**GITHUB**

1. **What is Github? What is using GitHub for? And the terms need to know**

GitHub is a project version and code management system that works like a social network for developers. But how to use GitHub? It is used to collaborate with multiple people, from anywhere in the world, to plan, track, and work on a project.

GitHub is also the largest online storage platform in the world for multi-person projects.

**What is git?**

First, we need to know what Git is first, because it is the heart of GitHub. Git is a version control system developed by Linus Torvalds (which is a familiar name, the creator of Linux).

**So, what is version control system?**

When developers create a new project, they will need to constantly update the source code. Even when the project is published, they still need to update it, fix bugs, add features, yes.

The version control system will help monitor changes in the code. Moreover, it also records who changes what can to restore the old code was deleted or the code has been corrected.

Codes are not overwritten because Git stores multiple copies of the repository in its repository. If you like learning more about Git, you can read more here (in English).

**What is a hub?**

If Git is the heart of GitHub, Hub is the soul of it. The Hub in GitHub is where the command line, Git, is turned into a huge social network for developers.

In addition to contributing to major projects, GitHub also allows users to interact in the style of social networks. You can follow, and see what people you like to do, who they are connecting with, yes.

**Repository**

Repository or repo is a library where the project files are stored. It can be located in the GitHub repository or in the local computer's repository. You can contain code files, images, audio or everything related to the project in a repository.

**Branch**

Branch is a copy of the repository. You can use Branch to deploy the project in isolation without affecting the main project.

Working with branches will therefore not affect the main reposting or other branches. If you're done, you can "Merge" the branch into other branches or the main repository by using the PullRequest command.

**Pull Request**

Pull request means that you notify others that you have pushed Branch changes to the master repository. Collaborators of this repository will accept or reject this request. Once it is open, you can discuss and review the work with other co-workers.

The steps to making a pull request in GitHub are:

1. Go to the repository and find the branch menu

2. In the branch menu, select the branch that contains your changes

3. Click the New pull request button next to the branch menu

4. Add a title and description to your pull request

5. Click the Create pull request button

**Fork a Repository**

Forking a repository (forking a repository) means you create a new project based on the old project. That is, completely replicate an existing repository, make the necessary changes, and save this new version as a completely new independent repository and call it your own project.

This feature is extremely convenient to accelerate the project. As a completely new project, the main repository will not be affected. If the repostiory master is up to date, you can also apply those updates to your fork.

The steps to fork a repository in GitHub are:

1. Find the repository you want to fork.

2. Click the Fork button.

**GitHub is not just for programmers**

GitHub is a great platform that has changed the way developers work. However, anyone who wants to manage projects effectively can join GitHub and collaborate on projects to achieve the highest efficiency.

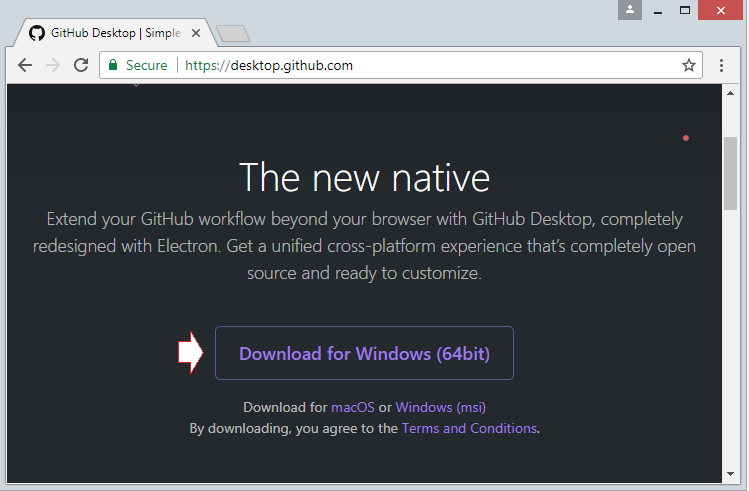
If your team works on a project that needs constant updates and needs to monitor all changes, GitHub is for you. Other similar solutions to GitHub are GitLab and BitBucket , but we still recommend using GitHub.

1. **Download & install Github Desktop**

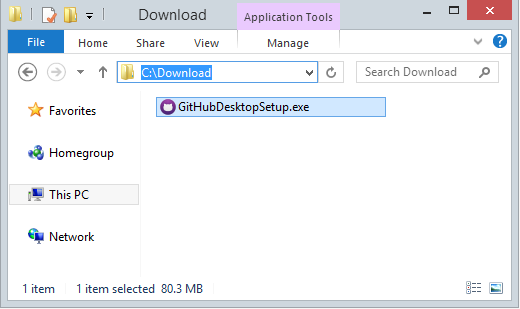
**GitHub Desktop** is essentially a visual tool that allows you to manage the **Local Repository** on your computer.

To download **GitHub Desktop,** go to:

<https://desktop.github.com/>



Download results:

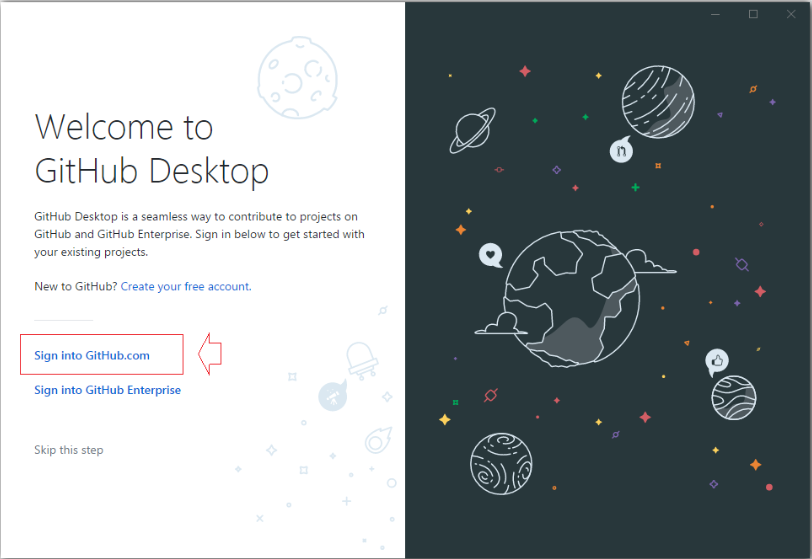


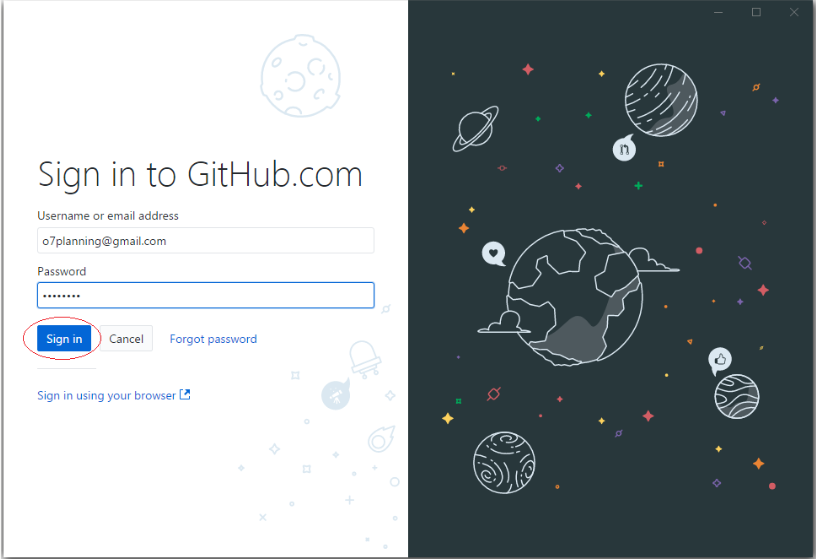
After downloading, you need to install **GitHub Desktop** on your computer:

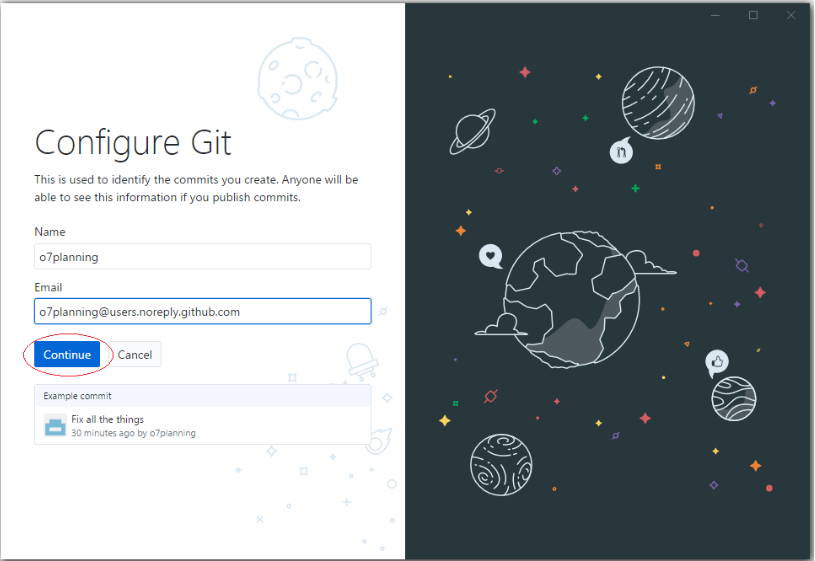
**GitHub Desktop** has been successfully installed.

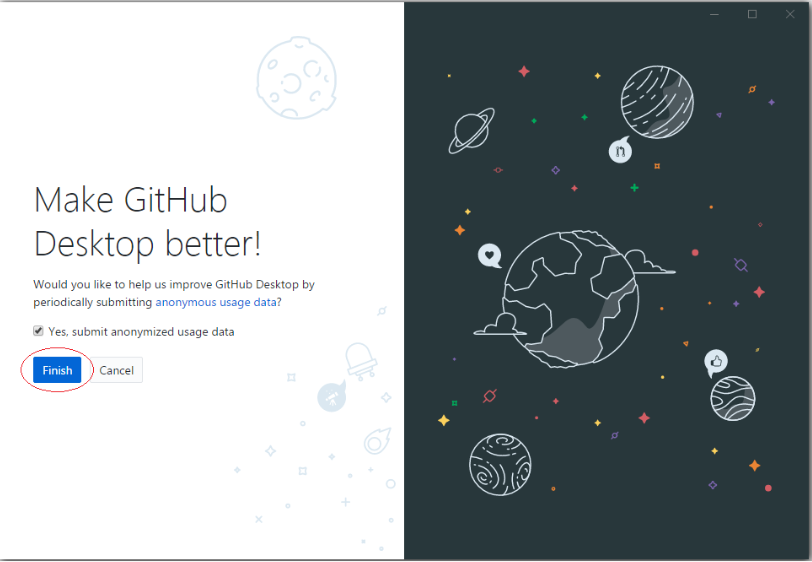
1. **Run Github Desktop**

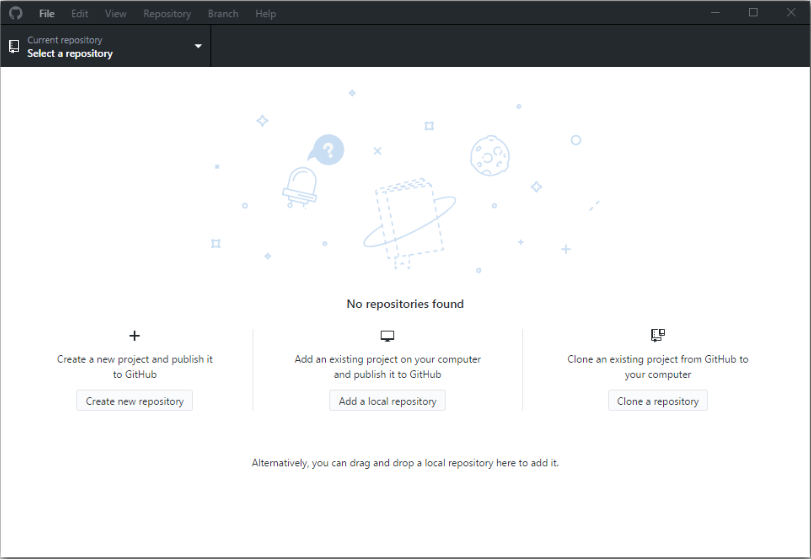
Log on **GitHub Desktop** to connect to your **GitHub** account.









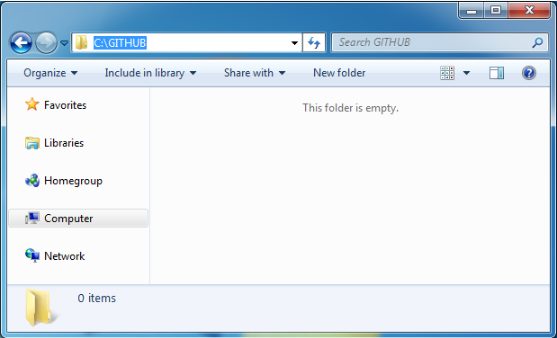


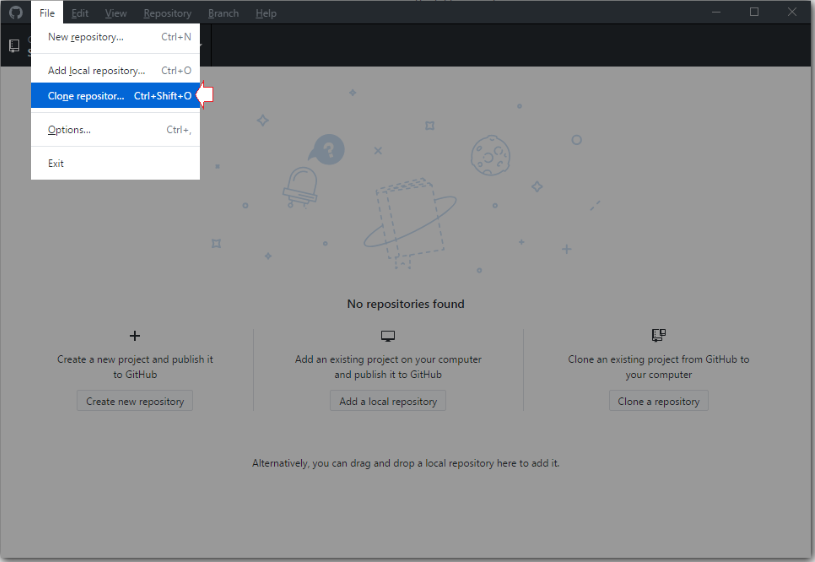
1. **Connect Github and Github Desktop**

Firstly, you need to choose an empty directory to be the location containing the local data.

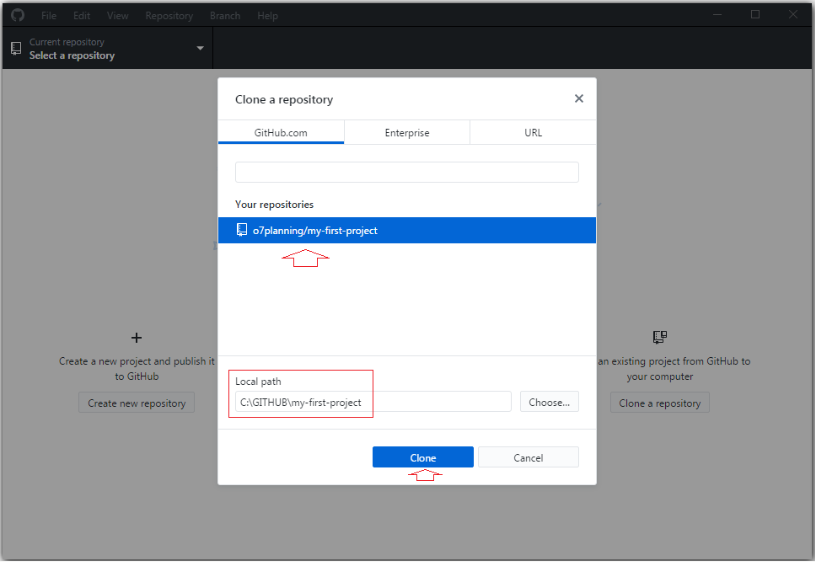
Such as:

**C: / GITHUB**

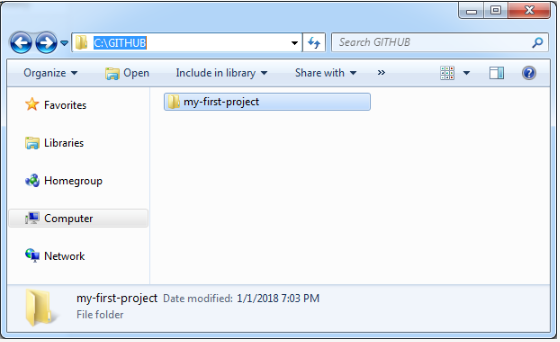


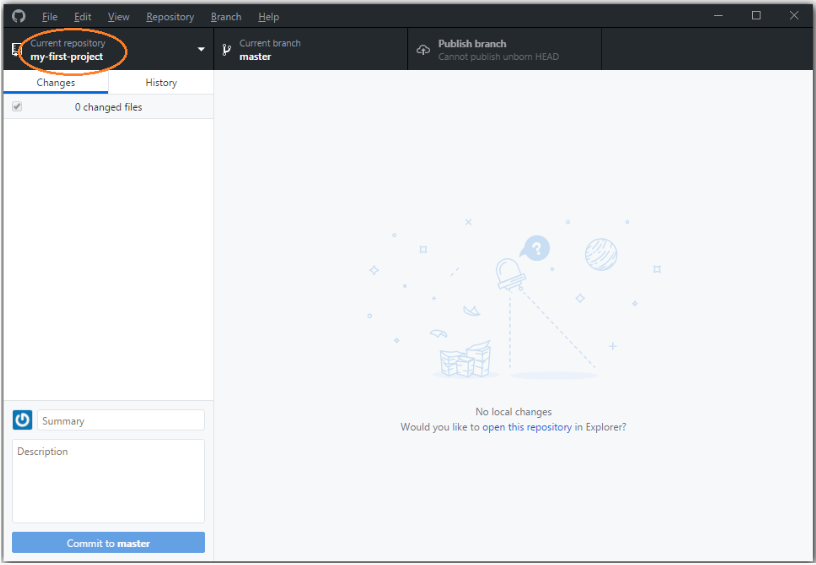


On **GitHub Desktop**, select a **Repository** you created on **GitHub** to **clone** (Make a copy) into a copy on your local computer.

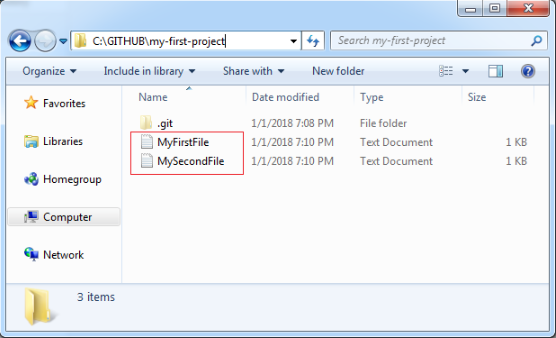


Now on **GitHub Desktop** you will see a **Local Repository** has been created.

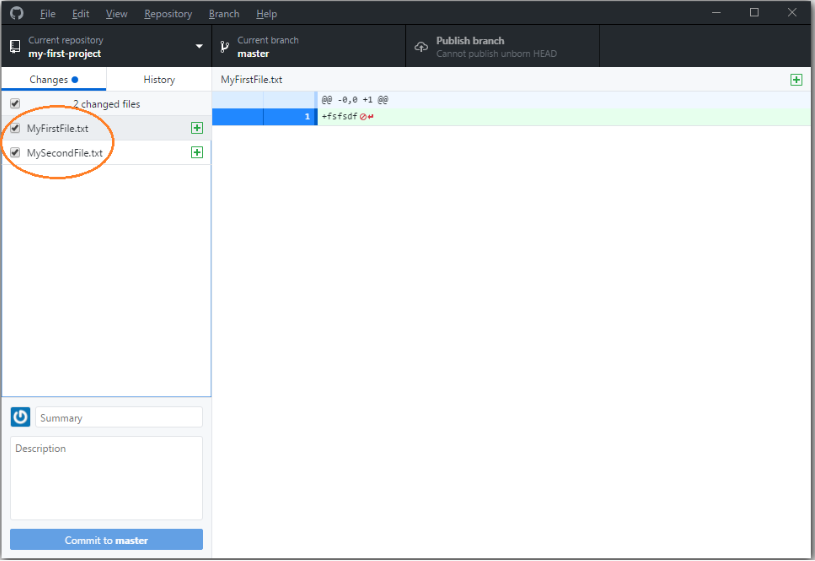




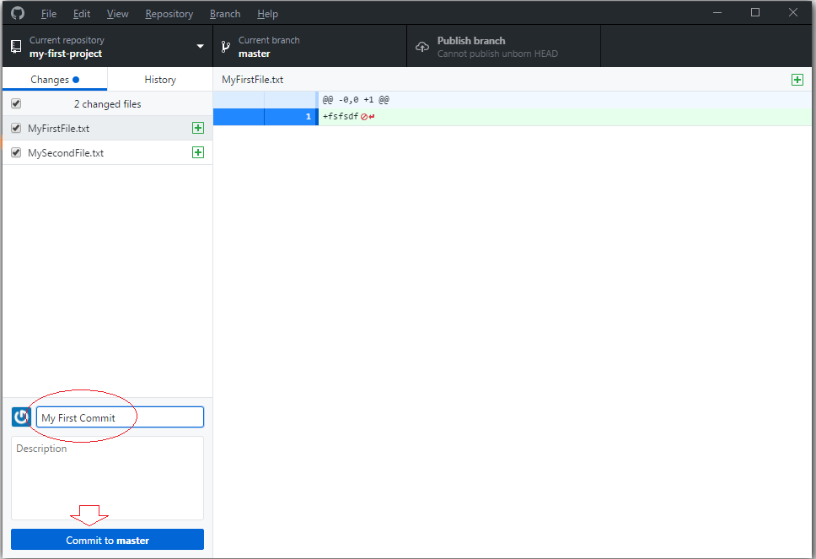
Copy some of your data files to the **Local Repository**:

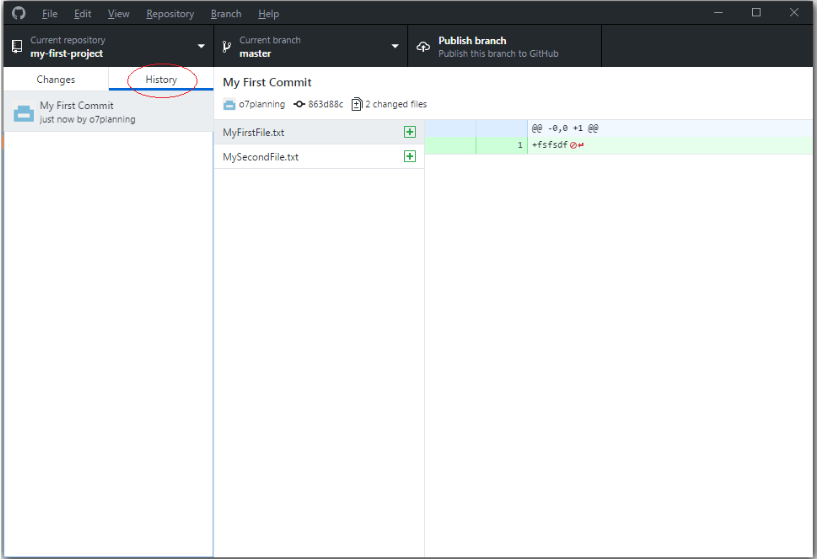


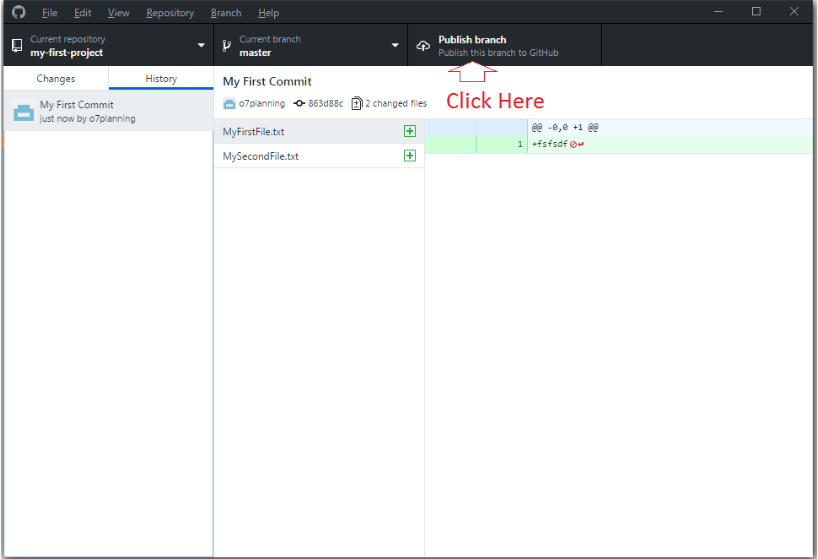
**GitHub Desktop** immediately recognizes changes at the **Local Repository**.



Enter the comment information (Comment) and click **Commit** data.







The data files you can see on the **Server**.

