating system (Windows or macOS) and click on the "Download" button. If you are unsure of your operating system, the appropriate choice will typically be shown by default.



The new native

Extend your GitHub workflow beyond your browser with GitHub Desktop, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize.

Download for macOS or Windows (msi)
By downloading, you agree to the Open Source Applications Terms.

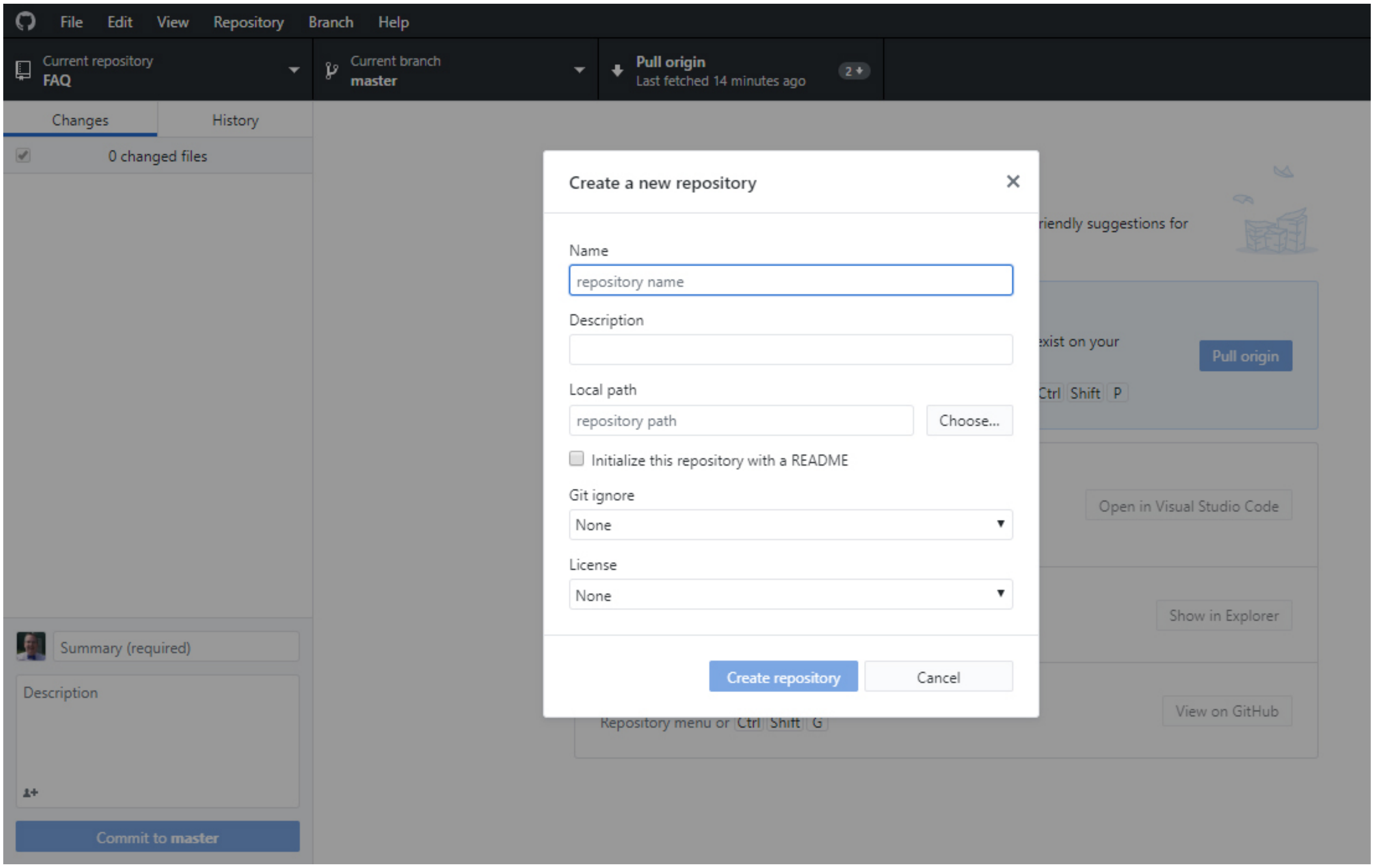
- Open the downloaded file (by default the installer will be downloaded to your Downloads folder). Opening this file should automatically install the Desktop application. The application should open once it is finished installing.
- Click on the "Sign into Github.com" option and enter the email address and password associated with your GitHub account.
- After signing in for the first time you should see a "Configure Git" screen where you will need to enter a name and email address. This information will be used to identify any commits (i.e. changes) that you make to your projects that you record with Git.
- The final step is to choose whether or not you want to send usage data to Github. Either choice is fine for the purposes of the Techdegree and won't affect how you use the application going forward. Choose your preference and click the "Finish" button.

Well done! You have successfully installed the GitHub Desktop application.

To learn more about downloading and installing the GitHub Desktop application, check out this video: Setting Up GitHub Desktop

Creating a GitHub Repository for a New Project Using the Desktop Application

- Open the GitHub Desktop application.
- Select the "New Repository" option from the "File" menu in the upper left hand corner of the page.
- This will display a form. Fill out the form as follows:
  - Enter a Name for your project.
  - Enter a brief Description of the project.
  - Enter the Local Path (the parent directory/outermost folder) of your project.
  - Check the "Initialize this repository with a README" option.
  - Select "Node" from the Git Ignore dropdown if you are working with JavaScript. Otherwise, select the language option most appropriate for your project.



Create a new repository

Name: repository name

Description:

Local path: repository path

□ Initialize this repository with a README

Git ignore: None

License: None

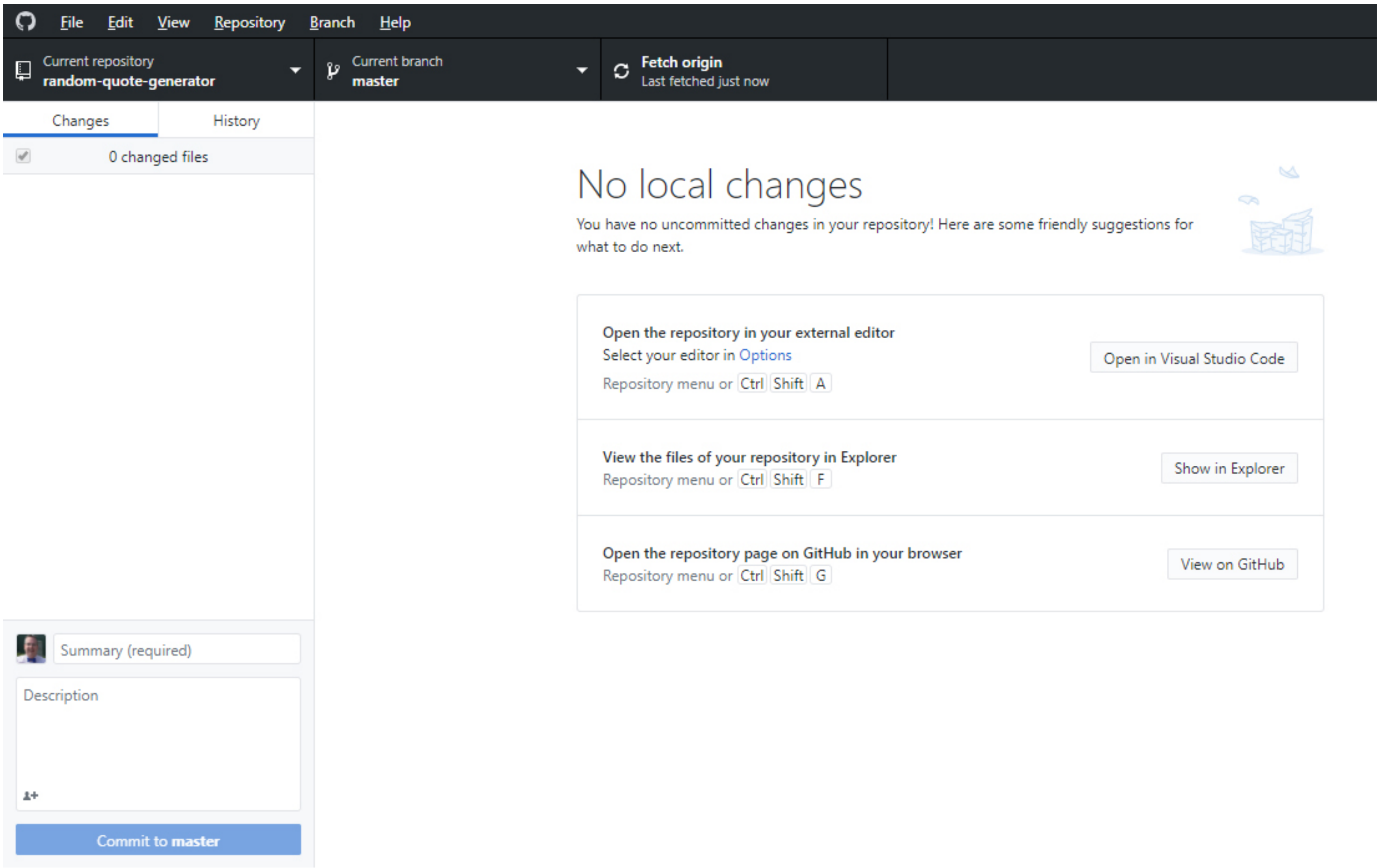
- Once you have filled out the form, click the "Create Repository" button.

Nice work! You have just created a new repository on your computer.

To learn more about creating a GitHub project using the Desktop application, check out this video: Creating a GitHub Project

Accessing Files in an Existing Repository Using the Desktop Application

- Open the GitHub Desktop application.
- Click on the "Current repository" tab, which should be at the top left of the application.
- Select the repository you would like to access.



No local changes

You have no uncommitted changes in your repository! Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A

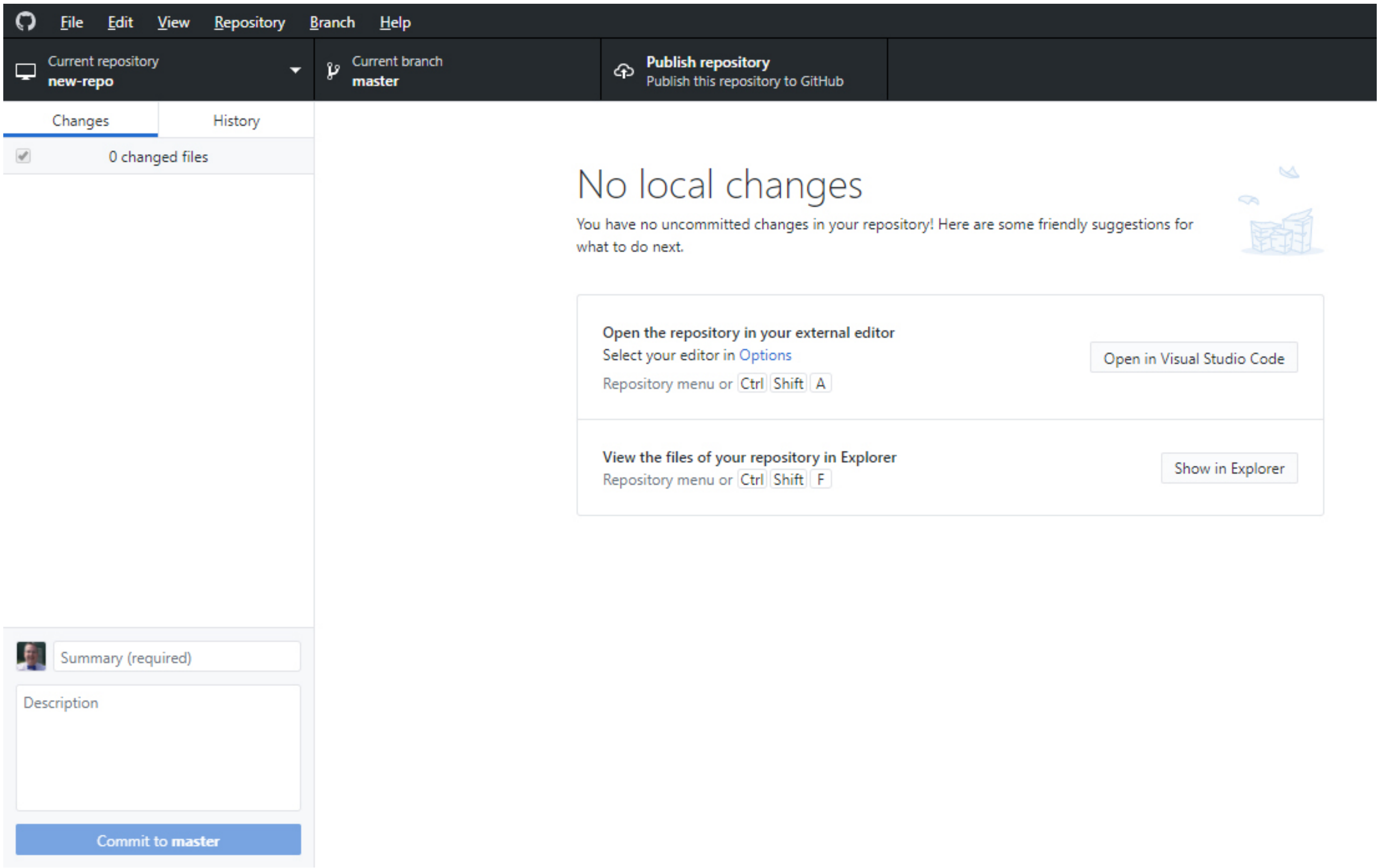
View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

- Find the section that reads "View the files of your repository in Explorer" and click on the "Show in Explorer" button. This will open a folder showing the project files for your repo. **Note**: On a Mac, the word "Explorer" will be replaced by "Finder".

Publishing a Repository Using the Desktop Application

- Open the GitHub Desktop application.
- Click on the "Current repository" tab, which should be at the top left of the application.
- Select the repository you would like to work with.
- Click on the "Publish repository" tab, which should be at the top right of the application.



No local changes

You have no uncommitted changes in your repository! Here are some friendly suggestions for what to do next.

Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl Shift A

View the files of your repository in Explorer

Repository menu or Ctrl Shift F

- A window will pop up with the repository name and description. Be sure to uncheck the box for "Keep this code private" and then click the "Publish repository" button. **Note**: If you do not uncheck the "Keep this code private" box other people will not be able to view your repository, even if you share the URL with them.

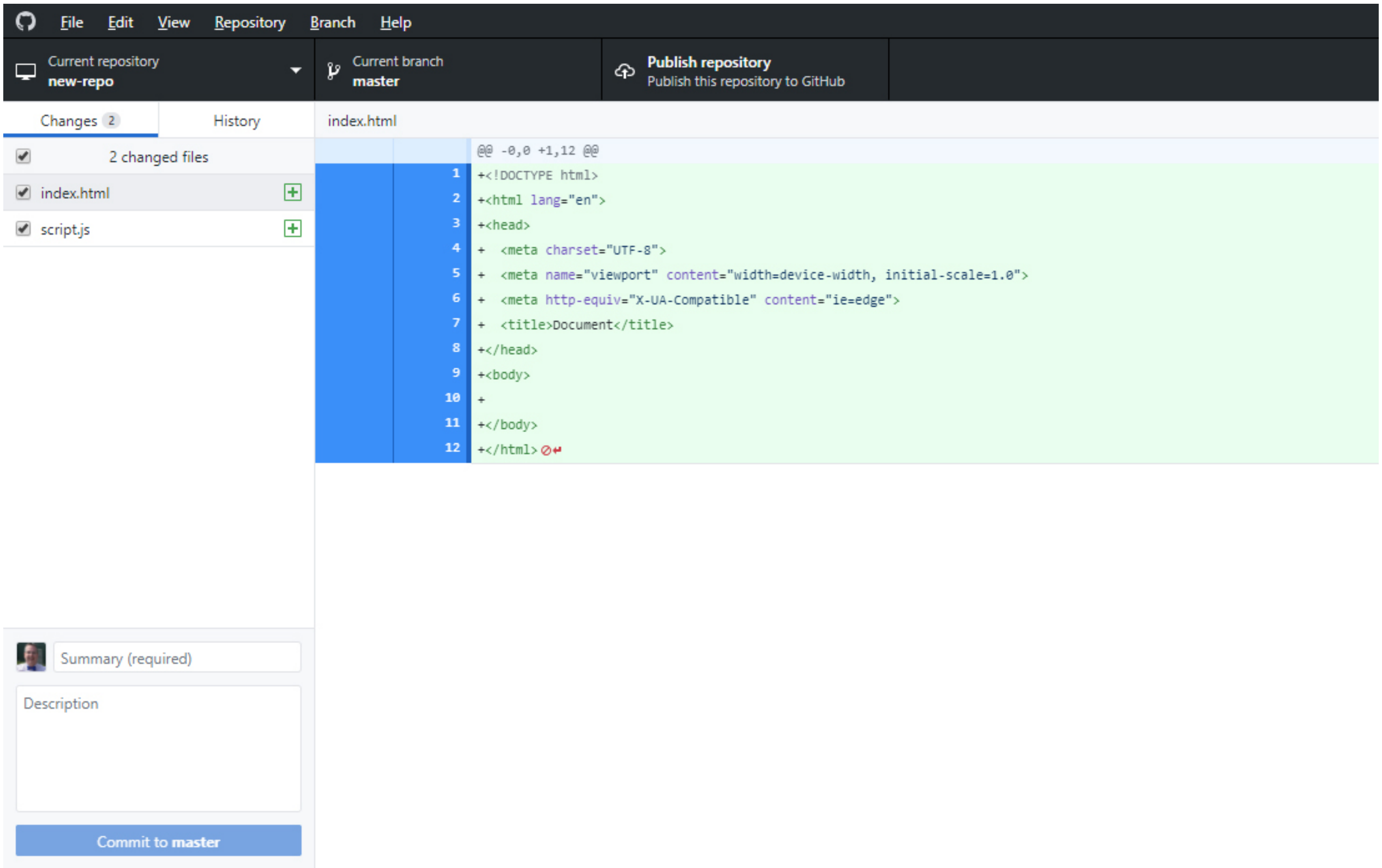
Congratulations! Your repository is now published on GitHub. You can now click on the "View on GitHub" button to open your repository in the browser.

To learn more about publishing a project using the Desktop application, check out the 4:05 mark of this video: Creating a GitHub Project

Updating a Repository Using the Desktop Application

Changes that you make to files in your local repository do not automatically get added to your GitHub repository. In order for these changes to show up in your GitHub repository, they must be "committed".

- Open the GitHub Desktop application.
- Click on the "Current repository" tab, which should be at the top left of the application.
- Select the repository you would like to update. Repositories that contain changes that have not been committed to GitHub will be marked with a blue circle.
- Once you select a repository, you will see a "Changes" tab on the left which shows a list of files that have been changed since the last commit.



```
@@ -0,0 +1,12 @@
+<!DOCTYPE html>
+<html lang="en">
+<head>
+  <meta charset="UTF-8">
+  <meta name="viewport" content="width=device-width, initial-scale=1.0">
+  <meta http-equiv="X-UA-Compatible" content="ie=edge">
+  <title>Document</title>
+</head>
+<body>
+
+</body>
+</html>
```

- Check the box to the left of the files that you would like to commit. Note that a commit can be for a single file or multiple files.
- In the summary field, enter a brief but descriptive message like "added code comments" or "created printQuote function" to summarize the changes that have been made since the last commit. You also have the option to include additional details on these changes in the Description field but this is often left blank.
- Click on the "Commit to main" button, which will record the changes to the main branch of your local repo.
- Click on the "Push origin" tab. This will update, or "push", these changes to your GitHub repository.

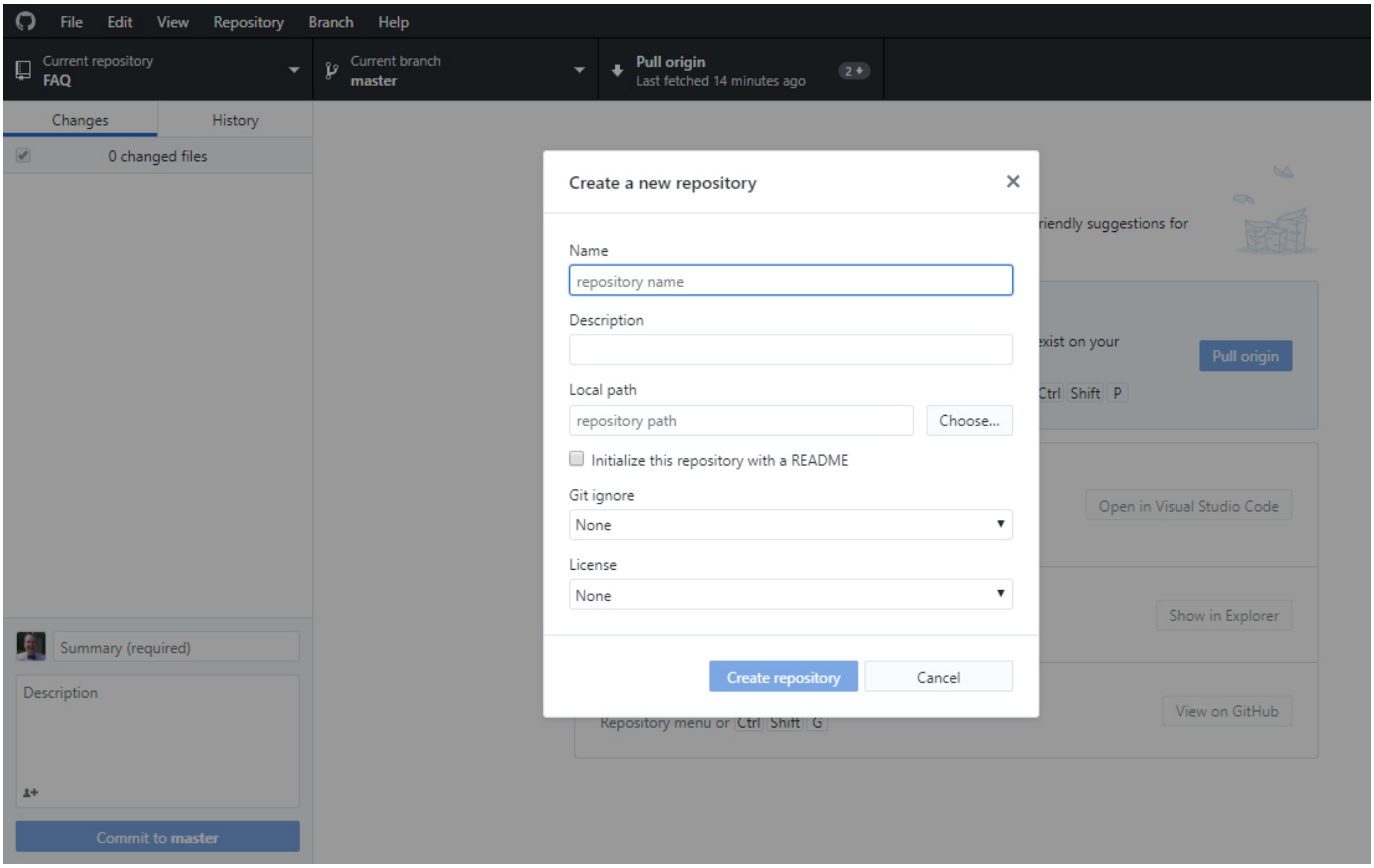
Good job! You have added files and/or changes to your project, committed them to the local repository, and pushed them to your GitHub repository.

It is a good habit to regularly make commits to your repositories as you build a project. It is also a good practice to double-check that all of your commits have been pushed before you share a link to your GitHub repository or submit your Techdegree project for review.

To learn more about using the Desktop app to update a repository, including how to add the project starter files, check out this video: Adding Changes to a Project on GitHub

Creating a Repository for an Existing Project Using the GitHub Desktop Application

- Open the GitHub Desktop application.
- Select "New Repository" from the File menu, which will open the "Create a New Repository" window.



Create a new repository ✕

Name

repository name

Description

Local path

repository path Choose...

☐ Initialize this repository with a README

Git ignore

None ▾

License

None ▾

Create repository Cancel

- Fill out the form as follows:
  - Enter a Name for your project. **Note**: The name of this new repository must be **exactly** the same as the name of the existing project folder.
  - Enter a brief Description of the project.
  - Enter the Local Path (the parent directory/outermost folder) of your project.
  - Check the "Initialize this repository with a README" option.
  - Select "Node" from the Git Ignore dropdown if you are working with JavaScript. Otherwise, select the language option most appropriate for your project.
- Once you have filled out the form, click the "Create Repository" button.
- After you have created a repository for your existing project, you will then need to publish the repo using the instructions in the **Publishing a Repository Using the Desktop Application** section of these instructions.

Bravo! You have created and published a repository for your existing project.

To learn more about creating a repository for an existing project, check out the 4:34 mark of this video: Adding Changes to a Project on GitHub

---

The GitHub Desktop application is a convenient and useful tool but it is best used as an entry point for working with Git and GitHub. As you continue on your coding journey and start to become familiar with the command line, you should consider transitioning away from the Desktop App and work on learning to create and manage your repositories using the command line.

Ok, I got it

- Questions?1
-

  Kevin Guisbert

  Front End Web Development Techdegree Student 2,972 Points

  ## Images are not displayed

  Posted on Sep 19, 2020 by Kevin Guisbert

  Kevin Guisbert

  Front End Web Development Techdegree Student 2,972 Points

- Development Tools
- GitHub Desktop Reference

0 Answers

View all discussions for this video

Related Discussions

Have questions about this instruction? Start a discussion with the community and Treehouse staff.

Get Help