GIT Push and Pull Tutorial

Learn how to perform Git PUSH and PULL requests through GitHub Desktop and the Command-Line.

Jul 2019 · 13 min read

CONTENTS

SHARE

You'll be using GitHub for this tutorial as it is widely used, however, Bitbucket, Gitlab, etc. are also popular, but Developers, Data Scientists, and Data Analysts mostly use the GitHub to PUSH and do PULL Request.

You can easily follow along with all of the materials in the tutorial, even if you are a beginner. However, if you don't have any concept about Git, then have a look at **Git Tutorial for Beginners: Command-Line Fundamentals** and set up your environment by using **GIT SETUP: The Definitive Guide**.

You do need to have a GitHub account. If you don't, you can create one here.

Git PUSH

The `git push` command is used to transfer or push the commit, which is made on a local branch in your computer to a remote repository like GitHub. The command used for pushing to GitHub is given below.

```
git push 'remote_name' 'branch_name'
```

In this tutorial, you'll be looking two different ways to PUSH to GitHub.

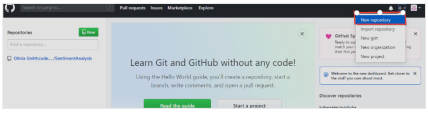
Deleting a Branch after the PULL Request is Merged
PULL Request through GitHub Desktop
Conclusion

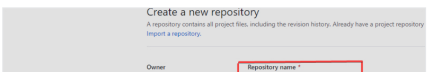
SHARE

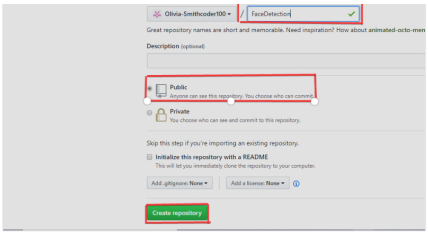
Using Command line to PUSH to GitHub

1. Creating a new repository

- You need to create a new repository and click on the plus sign.
- Fill up all the required details, i.e., repository name, description and also make the repository public this time as it is free.





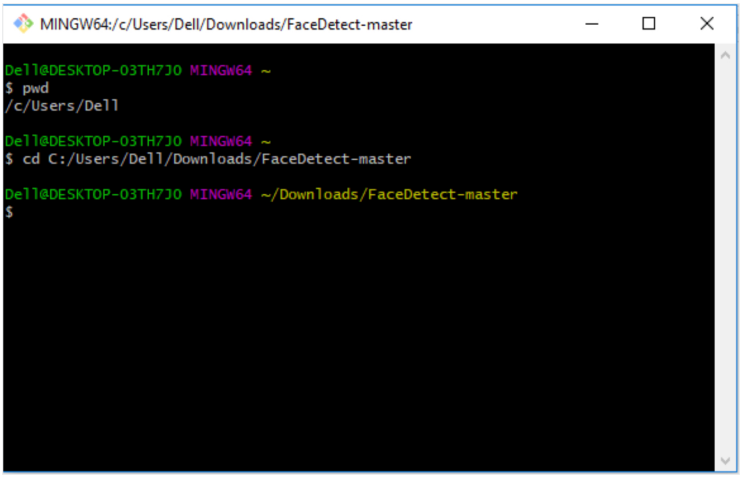


2. Open your Git Bash

- Git Bash can be downloaded in here, and it is a shell used to interface with the operating system which follows the UNIX command.

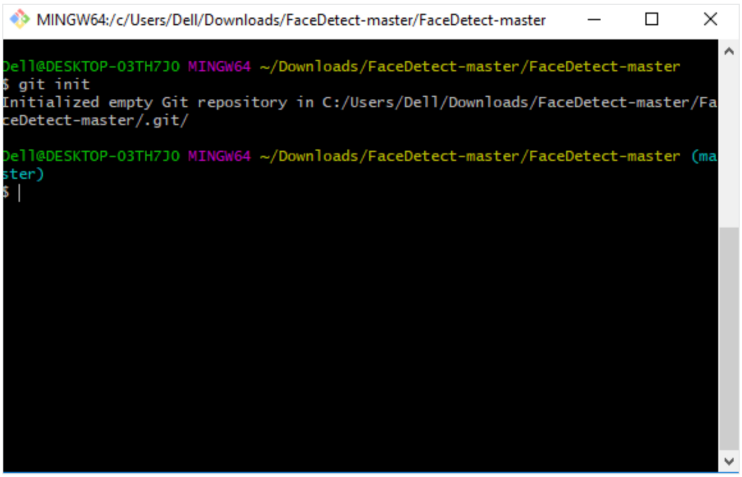
3. Create your local project in your desktop directed towards a current working directory

- `pwd` stands for 'print working directory', which is used to print the current directory.
- Move to the specific path in your local computer by `cd 'path_name'`. The cd commands stand for 'change directory' and it is used to change to the working directory in your operating system, and to locate your file, `'path_name'`, i.e., `C:/Users/Dell/Downloads/FaceDetect-master` needs to be given. This command can identify the required file that you are looking to work with.



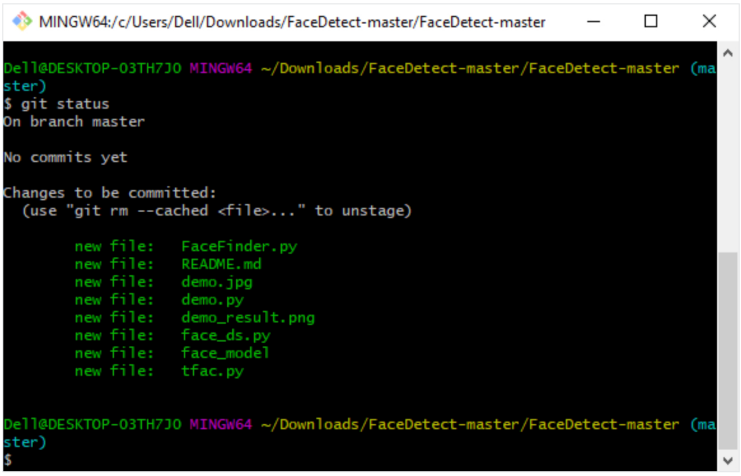
4. Initialize the git repository

- Use `git init` to initialize the repository. It is used to create a new empty repository or directory consisting of files' with the hidden directory. '.git' is created at the top level of your project, which places all of the revision information in one place.



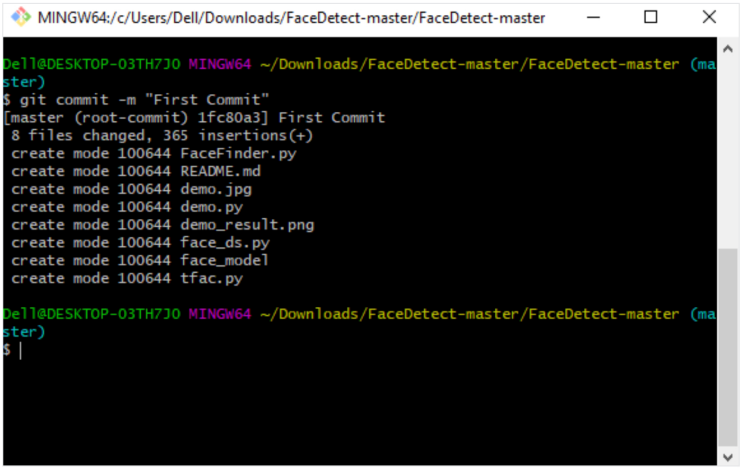
5. Add the file to the new local repository

- Use `git add .` in your bash to add all the files to the given folder.
- Use `git status` in your bash to view all the files which are going to be staged to the first commit.



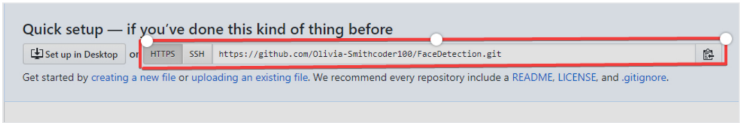
6. Commit the files staged in your local repository by writing a commit message

- You can create a commit message by `git commit -m 'your message'`, which adds the change to the local repository.
- `git commit` uses '-m' as a flag for a message to set the commits with the content where the full description is included, and a message is written in an imperative sentence up to 50 characters long and defining "what was changed", and "why was the change made".



7. Copy your remote repository's URL from GitHub

- The HTTPS or URL is copied from the given GitHub account, which is the place of the remote repository.



8. Add the URL copied, which is your remote repository to where your local content from your repository is pushed

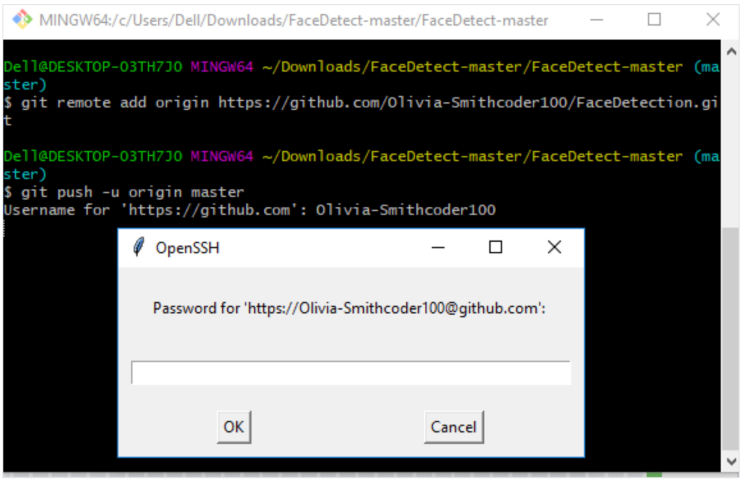
- `git remote add origin 'your_url_name'`
- In the above code, The 'origin' is the remote name, and the remote URL is "https://github.com/Olivia-Smithcoder100/FaceDetection.git". You can see the remote as GitHub in this case, and GitHub provides the URL for adding to the remote repository.

9. Push the code in your local repository to GitHub

- `git push -u origin master` is used for pushing local content to GitHub.
- In the code, the origin is your default remote repository name and '-u' flag is upstream,

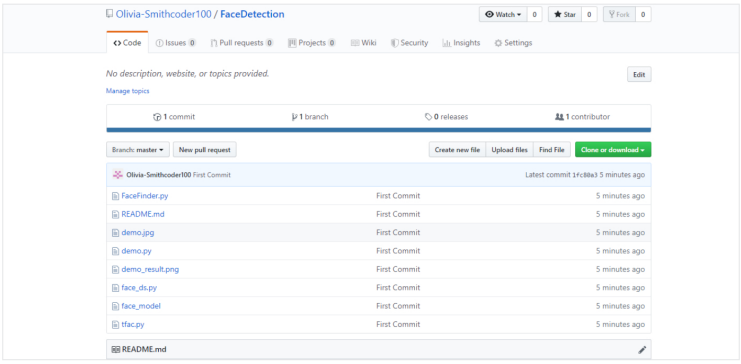
which is equivalent to '-set-upstream.' and the master is the branch, name.upstream is the repository that we have cloned the project.

- Fill in your GitHub username and password.



10. View your files in your repository hosted on GitHub

- You can finally see the file hosted on GitHub.



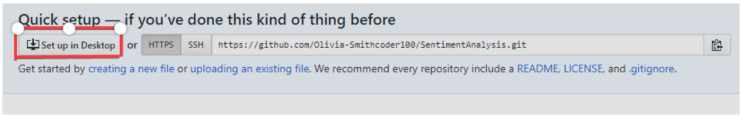
Using GitHub Desktop to PUSH to your local content to GitHub.

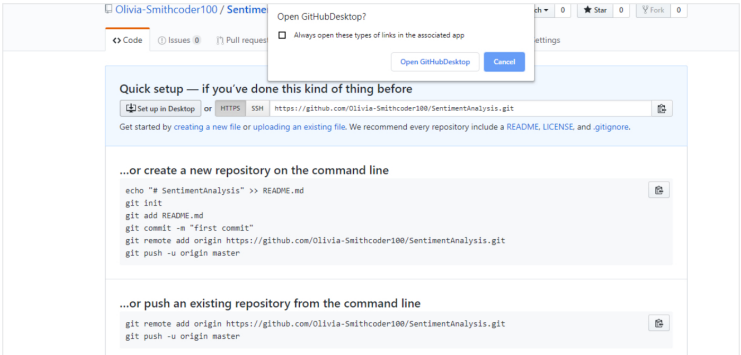
GitHub Desktop is available to download for any operating system, and it gives the GUI(Graphical User Interface) platform to push your local content from your local repository to a remote repository like GitHub.

You need to open your GitHub account in your browser and the process of creating a new repository, i.e., step 1 is the same as mentioned above in "Using Command line to PUSH to GitHub".

1. Click "Set up in a Desktop"

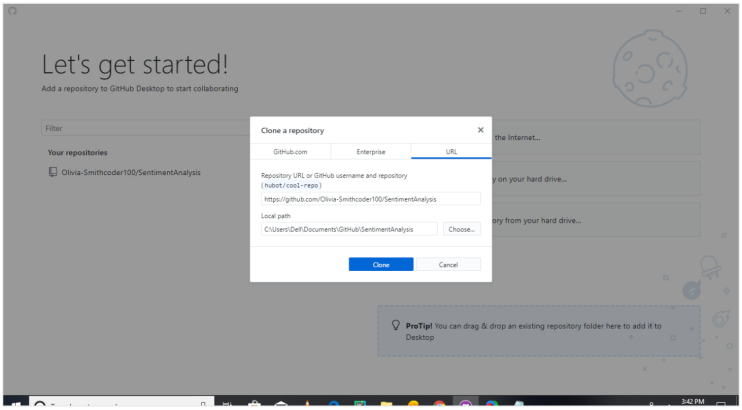
- You need to click on the button, as shown below where a pop up comes, and you click on "Open GitHub desktop".



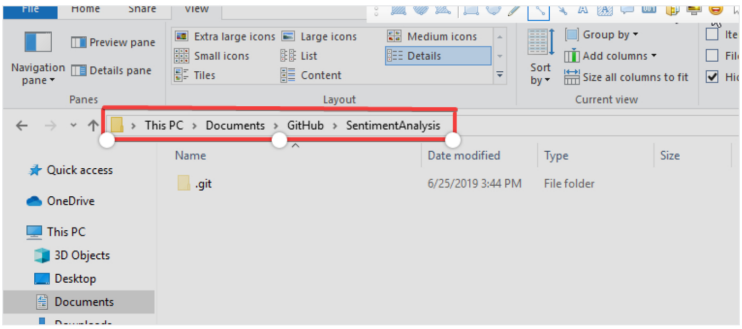


2. Cloning in a GitHub Desktop

- You can click the "Clone" button, as shown below.

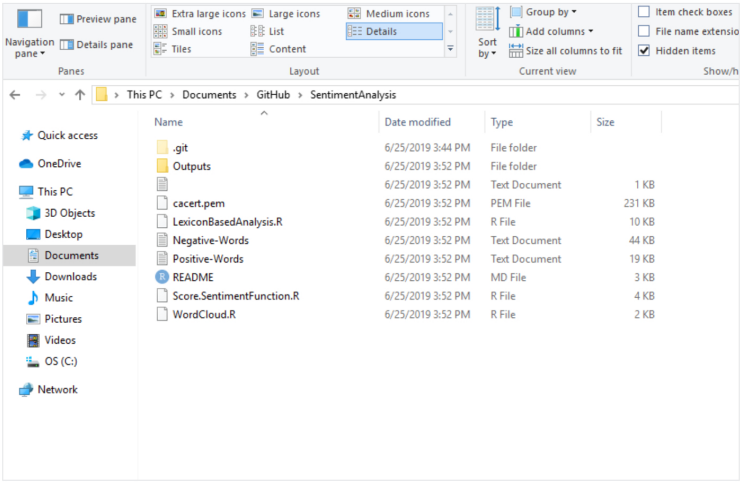


After cloning a new clone, the folder is created in your local computer where a hidden directory ".git" is also present.



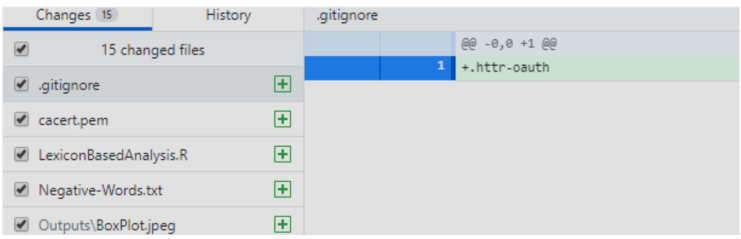
3. Copy all the required files from your local computer into the clone folder on your computer

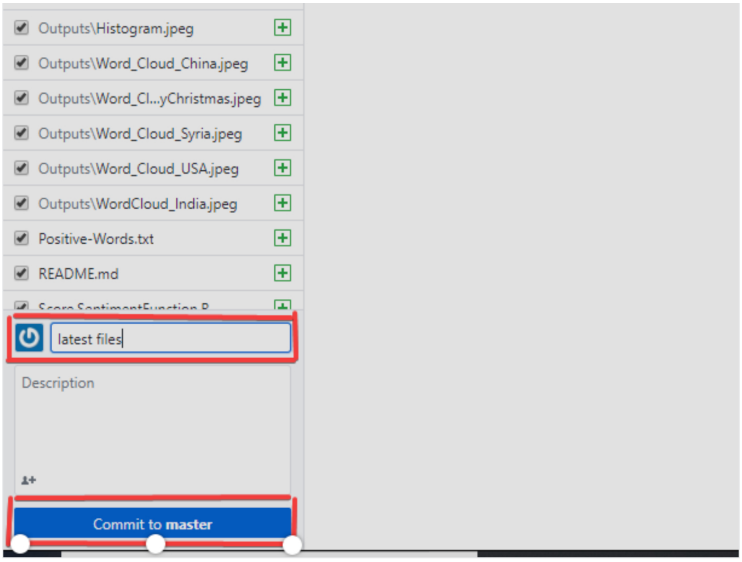
- You need to copy all the required files, images, README files, etc., to the clone folder.



4. Move to GitHub Desktop and commit to master

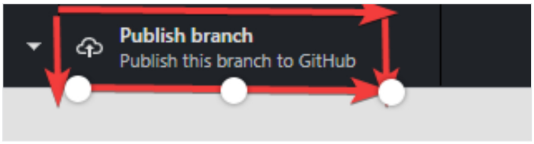
- You can see the files that are added into the clone folder are seen in GitHub Desktop too. Finally, write your message and push "Commit to master".



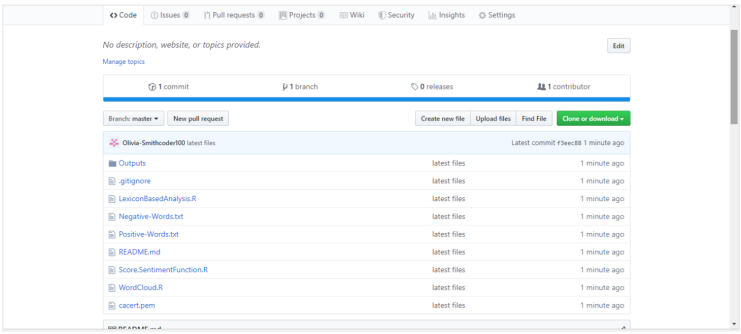


5. Publish branch in GitHub Desktop to upload your all files to GitHub

- You can click on "Publish Branch" to publish your all local content to GitHub.



You can view your repository in GitHub after you have completed all steps.



PULL Request

If you make a change in a repository, GIT PULL can allow others to view the changes. It is used to acknowledge the change that you've made to the repository that you're working on. Or also called a target repository.

The simple command to PULL from a branch is:

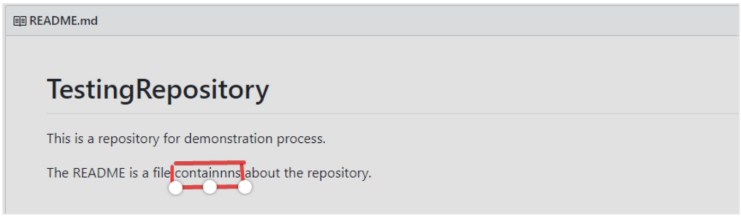
`git pull 'remote_name' 'branch_name'`.

The `git pull` command is a combination of `git fetch` which fetches the recent commits in the local repository and `git merge`, which will merge the branch from a remote to a local branch also 'remote_name' is the repository name and 'branch_name' is the name of the specific branch.

You"ll be looking at two different ways on how to use the PULL request.

PULL Request through Command Line

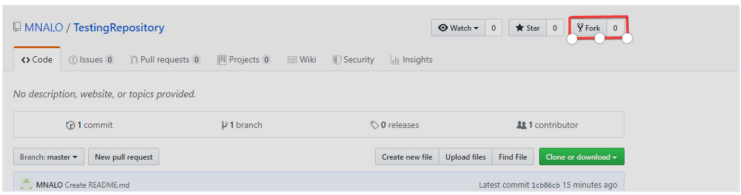
You can see the README files below which contains a typo. The README file has the word "contain" misspelled as "containnns". The owner of this repository is MNALO, and Olivia is the collaborator. She will solve the error and submit a PULL Request You'll see the process for making a PULL Request through a particular example given below.



In the file above, you can see a typo in the word "containnns".

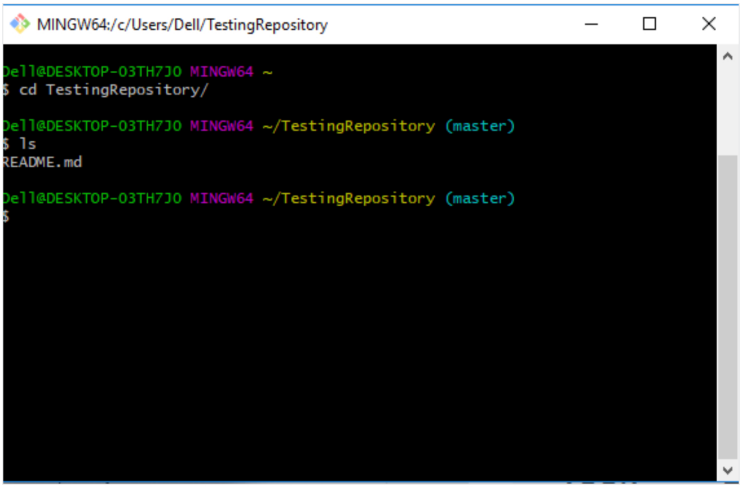
1. Fork the Repository

- "The "Fork" is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project."(Source)



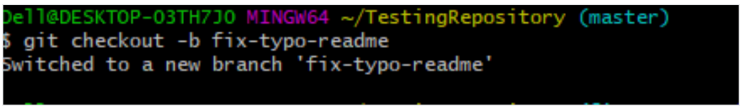
2. Open your bash in your computer

- You need to move to the required path or folder by using the `cd` command, and the content can be viewed by using the `ls` command, which will list all of the present files in the directory and in our case you can see the 'README.md' is present.



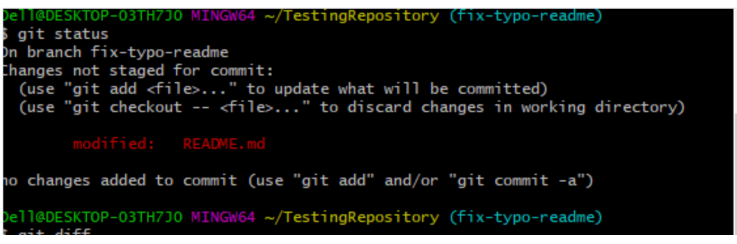
3. Make a new branch

- You can create a new branch by using the `git checkout -b 'branch_name'.` In the above code, '-b' flag is used to create a new branch, and 'branch_name' is used to give the branch a specific name, and with checkout, the branch is switched to the newly created branch.



4. Make a change by using vim from bash or direct replacement from the original README file

- You can change the word "containnns" to "contains" in the README file, and the changes with the current status can be viewed by using the following command.



```
diff --git a/README.md b/README.md
index bc04944..48fb41f 100644
--- a/README.md
+++ b/README.md
@@ -1,4 +1,4 @@
 # TestingRepository
 This is a repository for demonstration process.

-The README is a file containnns about the repository.
+The README is a file contains about the repository.
```

5. Adding and Committing a file to the repository

- You need to add and commit by the following commands.

```
Dell@DESKTOP-03TH7JO MINGW64 ~/TestingRepository (fix-typo-readme)
$ git add README.md

Dell@DESKTOP-03TH7JO MINGW64 ~/TestingRepository (fix-typo-readme)
$ git commit -m "Fix typo in README"
[fix-typo-readme 9f6cf3e] Fix typo in README
 1 file changed, 1 insertion(+), 1 deletion(-)

Dell@DESKTOP-03TH7JO MINGW64 ~/TestingRepository (fix-typo-readme)
$
```

6. Push the repository to the GitHub

- You need to push the content by `git push origin 'branch_name'`
- In the above code, the origin is the remote repository, and 'branch_name' is the required branch that you need to upload your local content.

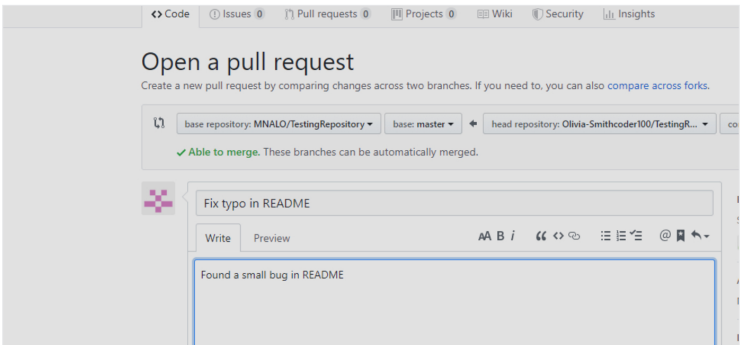
```
Dell@DESKTOP-03TH7JO MINGW64 ~/TestingRepository (fix-typo-readme)
$ git push origin fix-typo-readme
Username for 'https://github.com': Olivia-Smithcoder100
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 152.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'fix-typo-readme' on GitHub by visiting:
remote:      https://github.com/Olivia-Smithcoder100/TestingRepository/pull/new/
fix-typo-readme
remote:
To https://github.com/Olivia-Smithcoder100/TestingRepository.git
 * [new branch]      fix-typo-readme -> fix-typo-readme

Dell@DESKTOP-03TH7JO MINGW64 ~/TestingRepository (fix-typo-readme)
$
```

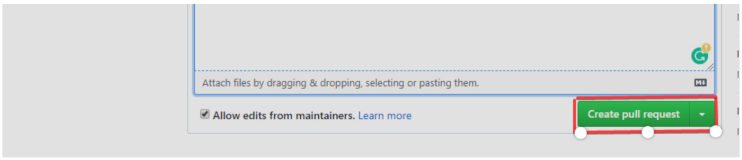
7. PULL request for a specific branch on GitHub

- You can move to your repository in GitHub and see that there is a new branch.
- You can now move to step 8, but there is a need for a local repository update with the upstream repository, read this detailed blog on How To Create a Pull Request on GitHub
- Alternatively, you can do `git pull-request` in the command line and complete the PULL Request to GitHub, where it will force push your current branch to a remote repository.

8. Open a Pull request

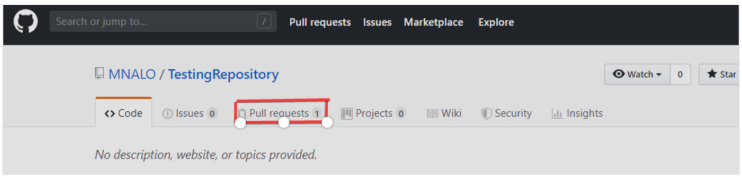
- You need to click the button on "Create pull request," to finish the action.



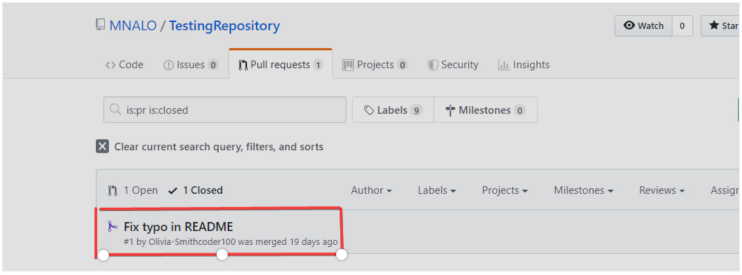


Deleting a Branch after the PULL Request is Merged

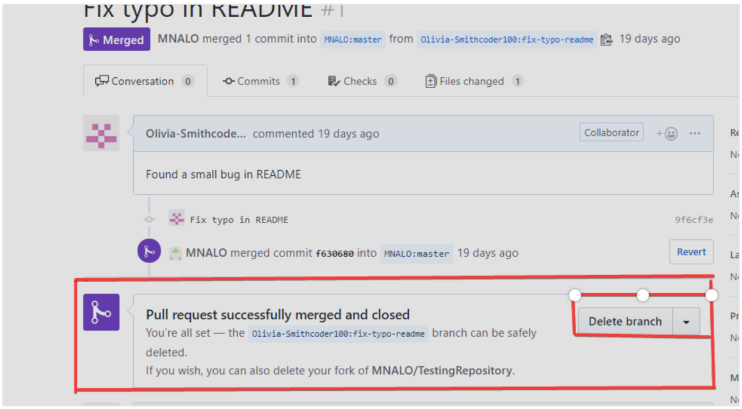
- You need to move to the main page of the repository and click "Pull requests".



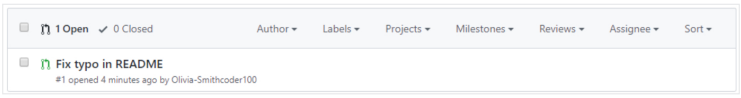
- You need to click 'Closed' to see the lists of all the PULL Requests that you've made, but there is only one at the moment which needs to be selected. It is the one related to your branch that you want to delete.



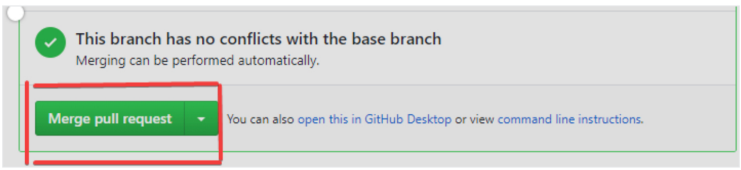
- You can now click 'Delete branch' to complete the action.

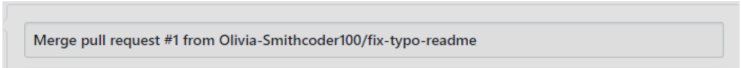


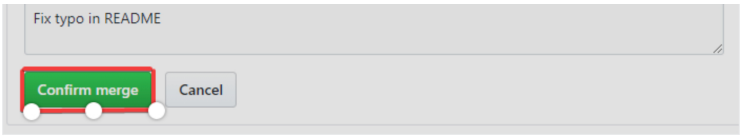
The owner of the repository can view all the commits, pull request, etc., made by collaborators and others. The changes made by someone can be significant, quick fixes for a bug, errors, etc., and are added to the project.



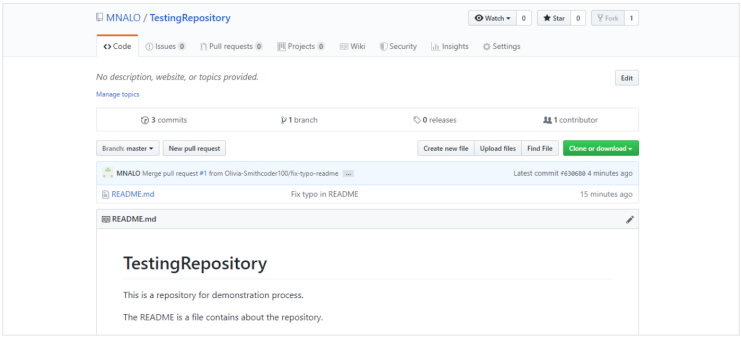
The owner now clicks "Merge pull request". Also, he/she will click "Confirm merge" through the following process.







The last change made to the README.md file with a corrected typo is below.

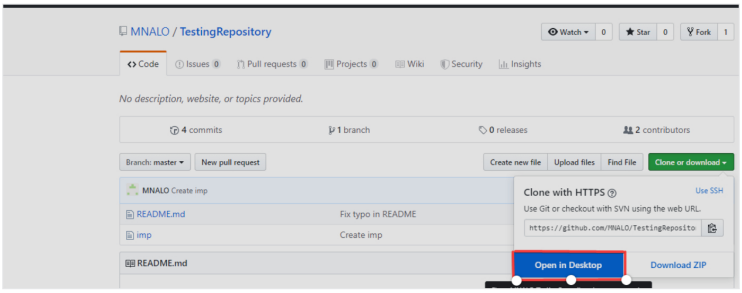


PULL Request through GitHub Desktop

The file "imp" contains a typo where MNALO is the owner and Olivia is collaborator follows the following process to create a PULL request from GitHub Desktop.

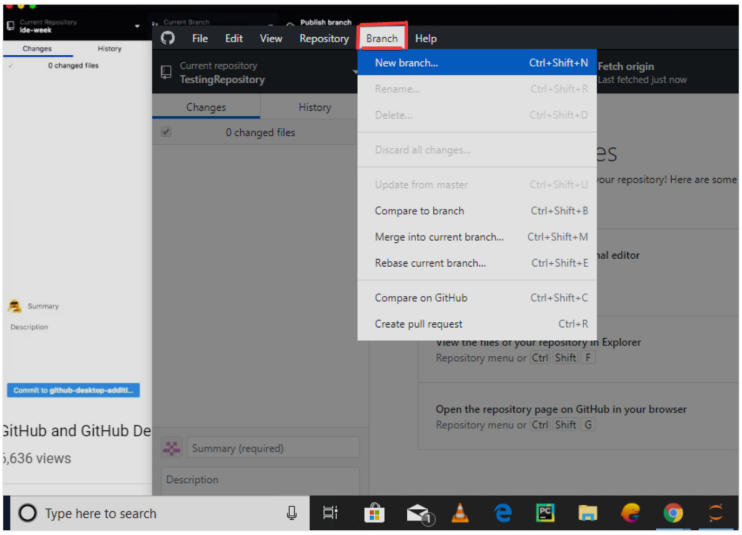
1. Cloning and Opening to Desktop

- A project is cloned and click to "Open in Desktop".



2. Create a new branch

- A new branch, "fix-typo-imp" is created.

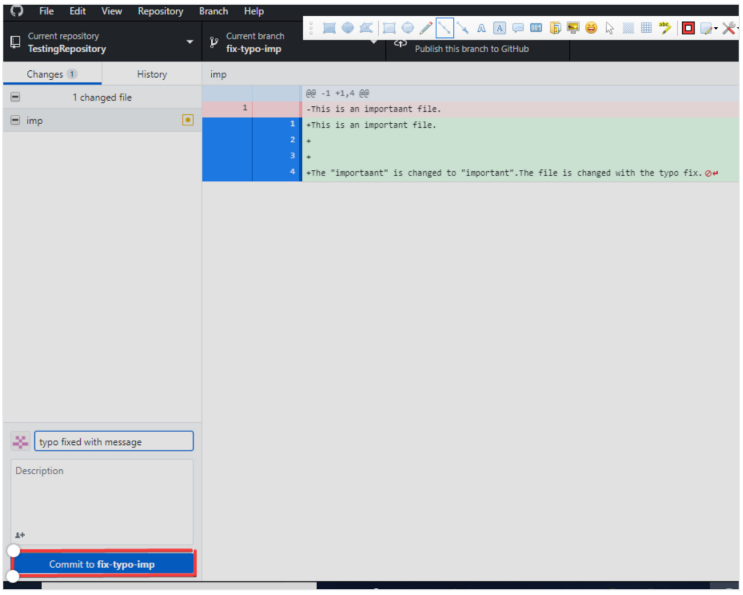


3. Make a change in the imp file from the text editor

- You can change the content of the imp file, fix a typo, and add some text.

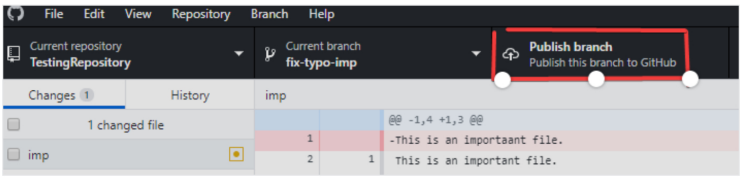
4. Commit the changes

- A commit message written and "Commit to fix-typo-imp" is clicked.



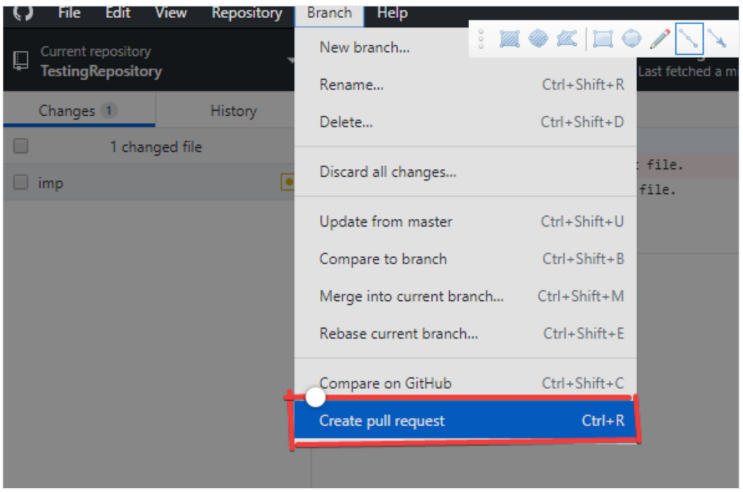
5. Publish the branch

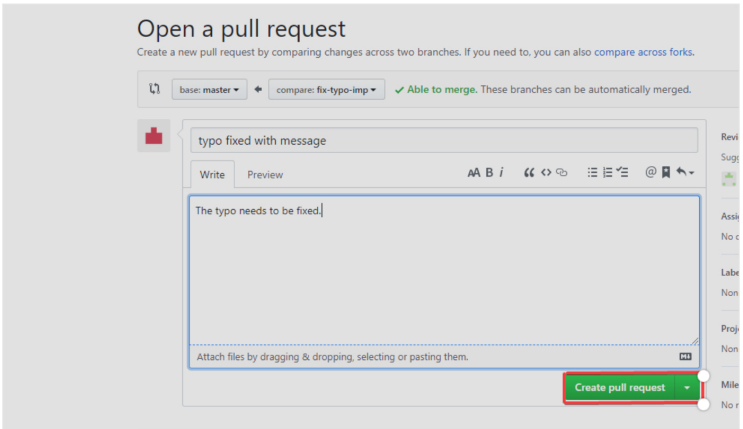
- You can now publish the branch, which pushes the commit to GitHub.



6. Create a PULL Request

- You can now make a PULL request by clicking "Create pull request".
- You can also now write a message and then click "Create pull request" again.





Olivia Smith

Senior developer at CMARIX TechnoLabs. Writes about trending technologies like AI & ML

TOPICS

Git

RELATED

What is Git? - The Complete Guide to Git

The process afterward is the same as above in "PULL Request through Command Line".

Conclusion

In this tutorial, you have learned the PUSH and PULL request and also the different ways through which the PUSH and PULL request is done through the command line and GitHub Desktop Applications.

If you would like to learn more data science skills, take DataCamp's intro courses:

- Introduction to R
- Introduction to Data Science in Python

TOPICS

Git

GIT SETUP: The Definitive Guide

Git Install Tutorial

GitHub and Git Tutorial for Beginners

Data Science Courses

Introduction to Git

Beginner · 4 hr · 127.9K

This course is an introduction to version control with Git for data scientists.

See Details → Start Course

Introduction to R

Beginner · 4 hr · 2.4M

Master the basics of data analysis in R, including vectors, lists, and data frames, and practice R with real data sets.

See Details → Start Course

Introduction to Python

Beginner · 4 hr · 4.7M

Master the basics of data analysis with Python in just four hours. This online course will introduce the Python interface and explore popular packages.

See Details → Start Course

See More →

Related

What is Git? - The Complete Guide to Git

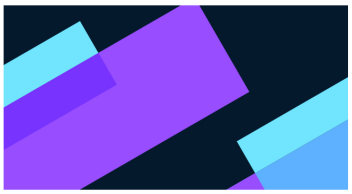
Learn about the most popular version control system and why it's a must-have collaboration tool for data scientists and...

Summer Worsley • 11 min

GIT SETUP: The Definitive Guide

In this tutorial, you'll learn how to set up Git on your computer in different operating systems.

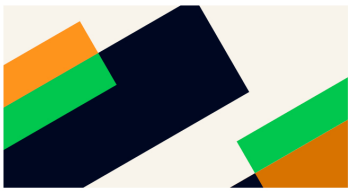
Olivia Smith • 7 min



Git Install Tutorial

Learn about Git initial setup, Git LFS, and user-friendly Git GUI applications in this in-depth tutorial.

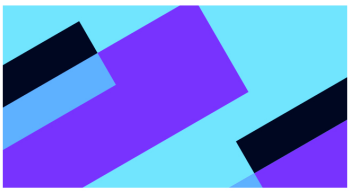
Abid Ali Awan • 9 min



GitHub and Git Tutorial for Beginners

A beginner's tutorial demonstrating how Git version control works and why it is crucial for data science projects.

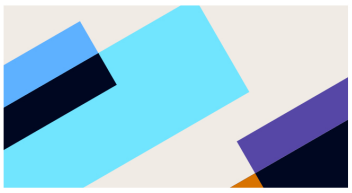
Abid Ali Awan • 17 min



How to Resolve Merge Conflicts in Git Tutorial

Learn various commands and tools for merging two branches and resolving conflicts in Git, an essential skill for data...

Abid Ali Awan • 16 min



Git Reset and Revert Tutorial for Beginners

A beginner's guide tutorial demonstrating how to use the Git Revert and Reset commands.

Zoumana Keita • 10 min

See More →

Grow your data skills with DataCamp for Mobile

Make progress on the go with our mobile courses and daily 5-minute coding challenges.