Lab 1 – Local and Remote Git Repository using Git Bash

Objectives

Learn to use GIT Bash and folder navigation

Learn to create a remote repository in GitHub

Learn to use common git commands

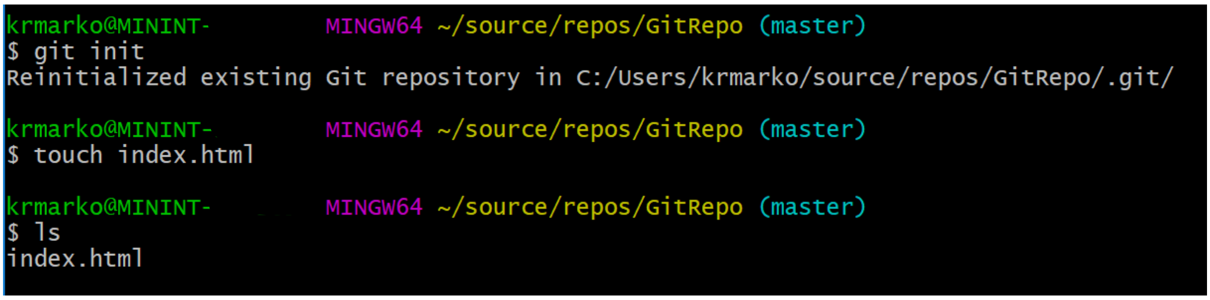
- git init
- git add
- git commit
- git status
- git log
- git push
- command parameters (e.g. -m)

Initialise Local Git Repository

Open Git Bash and using the ls and cd commands, navigate to wherever you want to create your repository (repo).

Use the cd **mkdir "folder-name"** command to create a new directory. Then once created you can go into that folder and use the **git init** command initialize your local repo.

Create index.html file in repo using the **touch** "**file-name"** command.

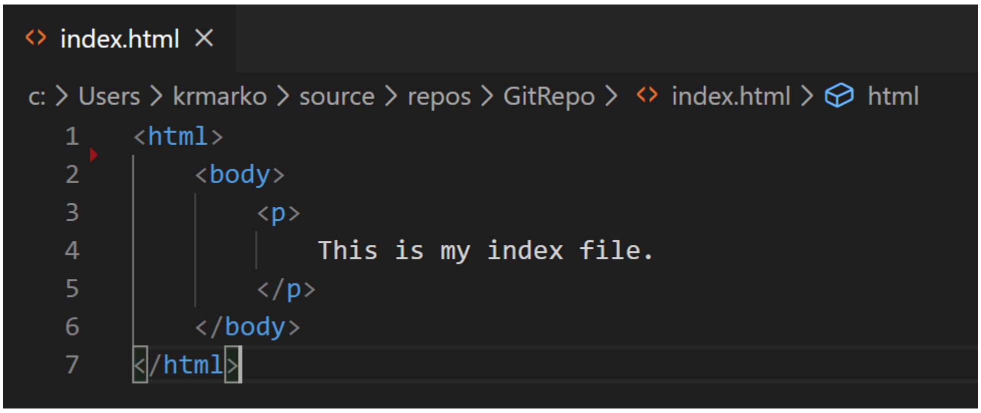


```
krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ git init
Reinitialized existing Git repository in C:/Users/krmarko/source/repos/GitRepo/.git/

krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ touch index.html

krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ ls
index.html
```

Update your file in VSCode

Navigate to your index.html file in Visual Studio Code and open it. Your file should be empty.

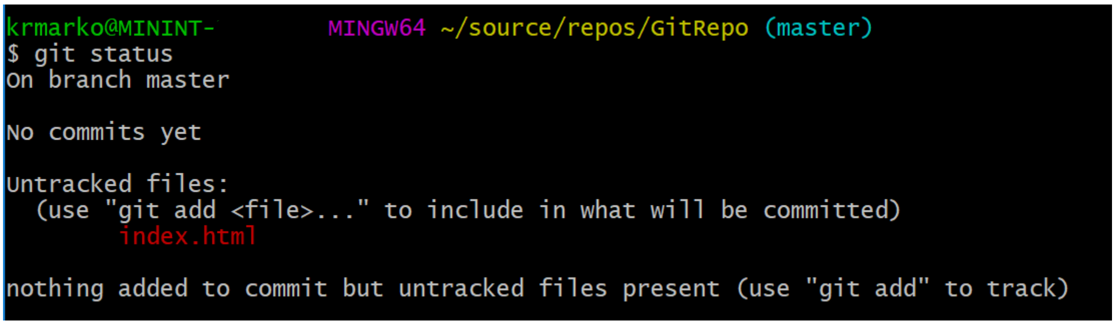
Populate your html file to contain some code (see below for sample) and save your changes.


index.html
c: > Users > krmarko > source > repos > GitRepo > index.html > html

```
<html>
    <body>
        <p>
            This is my index file.
        </p>
    </body>
</html>
```

Staging and Committing Changes

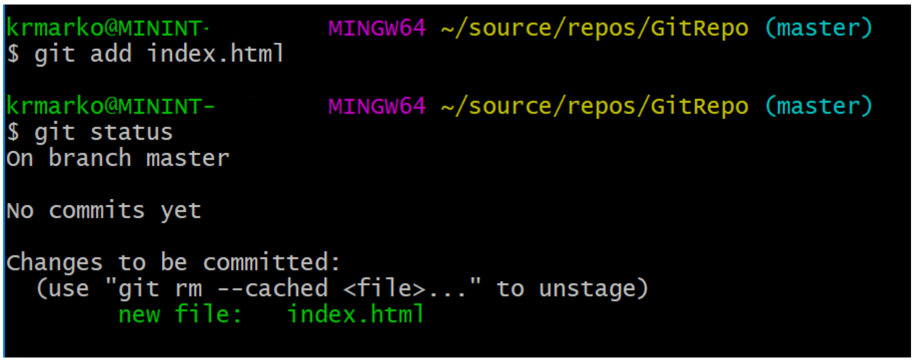
Once you have updated your file, go back to Git Bash. To view these changes, you can use the **git status** command.



```
krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
```

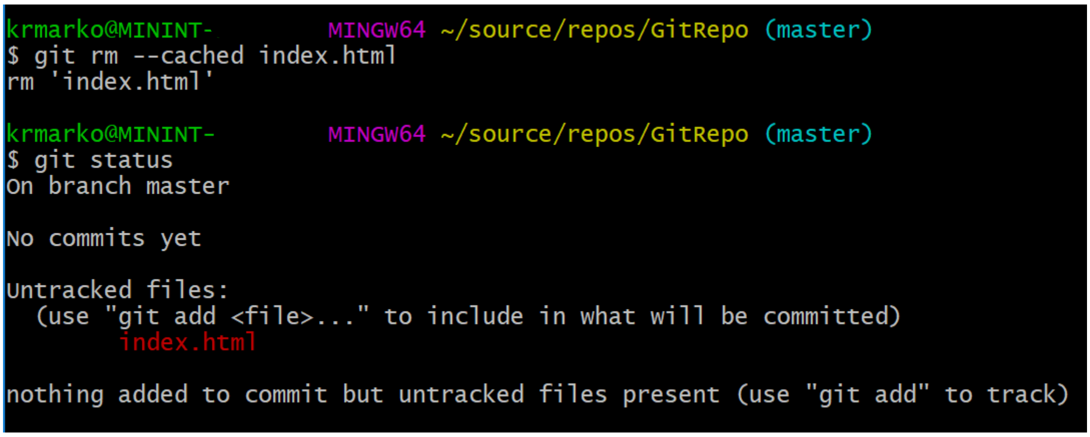
You can see that the file you have changed (index.html) is red and that the file is untracked. It is currently in your local repository and nowhere else.

In order to track these changes, you will need to move them to the staging environment and that is done by the **git add "file-name"** command. Once you have staged the index.html changes, you should see something like the below:



```
krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ git add index.html

krmarko@MININT-         MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
```

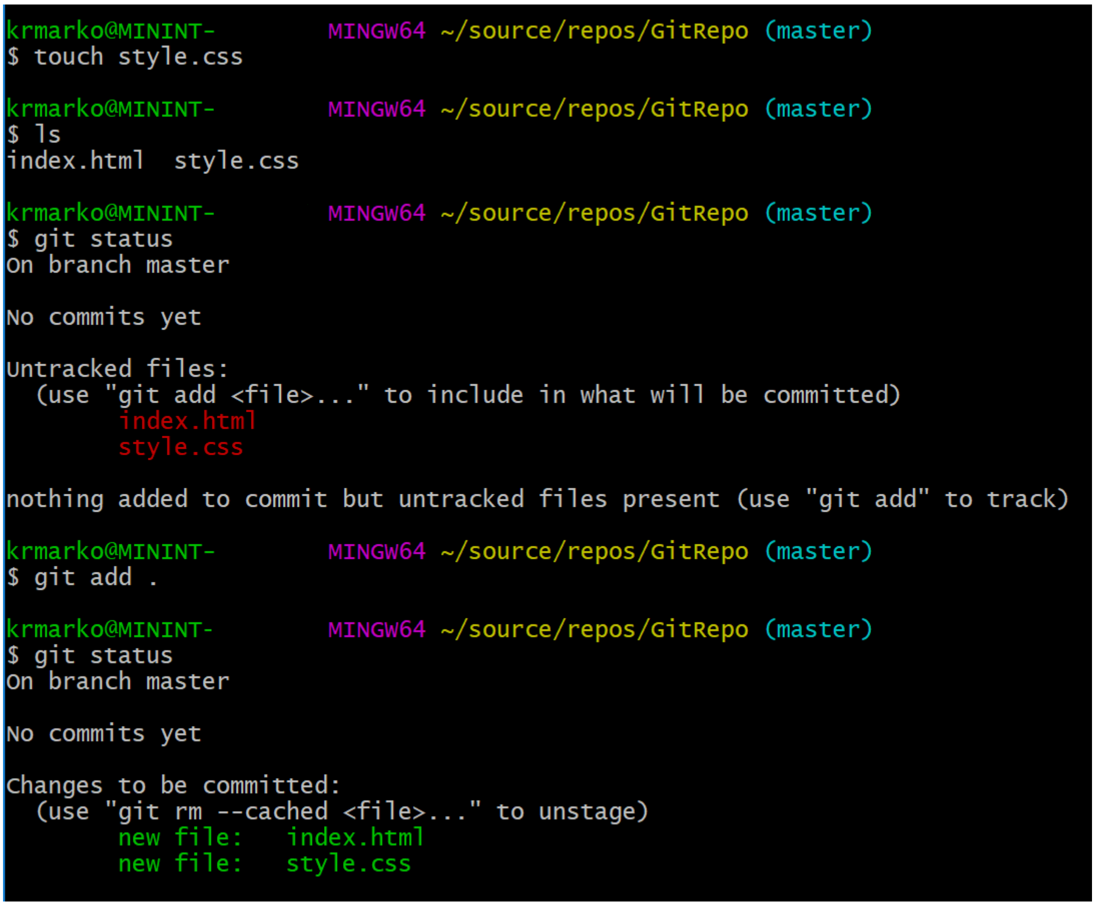
Before we commit the changes, we are going to add another file. So un-stage the changes (move them back to your local repository) using the **git rm –cached "file-name"** command. You can use the **git status** command to ensure that the file is untracked.



```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git rm --cached index.html
rm 'index.html'

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
```

Create a style.css file.

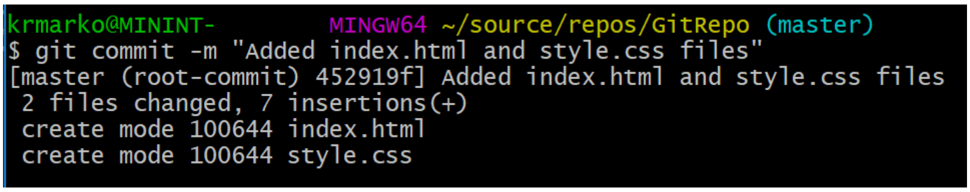
Stage all changes using the **git add .** command.



```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ touch style.css

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ ls
index.html  style.css

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        style.css

nothing added to commit but untracked files present (use "git add" to track)

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git add .

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   style.css
```

Now you need to commit your changes. Whenever you commit a change you need to have a message associated with it which will inform people of the changes made. In this case your message should include that you have added an index.html file and a style.css file.

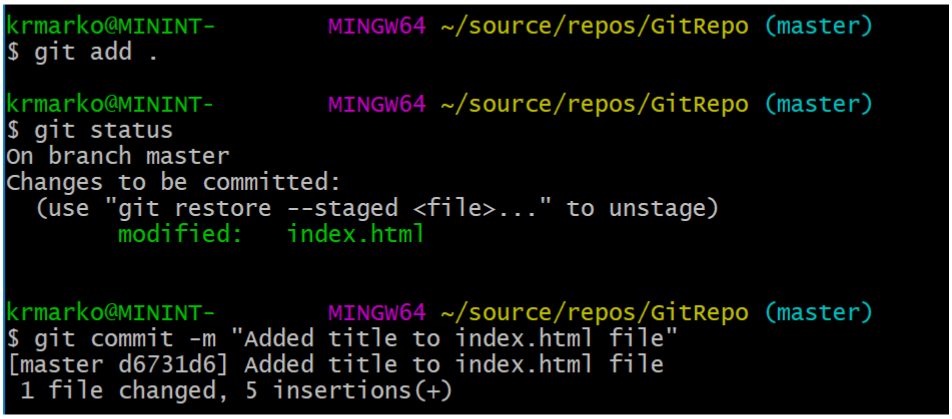
You are able to commit the change with the corresponding message using the **git commit -m "Message"** command.



```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git commit -m "Added index.html and style.css files"
[master (root-commit) 452919f] Added index.html and style.css files
 2 files changed, 7 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css
```

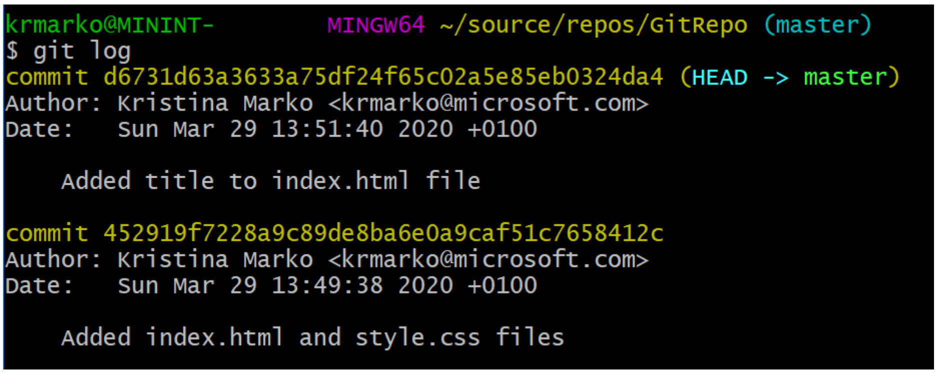
Congratulations! You have committed your first change. Now let's make another change.

Go back to Visual Studio Code and add a title to your index.html file and commit the changes. It should look something like this:



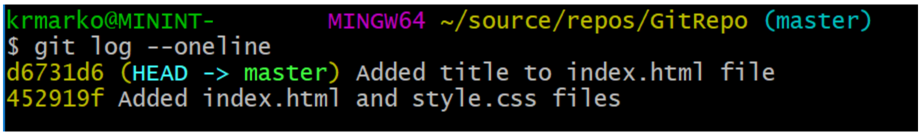
```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git add .

krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html


krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git commit -m "Added title to index.html file"
[master d6731d6] Added title to index.html file
 1 file changed, 5 insertions(+)
```

Now we have made 2 commits in total. If we wanted to see all commits made we would use the **git log** command.



```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git log
commit d6731d63a3633a75df24f65c02a5e85eb0324da4 (HEAD -> master)
Author: Kristina Marko <krmarko@microsoft.com>
Date:   Sun Mar 29 13:51:40 2020 +0100

    Added title to index.html file

commit 452919f7228a9c89de8ba6e0a9caf51c7658412c
Author: Kristina Marko <krmarko@microsoft.com>
Date:   Sun Mar 29 13:49:38 2020 +0100

    Added index.html and style.css files
```

To view the commits on one line (so just the commit number and the message associated with it, you would run command **git log --oneline** and it would look something like this:



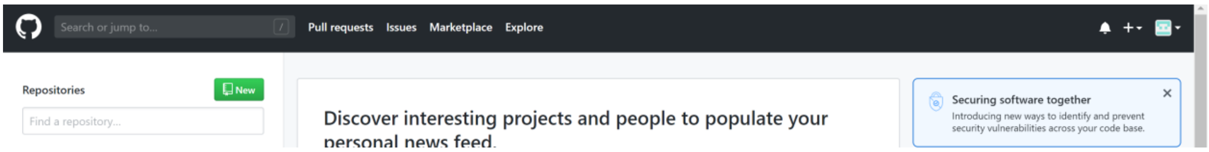
```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git log --oneline
d6731d6 (HEAD -> master) Added title to index.html file
452919f Added index.html and style.css files
```

Creating a Remote Repository

Up to now you have created a local repository and committed files to it. The next step is creating a remote repository that you can push your files to.

In order to do that we must first create a remote repository. Navigate to https://www.github.com

Create an account if you don’t have one already. Navigate to the home page which should look something like the image below. On the left hand side you should see your “repositories”. Click on “New”.



Search or jump to...

Pull requests Issues Marketplace Explore

Repositories

New

Find a repository...

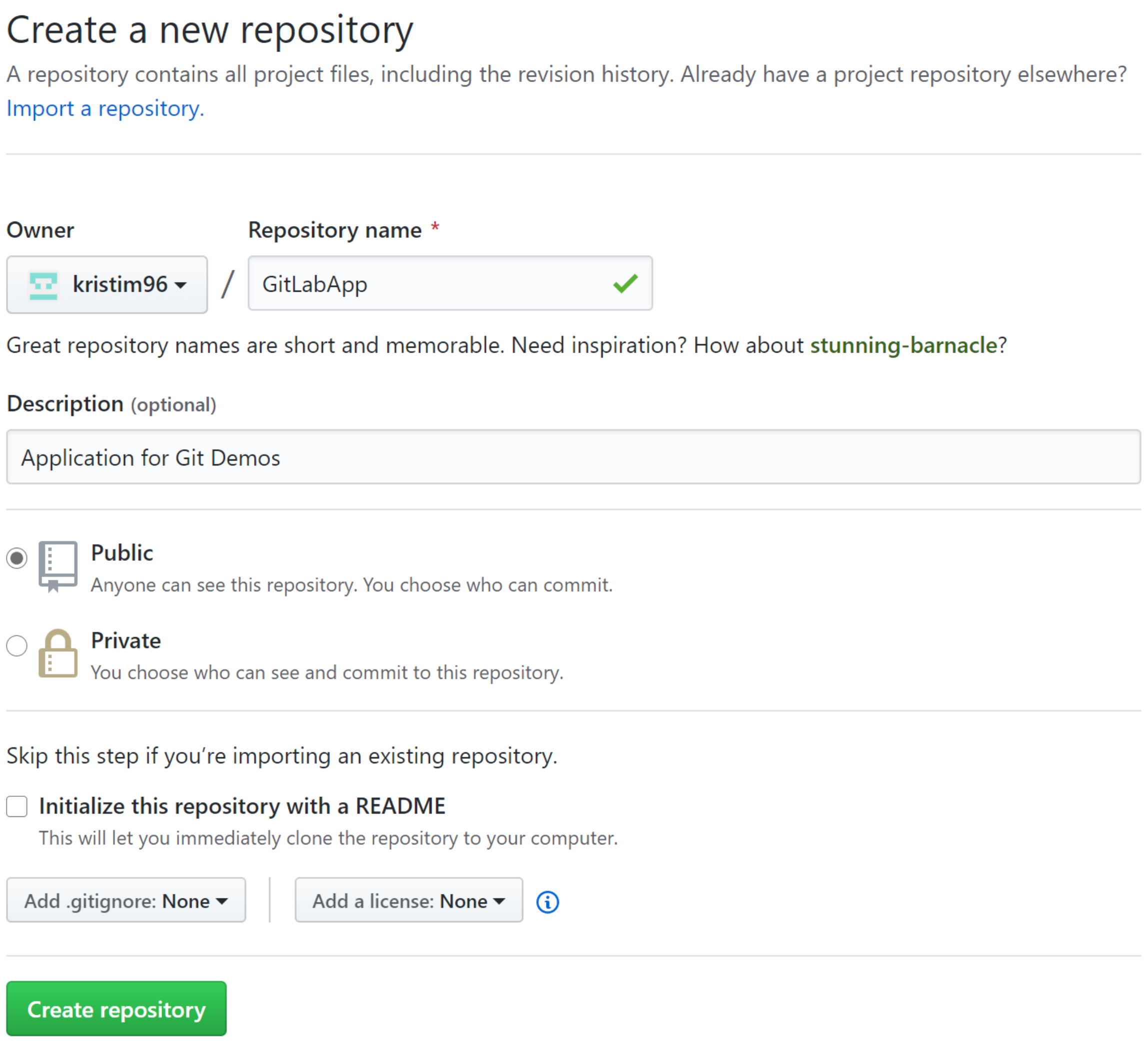
Discover interesting projects and people to populate your personal news feed.

Securing software together

Introducing new ways to identify and prevent security vulnerabilities across your code base.

This will enable you to create a new repository.

As we will be importing an existing repository, we want this one to be empty so provide a name for your repo along with a description. You don’t need to tick the checkbox. Once completed, hit the “create repository” button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: kristim96 / Repository name *: GitLabApp ✓

Great repository names are short and memorable. Need inspiration? How about stunning-barnacle?

Description (optional): Application for Git Demos

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Skip this step if you’re importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None

Add a license: None

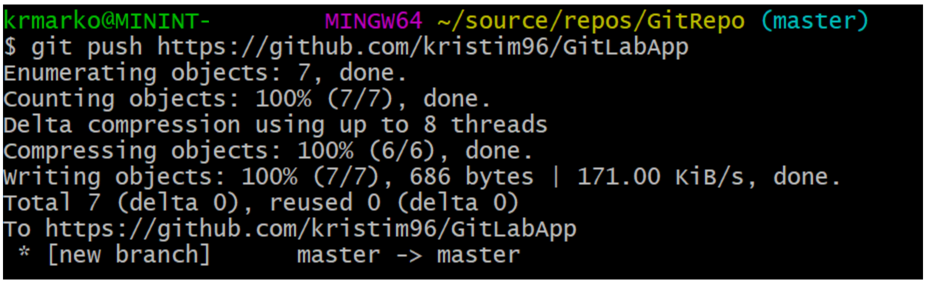
Create repository

You now have created an empty remote repository. Copy the URL of your repo

You now have 2 repos: One local repository where you have committed your changes and one remote repository that is empty.

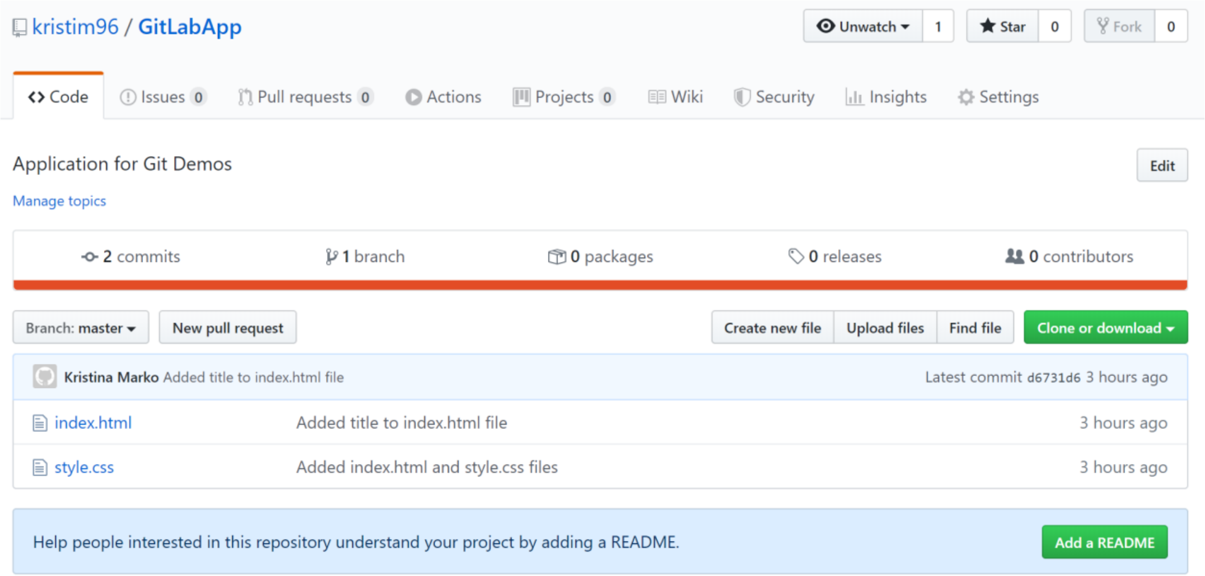
You now need to push your local repository to the remote repo. In order to do that we will need the URL of your remote repo.

Then, go back to Git Bash and use the **git push <URL REMOTE REPO>** command.



```
krmarko@MININT-        MINGW64 ~/source/repos/GitRepo (master)
$ git push https://github.com/kristim96/GitLabApp
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 686 bytes | 171.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/kristim96/GitLabApp
 * [new branch]      master -> master
```

If you check on GitHub now, you should see be able to see your local repository's changes/files.



Application for Git Demos

Kristina Marko Added title to index.html file   Latest commit d6731d6 3 hours ago

index.html	Added title to index.html file	3 hours ago
style.css	Added index.html and style.css files	3 hours ago

Help people interested in this repository understand your project by adding a README.