Command

Wednesday, September 23, 2020 3:06 PM

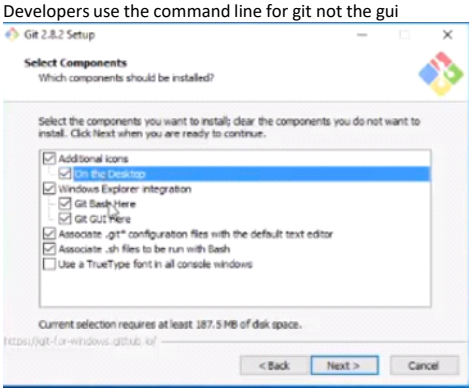
git	Everything begins with git
git config	Config the tooling
git config --global user.name "duan"	Configure the user name globally
git config --global user.email "duan@test.com"	Configure the user email globally
git config --edit --global	Open a file to edit global configs
git init	Initialize a git local repo
git clone	Download a project from remote
git add	Prepare a file (to staging)
git commit	Commit a file to the repo
git --version	Check the version of git you are running
git status	- Tracking progress - Current Branch - Which files were modified - Next steps
git add .	Recursively add all files to staging
git help	Allows you to get hellp for a specific command
git commit -m "test"	Commits files with a message
git checkout --test.py	Removes a change in working directory before adding to staging
git reset HEAD	Unstages file but retains the most recent changes.
git checkout -b test	Create a new branch and switch to it
git checkout -d test	Removes a branch
git log --all --decorate --oneline -graph	Ac1274e(HEAD -> New_Folder, origin/New_folder) compare_acl/
git diff	
git remote add origin git@github.com:duan/test.git	Meaning you can push up and pull down from origin
git push -u origin master	Push your code to repo
git log	Check log
Git clone	To clone a repo
Git rebase -i master	To rebase your branch with master

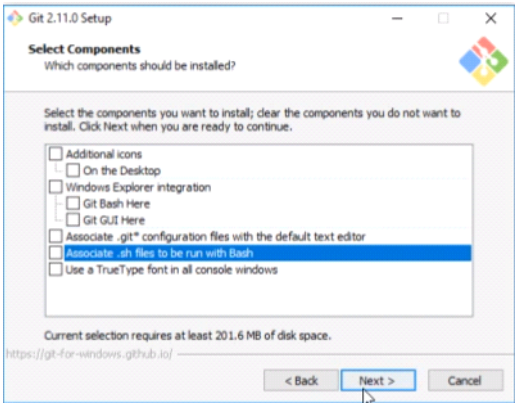
Git For Windows

Friday, December 21, 2018 8:57 PM

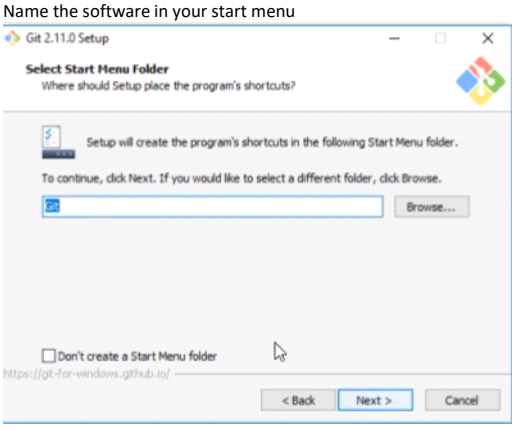
1. Go to chrome and type git for windows
   https://git-scm.com/download/win

2. Right click file and run as an administrator
3. Uncheck the options. You can choose to run in windows explorer but not recommended. Developers use the command line for git not the gui



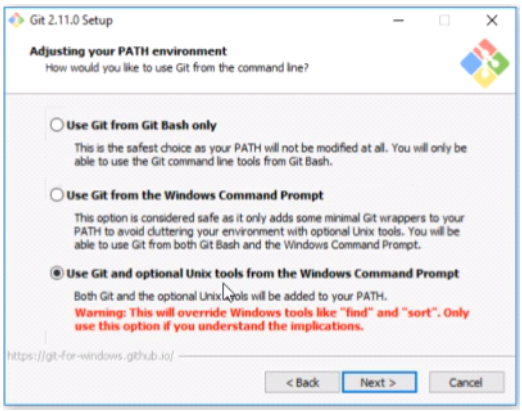


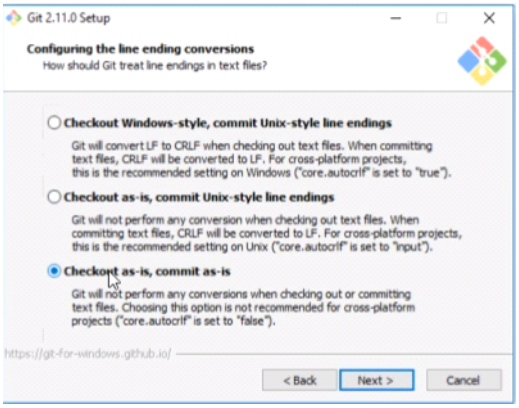
4. Name the software in your start menu

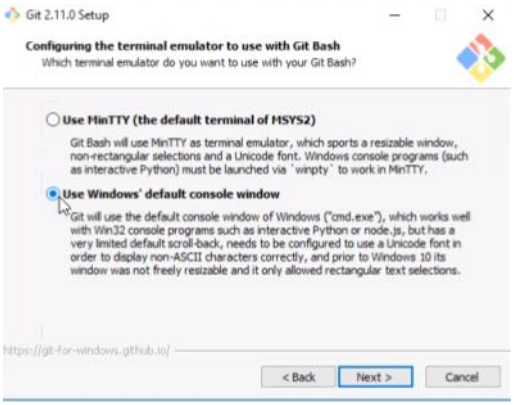


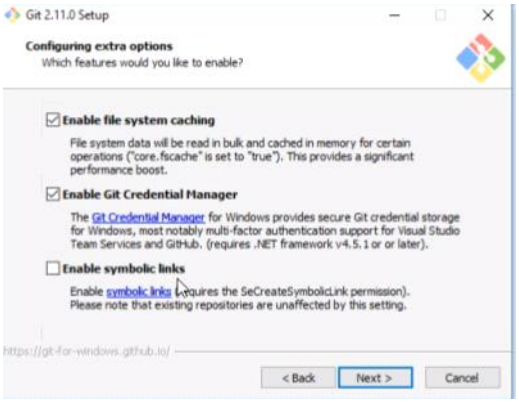
Reasons for using GIT

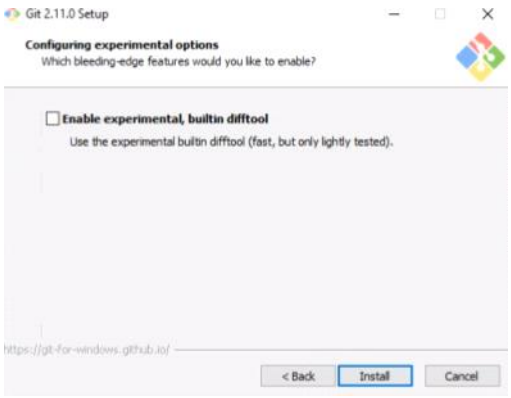
1. Share your code
2. Maintain your version - Writing code and making a change and then losing or confusing your scripts sucks


Git 2.11.0 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Use Git from the Windows Command Prompt
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.
Use Git and optional Unix tools from the Windows Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
https://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.11.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.11.0 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.11.0 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable Git Credential Manager
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
https://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.11.0 Setup
Configuring experimental options
Which bleeding-edge features would you like to enable?
Enable experimental, builtin difftool
Use the experimental builtin difftool (fast, but only lightly tested).
https://git-for-windows.github.io/
< Back
Install
Cancel

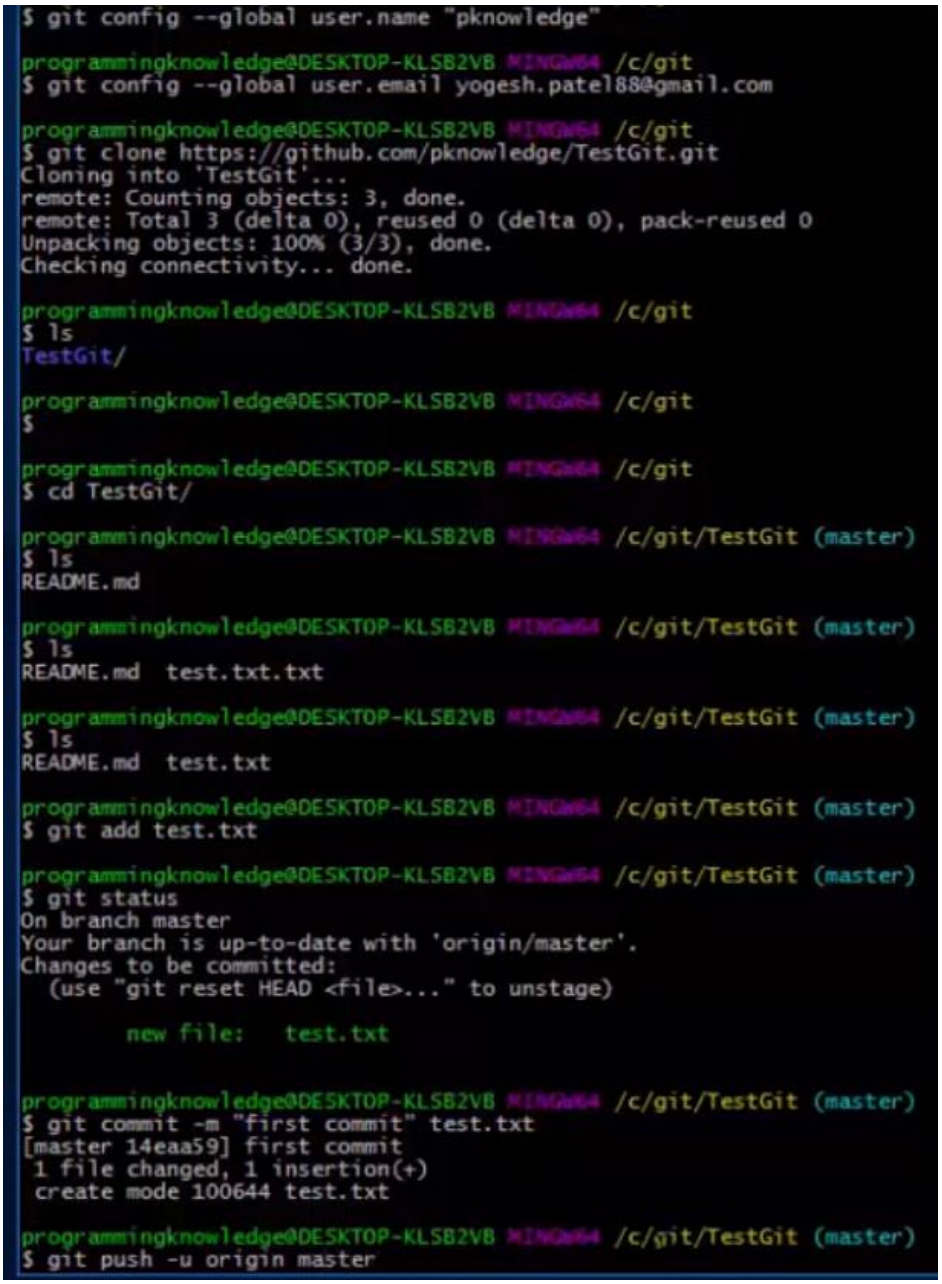
Git Bash

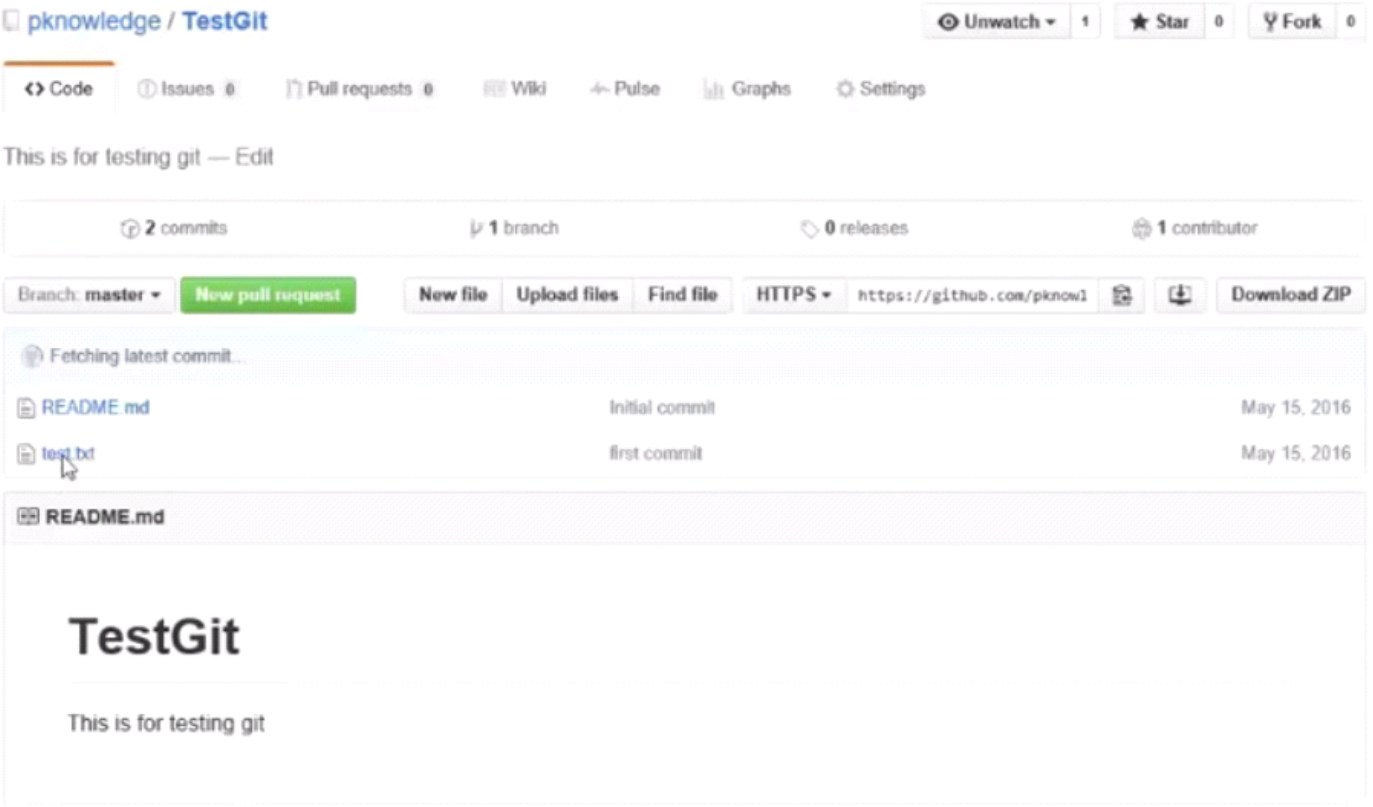
Friday, December 21, 2018 9:33 PM

Set Up Git: git config --global user.name "Your Name"
git config --global user.email "youremail@domain.com"

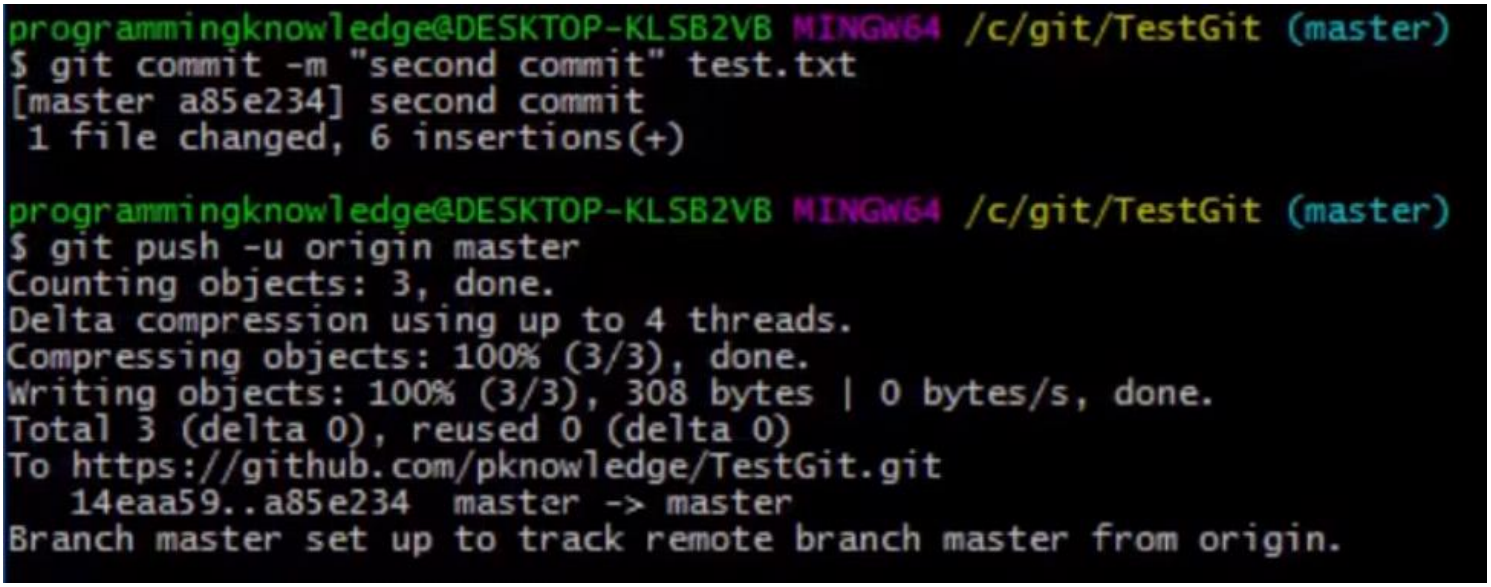
other commands
git clone (url)
git add (file names)
git commit -m "(committed message) (Commit saves it in a local repository)
git push -u origin master (uploads to github - You will have to use your password after the first time)
git remote add origin master (url)

From <https://www.youtube.com/watch?v=J_Clau1bYco>





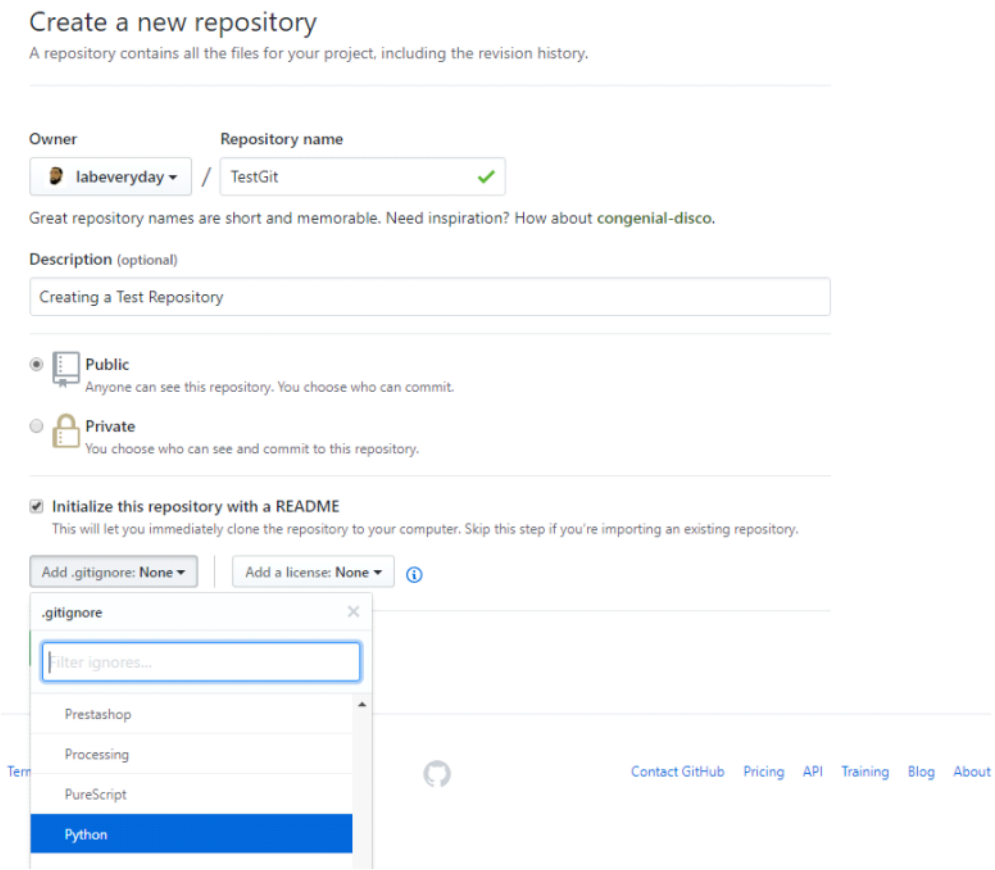
After making changes to your file you can do the commit and then push

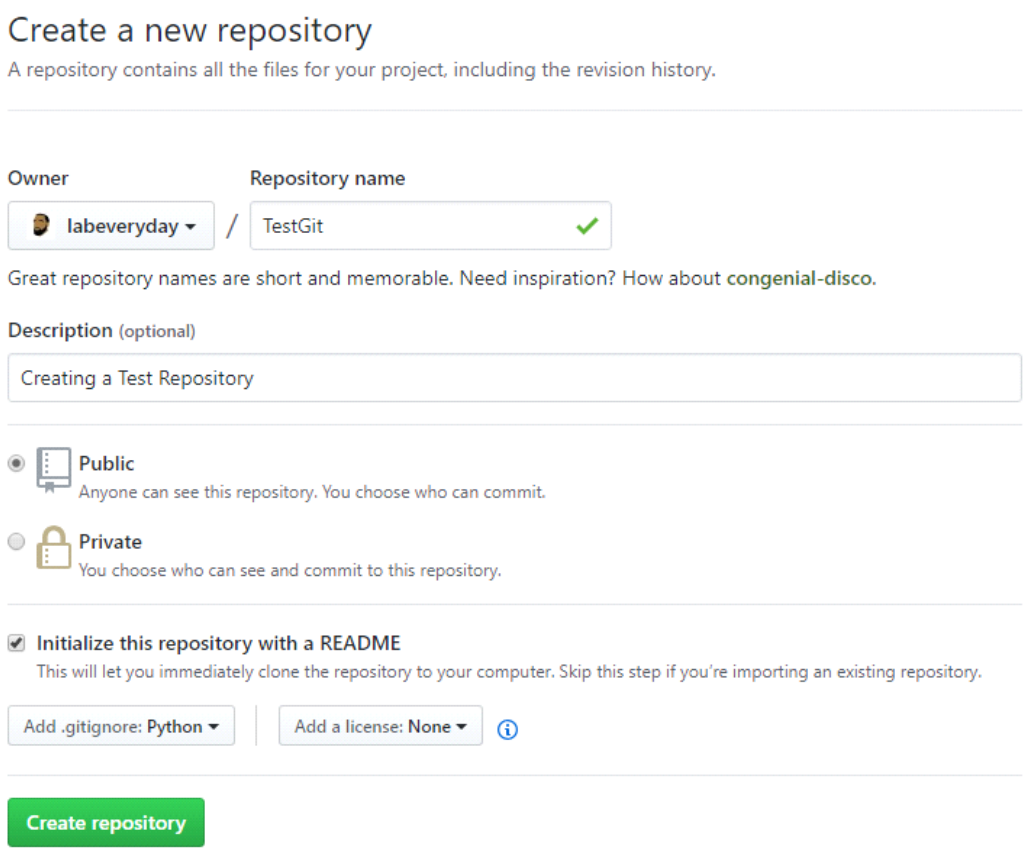

programmingknowledge@DESKTOP-KLSB2VB MINGW64 /c/git/TestGit (master)
$ git commit -m "second commit" test.txt
[master a85e234] second commit
1 file changed, 6 insertions(+)
programmingknowledge@DESKTOP-KLSB2VB MINGW64 /c/git/TestGit (master)
$ git push -u origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 308 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pknowledge/TestGit.git
14eaa59..a85e234 master -> master
Branch master set up to track remote branch master from origin.

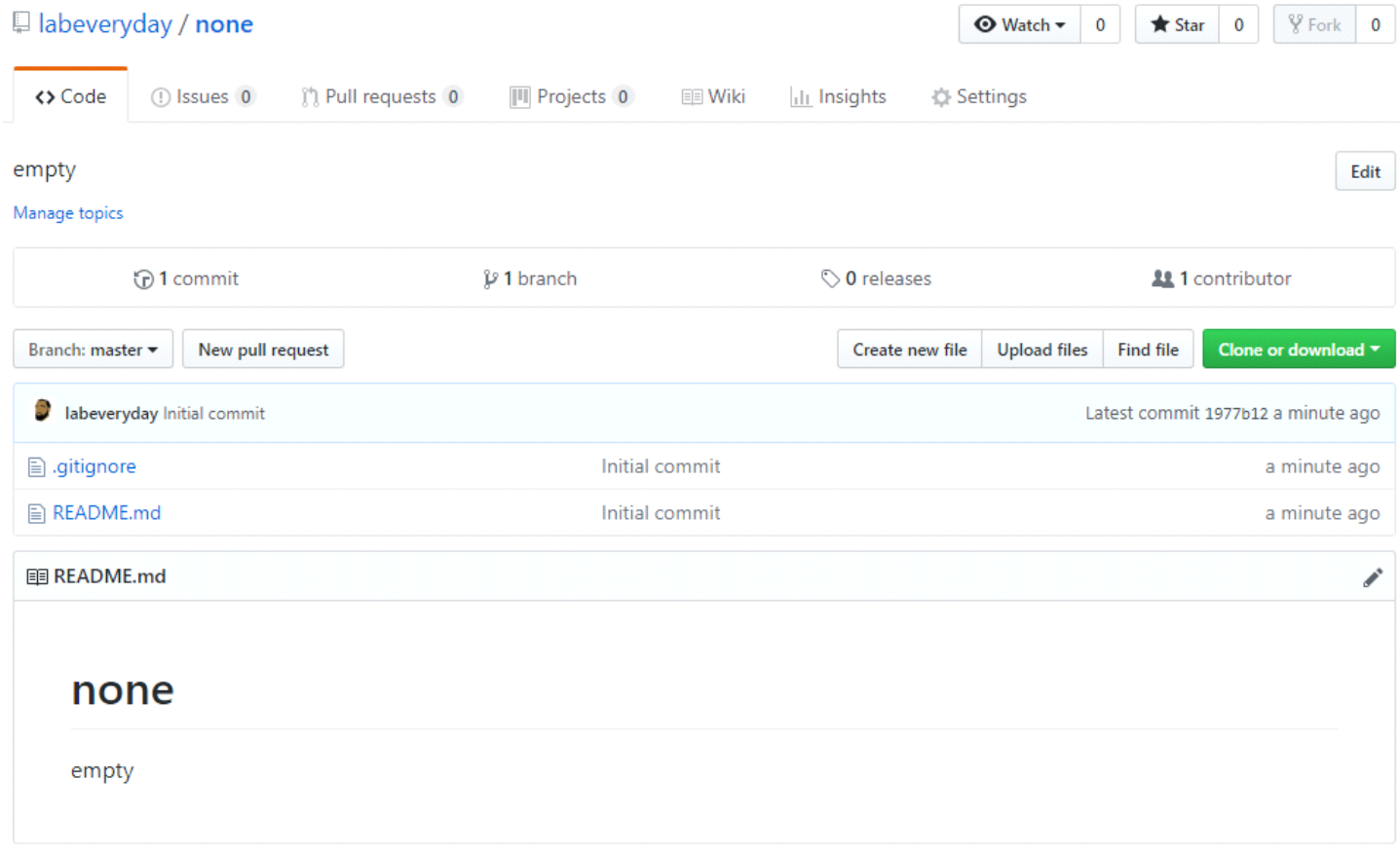
GitHub

Friday, December 21, 2018 9:28 PM

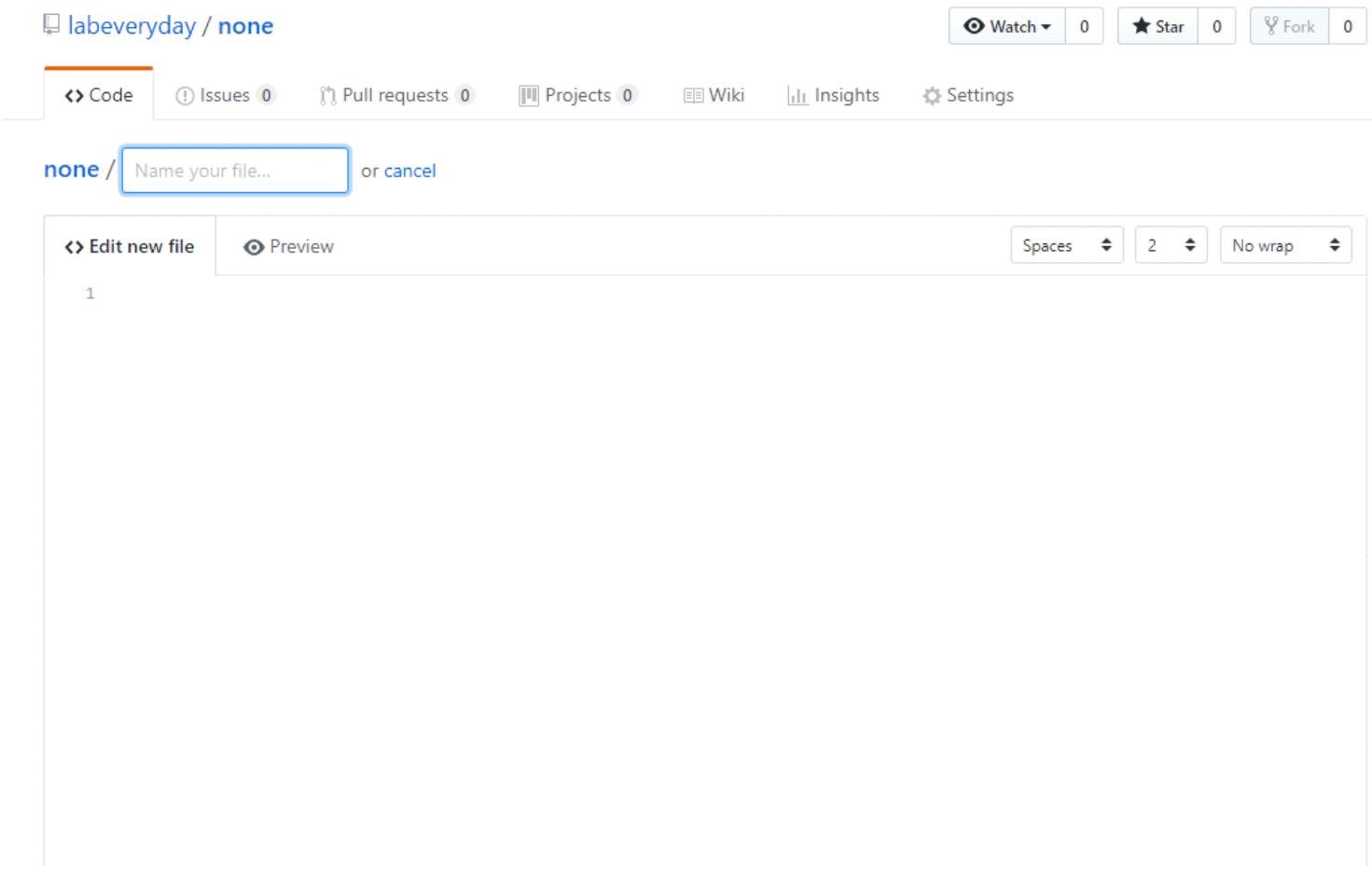
1. Create account
2. Two-factor security
3. Create a new repository (private comes with fees)



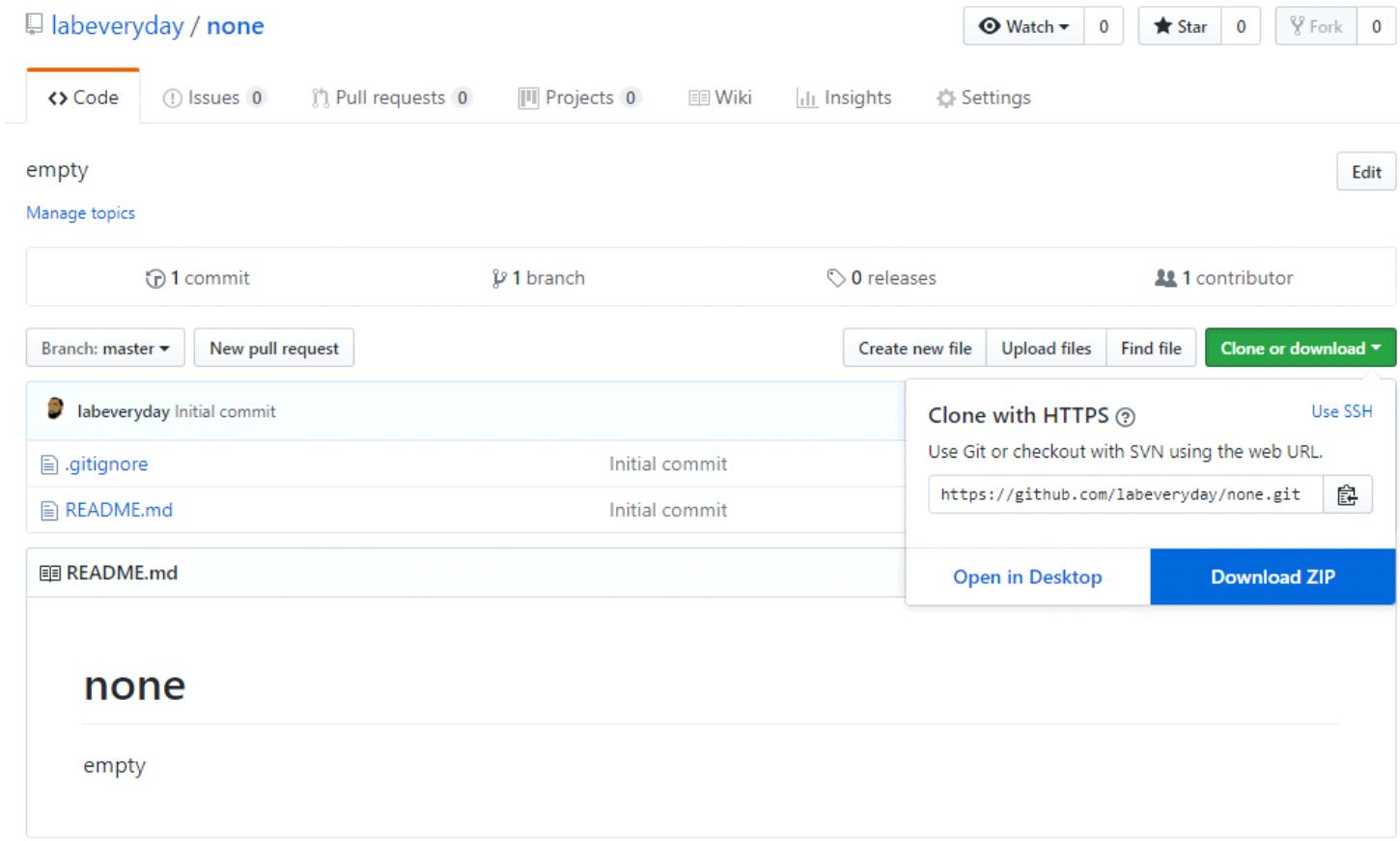




Create new file to copy code here. But we will be using pycharm to push code.



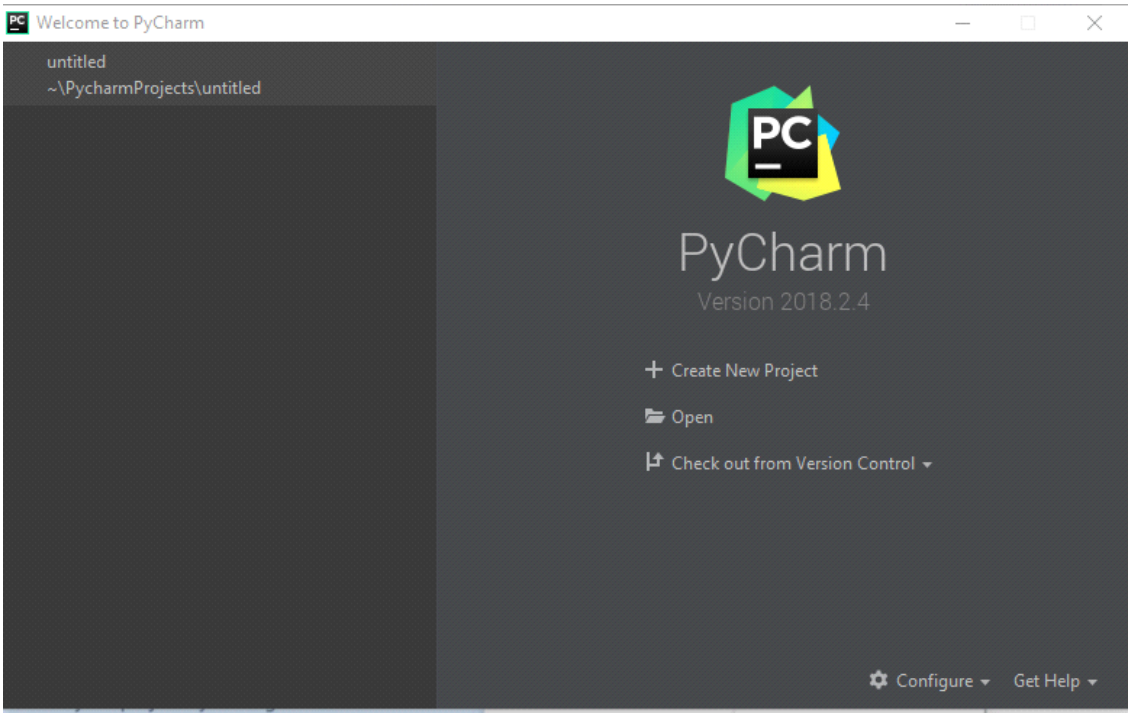
Download code from github


labeveryday / none
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
empty
Edit
Manage topics
1 commit
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
labeveryday Initial commit
.gitignore Initial commit
README.md Initial commit
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/labeveryday/none.git
Open in Desktop
Download ZIP
README.md
none
empty

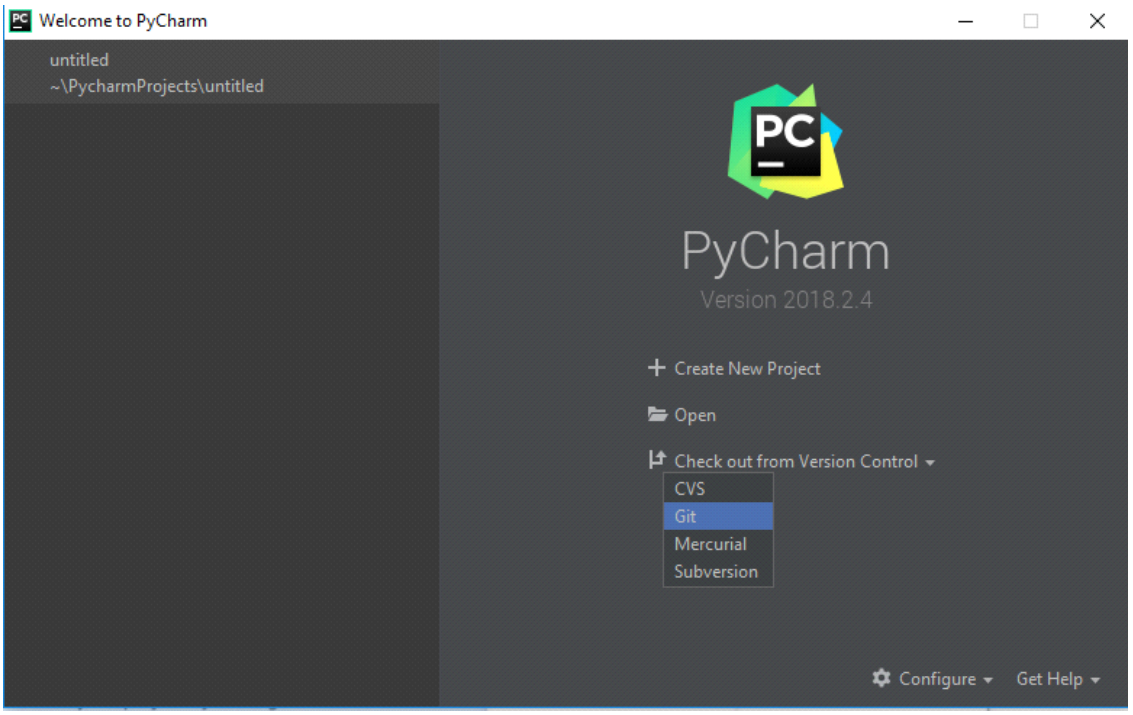
Pycharm Github

Friday, December 21, 2018 9:50 PM

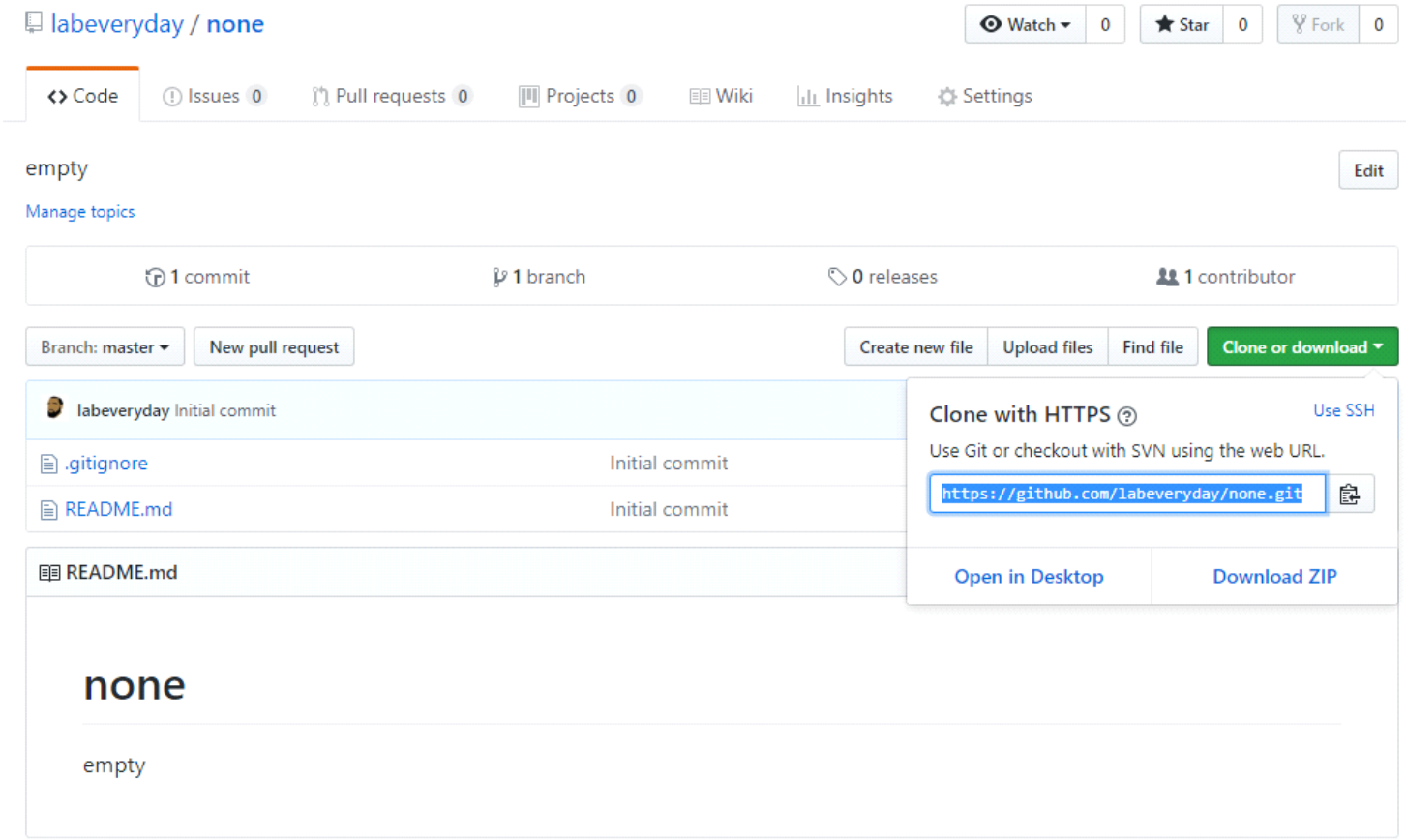
Using PyCharm and GitHub to push, share, and maintain code

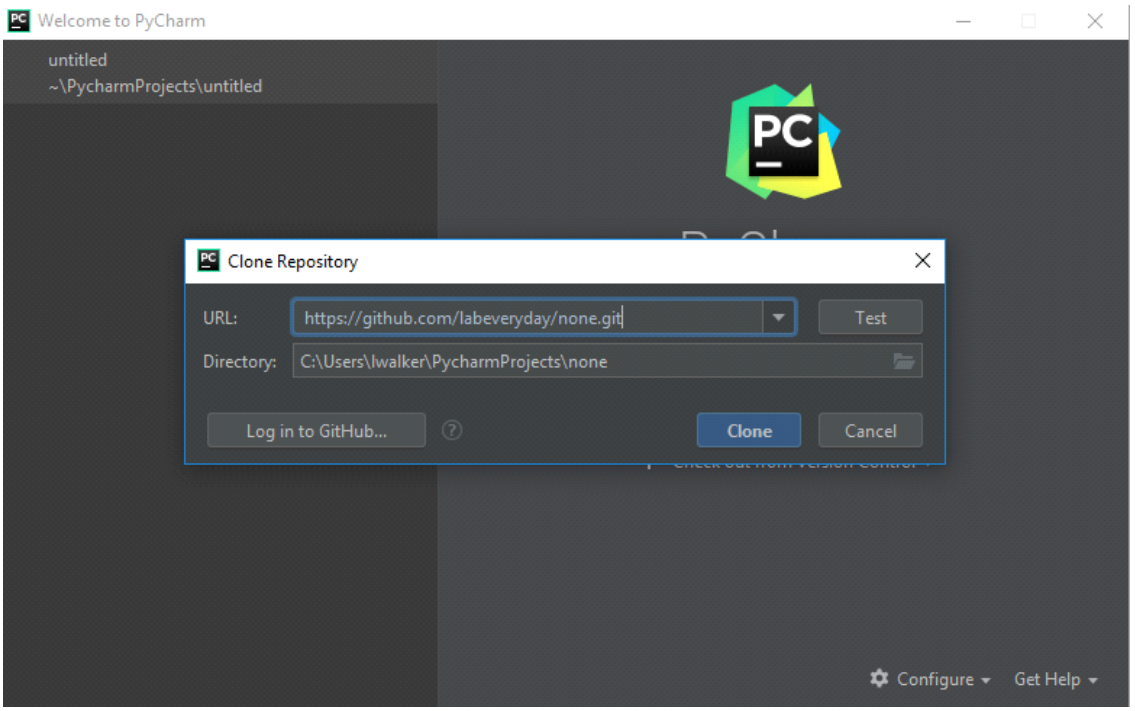


To pull project from Github into pycharm

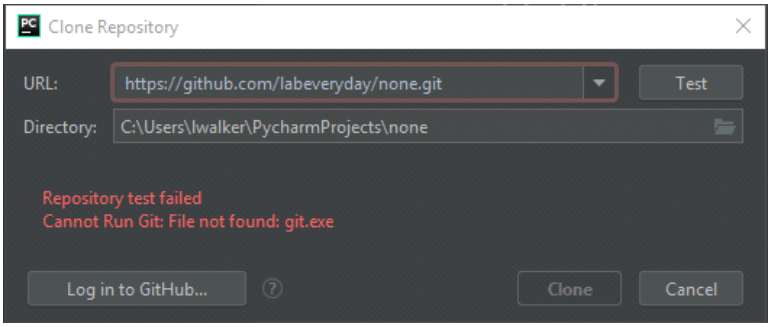


Copy URL to clone to pycharm





If you click Test and receive this error then you will have to Install GIT. Watch this tutorial on how to install and setup git.

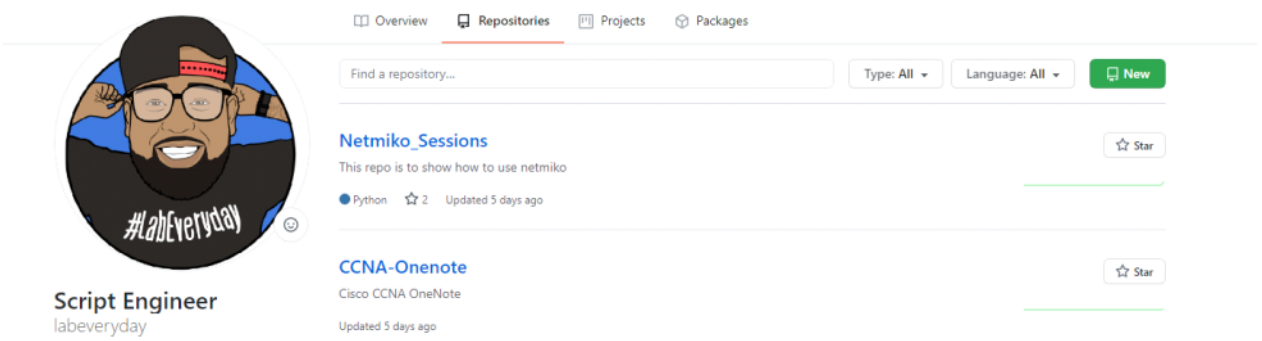


Once installed it will create a directory for you.

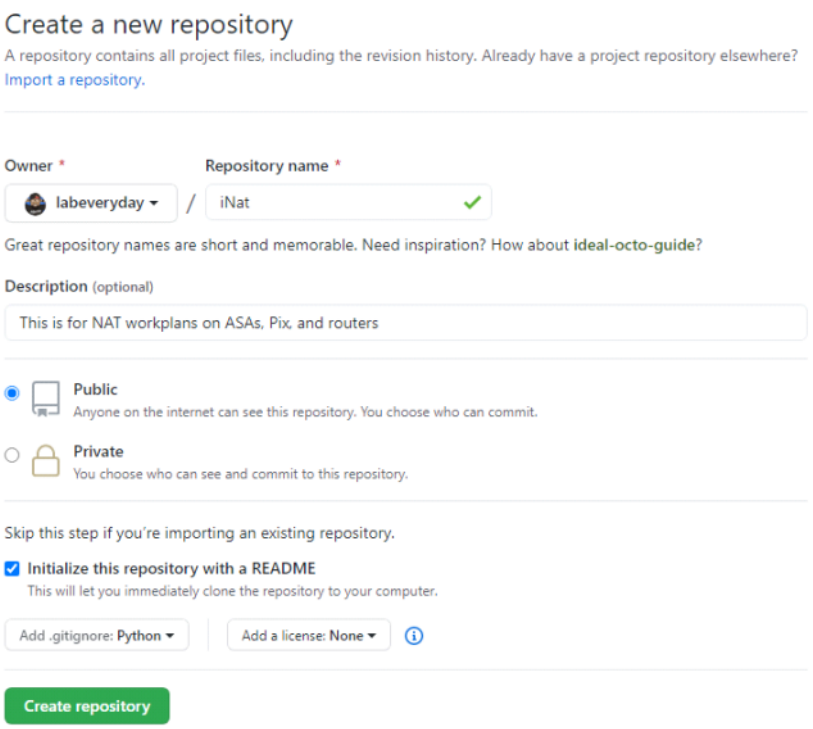
Steps to Creating a project

Tuesday, July 7, 2020 9:10 PM

1. **Create the repository on GitHub**



2. **Add the settings**



3. **Clone the remote repo to local**

```
PS C:\> cd .\github\
PS C:\github> git clone https://github.com/labeveryday/iNat.git
Cloning into 'iNat'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
```

4. **Cd to your project directory. Create and activate your virtual environment**

```
C:\>cd github\iNat

C:\github\iNat>python -m venv inat_venv

C:\github\iNat>ls
README.md  inat_venv

C:\github\iNat>inat_venv\Scripts\activate.bat
```

```
C:\>cd github\iNat

C:\github\iNat>python -m venv inat_venv

C:\github\iNat>ls
README.md  inat_venv

C:\github\iNat>inat_venv\Scripts\activate.bat
```

5. **Verify your python packages that are installed**

```
(inat_venv) C:\github\iNat>pip list
Package    Version
---------- -------
pip        18.1
setuptools 40.6.2
You are using pip version 18.1, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

6. **Install the needed packages**

```
(inat_venv) C:\github\iNat>pip install Jinja2
Collecting Jinja2
  Using cached https://files.pythonhosted.org/packages/30/9e/f663a2aa66a09d838042ae1a2c5659828bb9b41ea3a6efa20a20fd92b121/Jinja2-2.11.2-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2)
  Using cached https://files.pythonhosted.org/packages/5b/d4/1deb3c5dc3714fb160c7e2116fc6dff36a063d9156a9328cce54ef35cc52/MarkupSafe-1.1.1-cp37-cp37m-win32.whl
Installing collected packages: MarkupSafe, Jinja2
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1
You are using pip version 18.1, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

(inat_venv) C:\github\iNat>pip list
Package           Version
----------------- -------
astroid           2.4.2
colorama          0.4.3
isort             4.3.21
Jinja2            2.11.2
lazy-object-proxy 1.4.3
MarkupSafe        1.1.1
mccabe            0.6.1
pip               18.1
pylint            2.5.3
setuptools        40.6.2
six               1.15.0
toml              0.10.1
typed-ast         1.4.1
wrapt             1.12.1
You are using pip version 18.1, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

7. **You can also install your packages from a setup.py file or a requirements.txt**

```
(inat_venv) C:\github\iNat>pip freeze > requirements.txt
```

requirements.txt ×

requirements.txt

```
astroid==2.4.2
colorama==0.4.3
isort==4.3.21
Jinja2==2.11.2
lazy-object-proxy==1.4.3
MarkupSafe==1.1.1
mccabe==0.6.1
pycodestyle==2.6.0
pydocstyle==5.0.2
pyflakes==2.2.0
pylama==7.7.1
pylint==2.5.3
six==1.15.0
snowballstemmer==2.0.0
toml==0.10.1
typed-ast==1.4.1
wrapt==1.12.1

```

8. **To deactivate your virtual environment**

```
(inat_venv) C:\github\iNat>deactivate
C:\github\iNat>
```

9. **NOTE: Never put files inside your virtual environment directory**
10. **NOTE: Never commit your virtual environment to source control**
    a. **If you create your .gitignore file on github it will have your virtual environments listed.**


OPEN EDITORS
.gitignore
INAT
inat_venv
.gitignore
README.md
.gitignore
104 # Environments
105 .env
106 .venv
107 env/
108 venv/
109 ENV/
110 env.bak/
111 venv.bak/
112