



How to Make a Commit in git and Host source files in GitHub

Table of Contents

Background.....	2
Overview	2
Joining GitHub (it's free!)	2
Installing git	2
Downloading a text editor	2
Section 1: Making a commit	3
Creating a local git repository	3
Adding a new file to the repository	4
Staging a file	4
Committing a file	4
Section 2: Hosting a source file.....	5
Creating, pushing, and hosting a file.....	5
Section: 3: Viewing the source file in GitHub	5
Section 4: Unable to view source file	6
Additional Resources	6

Navigation Tip: If you click a hyperlink to view a different section within this procedure, and wish to return to your previous location in the document, **press alt+ left back arrow**.



Background

GitHub is an open source repository for developers and organizations to seamlessly host and review code, manage projects, and build software together. **Git, or git**, is system used to track incremental revisions made to source files using the command line. **Git** and **GitHub** are terms often erroneously used interchangeably, but the two have distinct functions. It's best to view **GitHub** as a giant container or folder used to easily store millions of small-to-large-scale projects, and **git** as the tool used to record the changes made to these projects.

WARNING: Proprietary and sensitive information should never be shared in the public **GitHub repository (repo)**. Developers who wish to keep their source files secure can pay for a private account.

Overview

This procedure outlines the steps required to make a **commit** and **host** your source files in the **GitHub repo**. This procedure is also written from a Windows OS user's perspective. Terms related to **git** and **GitHub** are bold.

Before making your first **commit** in **git** and **hosting** source files in **GitHub**, refer to sections: [joining GitHub \(it's free!\)](#), [installing git](#), and [downloading a text editor](#).

Joining GitHub (it's free!)

For MAC and Windows Users:

- Join [GitHub](#)
- Set up your User Profile

Installing git

For Windows OS:

- Install [git](#)

For MAC OS X:

- Install [git](#)

Downloading a text editor

Download one of the options below:

For MAC and Windows Users:

- Download [Sublime Text](#)

For Windows OS:

- Download [Atom](#)
- Download [Notepad ++](#)



For MAC OS X:

- Download [Atom](#)
- Download [Text Wrangler](#)

Section 1: Making a commit

Commits are incremental changes to source files that are made permanent. Every time a change is **committed**, **git** keeps a record of who made the changes and when, allowing developers to have access to all revisions made throughout the entire project life cycle. Version control is useful for recalling specific changes later.

This section guides you through the process of committing your first **commit** using **git**.

Creating a local git repository

To create a new project on your local machine using **git**, you will first have to create a new directory. If you don't have much experience using the command line, for Windows users, watch this [tutorial](#). For Mac OS X users, watch this [tutorial](#). Both videos are helpful for learning how to navigate the file system on your computer using the command line. To create a local **git repo**, take the following steps:

1. Open **git**.
2. Type the **pwd** command to determine your current working directory.
3. Locate where you want to store your project.

Note: If your current working directory is on the **C: drive** and you wish to store the folder on your desktop, type **cd Desktop** to change the directory.

```
Talya@Talya-PC MINGW64 ~ (master)
$ pwd
/c/Users/Talya

Talya@Talya-PC MINGW64 ~ (master)
$ cd Desktop
```

4. Type **mkdir<filename>** to create a new directory.

```
Talya@Talya-PC MINGW64 ~/Desktop (master)
$ mkdir myGitProject
```

5. Type **cd <filename>** to get inside the folder and to add files.

```
Talya@Talya-PC MINGW64 ~/Desktop (master)
$ cd myGitProject

Talya@Talya-PC MINGW64 ~/Desktop/myGitProject (master)
$
```



6. Type the **git init** command to **initialize** the project. By typing this command, you authorize **git** to begin tracking your source files. The **git init** command also creates an empty repository to store your current project.

Adding a new file to the repository

After the **repo** has been **initialized**, you can add files inside of the folder by typing the **touch** command in **git**. To create a new file inside of the directory, take the following steps:

1. Type **touch<filename>** to create a new file.
2. Type the **ls** command to list the files.
3. Minimize but do not close **git**.
Note: You can minimize **git** by clicking on the “_” at the top-right corner.
4. Open your text editor. If you have not downloaded a text editor, refer to section: [Downloading a text editor.](#)
5. Locate and open the file previously created in **git**.
6. Add content to the file and save.
7. Maximize **git**.
8. Type the **git status** command to see the **status** of the modified source file. **Git** will acknowledge that a file has been altered by displaying the filename in red.

Staging a file

The staging environment is a holding period for tentative changes made to a source file. During this stage, changes can be undone. To **stage** a change in **git**, take the following steps:

1. Type **git add<filename>** to add the changes.

Committing a file

You are finally ready to make a **commit**! When you make a **commit**, all tentative revisions in the staging environment are made permanent. To **commit** a change in **git**, take the following steps:

1. Type **git commit -m <“commit message here”>** to **save** the changes. The **commit** message that you use should be specific and describe the changes made to the source file.

Congratulations on making your first commit, it’s a significant milestone! If you only want to use **git** on your local machine, skip Section 2. If you are interested in **hosting** your source files in the **GitHub repo**, you are right where you need to be so continue reading.



Section 2: Hosting a source file

The **GitHub repo** is useful for developers who wish to host and review code, manage projects, and build software together. Often developers host their code for reviews to resolve software bugs, fix broken code, and update old features. To **host** a file, you are **pushing** the file from your local machine to the **GitHub repo** using **git**.

WARNING: Proprietary and sensitive information should never be shared in the public **GitHub repo**. Developers who wish to keep their source files secure can pay for a private account.

This section guides you through the process of **pushing** and **hosting** source files in **GitHub**.

Creating, pushing, and hosting a file

To create, push, and host a file in a new **repo** in **GitHub**, take the following steps:

1. Go to your [GitHub account](#).
Note: If you do not have a **GitHub** account, refer to section: [Joining GitHub \(it's free!\)](#).
2. Log in to your account.
3. Click the **Start a project** button.
4. Create a filename for your repo underneath the **Repository name** field.
Note: The **Description** and **Initialize this repository with a README** is optional.
5. Click the **Create repository** button. By clicking the button, you are setting up the **GitHub repo**.
6. After clicking the **Create repository** button, choose the **Create a new repository on the command line** option and follow the **git** commands.

Congratulations on hosting your source files in the GitHub repo, it's a significant milestone!

Section: 3: Viewing the source file in GitHub

Once the file has been **pushed** to the **repo**, it's time to view your source file. To view the contents of your source file, take the following steps:

1. At the top-right corner of the **GitHub** homepage **ribbon**, click the down arrow and then click on **Your Profile**.
Note: Your **repo** filename will be displayed underneath the **Popular Repositories** field.
2. Click on your **repo** filename
Note: All pushed content is now shown in the **repo**.
3. Click on the file to view the source file. If you cannot see the contents of your file, refer to section: [Unable to view source files](#).



Section 4: Unable to view source file

If you push the file and still cannot see its contents, take the following steps:

1. Minimize but do not close **git**.
2. Open your text editor and make a minor change to the source file that is being tracked by **git**.
3. Save the file.
4. Maximize **git**.
5. Type the following **git** commands:
 - **git status**
 - **git add <filename>**
 - **git commit -m <“commit message here”>**
 - **git push -u origin master**
6. To view the contents of your source file, refer to section: [Viewing the file in GitHub](#).

Congratulations on committing in git and hosting your source files in GitHub, it's a significant milestone!

Additional Resources

Free:

[YouTube](#)

Paid:

[Lynda.com](#), an online learning library

[Udemy.com](#), an online learning resource

[Go Back to Top](#)