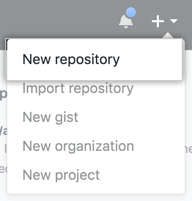
Adding an existing project to GitHub using the command line

Create a new repository on GitHub. To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project has been pushed to GitHub.



Open Git Bash.

Change the current working directory to your local project.

Initialize the local directory as a Git repository.

$ git init

Add the files in your new local repository. This stages them for the first commit.

$ git add .

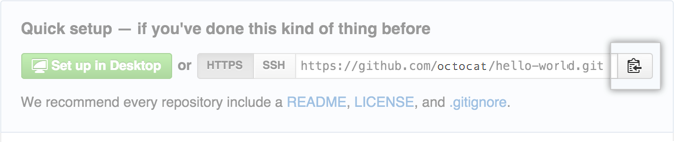
# Adds the files in the local repository and stages them for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.

Commit the files that you've staged in your local repository.

$ git commit -m "First commit"

# Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.

At the top of your GitHub repository's Quick Setup page, click to copy the remote repository URL.



In the Command prompt, add the URL for the remote repository where your local repository will be pushed.

$ git remote add origin remote repository URL

# Sets the new remote

$ git remote -v

# Verifies the new remote URL

Push the changes in your local repository to GitHub.

$ git push origin master

# Pushes the changes in your local repository up to the remote repository you specified as the origin

Adding a file to a repository using the command line

You can upload an existing file to a GitHub repository using the command line.

This procedure assumes you've already:

Created a repository on GitHub, or have an existing repository owned by someone else you'd like to contribute to

Cloned the repository locally on your computer

On your computer, move the file you'd like to upload to GitHub into the local directory that was created when you cloned the repository.

Open Git Bash.

Change the current working directory to your local repository.

Stage the file for commit to your local repository.

$ git add .

# Adds the file to your local repository and stages it for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.

Commit the file that you've staged in your local repository.

$ git commit -m "Add existing file"

# Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.

Push the changes in your local repository to GitHub.

$ git push origin your-branch

# Pushes the changes in your local repository up to the remote repository you specified

Remote origin already exists on 'git push' to a new repository

METHOD1->

Since origin already exist remove it.

git remote rm origin

git remote add origin https://github.com/USERNAME/REPOSITORY.git

METHOD2->

One can also change existing remote repository URL by ->git remote set-url

If you're updating to use HTTPS

git remote set-url origin https://github.com/USERNAME/REPOSITORY.git

If you're updating to use SSH

git remote set-url origin [git@github.com:USERNAME/REPOSITORY.git](mailto:git@github.com:USERNAME/REPOSITORY.git)

If trying to update a remote that doesn't exist you will receive a error. So be careful of that.

METHOD3->

Use the git remote rename command to rename an existing remote. An existing remote name, for example, origin.

git remote rename origin startpoint

# Change remote name from 'origin' to 'startpoint'

To verify remote's new name->

git remote -v

Delete Files

In the command-line, navigate to your local repository.

Ensure you are in the default branch:

git checkout master

The rm -r command will recursively remove your folder:

git rm -r folder-name

Commit the change:

git commit -m "Remove duplicated directory"

Push the change to your remote repository:

git push origin master