How To Use GitHub: A tutorial for beginners

Written by John Thomas on 13 Aug 18:06

This article is meant to explain to a novice how to use GitHub. If you are just starting out, GitHub can be overwhelming, but after this article, you should have a good understanding of what it is along with the basics about how to use GitHub.

**What is GitHub?**

If you have been reading tech news or programming blogs, then you have probably heard of [GitHub](https://github.com/). So, what is GitHub? GitHub is a web-based version control and source code management tool for programming projects (or any documents really). GitHub allows software developers to store their programming projects online, for easy access and collaboration. It also allows coders to take periodic 'snapshots' of their files as they continue to edit and update their programs. You can think of this as a checkpoint in your favorite video game. You can always go back to this checkpoint again and again. Not surprisingly, given its name, GitHub uses [Git](http://en.wikipedia.org/wiki/Git_(software)) as the backbone to its functionality.

**What is Git?**

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds in 2005. There are other flavors of SCM's but Git is widely used, and among the best that exist. Although Git is not technically needed for software development, it is an extremely useful tool. Even if you are just starting out as a programmer, using Git (or another like it) is good practice, and something that you should be familiar with.

**How do I use it?**

Now that you know what GitHub is, how do you go about using it? For someone without experience, Git can be a little confusing to deal with. Just follow these steps to get a basic understanding of how it all works.

* **Step 0.1)** Create a [GitHub](https://github.com/join)account, and install [Git](http://git-scm.com/downloads).
* **Step 1)** Use Git Bash.

When you use install Git, you can install both Git Bash (Command Prompt/Terminal) and Git GUI. Since Git is mostly used for programming, we are going to use Git Bash. You can use Git GUI, but if you are interested in programming, then you should probably strive to get comfortable with the terminal.

* **Step 2)**Create your user account on your local machine. "Configure" your account.

Once Git Bash is open, you should start by configuring your account. Just type the following 2 lines:

git config --global user.name "PutYourNameHere"

git config --global user.email "[PutYour@Email.Here](mailto:PutYour@Email.Here)"

Git will use the credentials that you entered above to interact with Git repositories. Just make sure that you use the same email in Git Bash as you did when you signed up for GitHub.

* **Step 3)** Initialize a Repository.

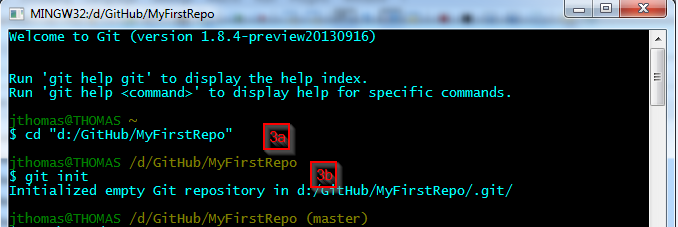
A repository is just a folder that stores all of the files associated with a project. Create a new folder on your computer to use as your project's repository. In Git Bash, navigate to that folder using the "cd" command:

cd "YourNewFolderHere"

Now, you can initialize your first repository with this command:

git init

This tells Git that the current folder you are in is now a Git repository.

[](http://www.betterwaytoweb.com/images/github1.png)

* **Step 4)** Add files to your new repository

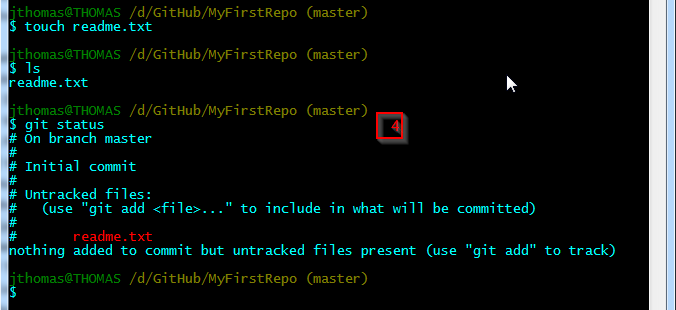
We will start by creating a simple readme file. It is good practice to have a readme file for every GitHub project. You can create a file using the "touch" command:

touch readme.txt

After you create the readme.txt file, you can verify that it was created by using the "ls" command. This command will show you all the files and folders that exist in your current folder.

Lastly, you can use the Git "status" command to check on the status of your current directory:

git status

[](http://www.betterwaytoweb.com/images/github2.png)

The output shows that Git sees the readme.txt file, but it is listed as an “untracked” file. We need to make sure Git will track this file using the "add" command:

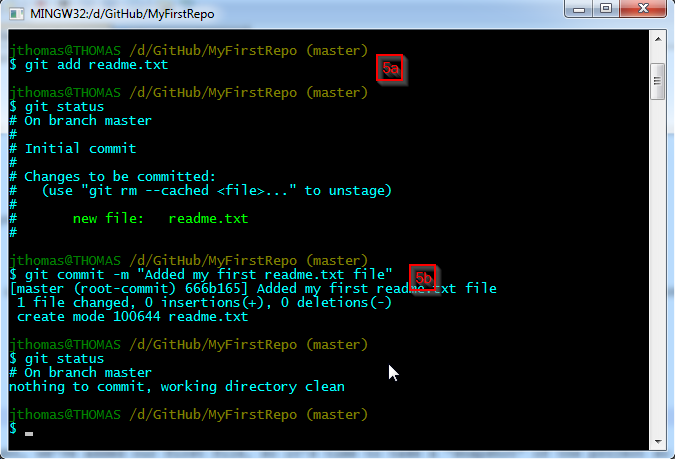
* **Step 5)** Commit the file you just created

git add readme.txt

Now that Git is tracking our file, it is time to take a 'snapshot' of the project so far. We can do this using the "commit" command:

git commit -m “Added Readme.txt”

The -m "..." portion of the command simply appends a message to your snapshot. This will allow you to keep track of which edits you made for each new snapshot.

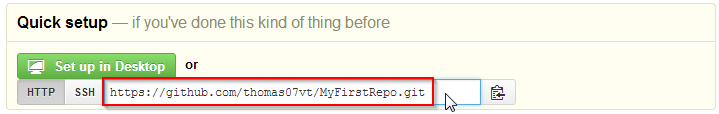
[](http://www.betterwaytoweb.com/images/github3.png)

* **Step 6)** Create a GitHub Repository

So far, we have just been working locally, but you came here to figure out how to use GitHub. GitHub allows us to "push" your local repositories to a personal web-hosted repository on github.com. After logging into GitHub, create a new repository:

<https://github.com/repositories/new>

After you have created a repository, grab the URL for your new GitHub hosted repository.

[](http://www.betterwaytoweb.com/images/github4.png)

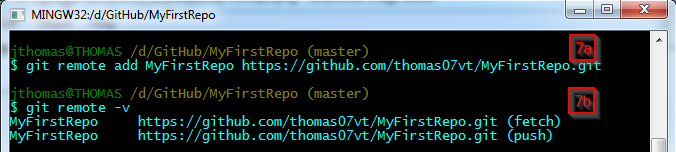
* **Step 7)** Link your local Git repository with your GitHub repository

Back in Git Bash, you will need to add your remote GitHub Repository to this project.

git remote add MyFirstRepo <https://github.com/thomas07vt/MyFirstRepo.git>

In the above command, "MyFirstRepo" is the name I assigned for this specific repository. You can, of course, name it whatever you would like. Now, you can use this command to verify that Git recognizes your first repository:

git remote -v

[](http://www.betterwaytoweb.com/images/github5.png)

You can see that the remote repository has a fetch and push feature. Basically, we will be able to send and receive files to/from the GitHub repository.

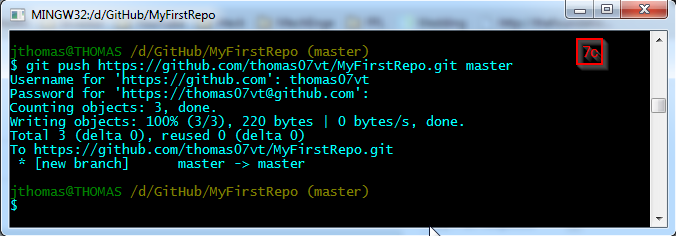
At this point, we should be able to push our files (just the readme.txt file) to the GitHub repo:

git push MyFirstRepo master

You can also use the full path to the remote repository:

git push <https://github.com/thomas07vt/MyFirstRepo.git> master

After you run this command, you will be prompted for your GitHub Username and Password. After you successfully enter those credentials, it will upload your committed files to your GitHub repository.

[](http://www.betterwaytoweb.com/images/github6.png)

You may be wondering, what the 'master' is at the end of the push command. Git allows you to have multiple branches for the same program. This allows coders to try completely different things, using the same initial source code. Since we have not created any new branches, we are on the 'master' branch.

Now, if you go to your GitHub repository at github.com, you should be able to see your readme.txt file. There you have it! You have successfully created a GitHub repository, and added a file to it.

**Errors that you may encounter:**

**Error when trying to push files to GitHub:**

! [rejected] master -> master (non-fast-forward)

error: failed to push some refs to '<https://github.com/thomas07vt/MyFirstRepo.git>'

hint: Updates were rejected because the tip of your current branch is behind

hint: its remote counterpart. Integrate the remote changes (e.g.

hint: 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

When you create a new repository on GitHub, GitHub may ask you to create a readme file. If you create a readme file directly on GitHub, then you will need to first make a 'pull' request before the 'push' request will be successful. These commands will 'pull' the remote repository, merge it with your current files, and then 'push' all the files back to GitHub:

git pull <https://github.com/thomas07vt/MyFirstRepo.git> master

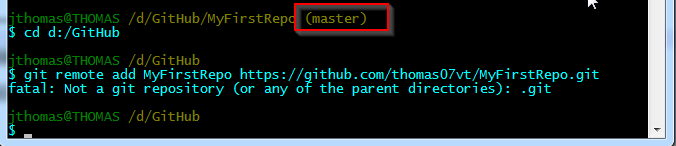
git push <https://github.com/thomas07vt/MyFirstRepo.git> master

**Error when trying to add remote GitHub repository:**

When trying to add a remote GitHub repository, you may receive this error:

fatal: Not a git repository (or any of the parent directories): .git

Make sure that your current Git Bash directory is a local Git repository. You can tell because of the branchname to the right of the directory:

[](http://www.betterwaytoweb.com/how-to-use-github-a-tutorial-for-beginners/images/github7.png)

**Tips**

If you have a large project that you want to add to a new Git repository, instead of using 'git add <filename>' for each file, simply use this command to add all files:

git add .