Git & Github terminal guide

This workflow is designed particularly for solo developers, students looking to start using Github for backup and time based recovery purposes etc.

This workflow is for those wanting to use git via the terminal. If you just want to use Git & Github as a backup tool without bothering with the terminal, see my guide for using the Github Desktop app.

Setup: Github account

Scenario

- You need to create your Github account before being able to do anything else in this document.

Procedure

1. Visit https://github.com/
2. Click “Sign up”
3. Enter a username, email and password.
4. Verify you are human, click “Create account”
5. “Choose your subscription” = Free
6. “What is your level of programming experience” and “What do you plan to use Github for” you can click “skip this step” at the bottom.
7. Please verify your email address – open your email, click on the link provided in the email you will have received from Github.
8. When the “create a new repository” screen appears you are done. Don’t create a repository through the online site.

Git & Github terminal guide

Setup: Local computer

Scenario

- You have a Github account, but this computer has not been setup for Git or your Github account

```
# FOR MAC OR LINUX
# install git and openssh
sudo apt install git openssh
# Create a secure key pair
ssh-keygen -t rsa -C "your@email.com"
# Print your public key to screen
cat ~/.ssh/id_rsa.pub
```

WINDOWS

- Download and install git from https://git-scm.com/download/win
- Open a command prompt or powershell (press Windows and X keys together)

```
# FOR WINDOWS
# Create a secure key pair
ssh-keygen -t rsa -C "your@email.com"
# Print your public key to screen
type ~/.ssh/id_rsa.pub
```

MAC, LINUX AND WINDOWS

1. Copy the print out of this public key
2. Login to Github, goto tools, account settings, SSH keys, add SSH key.
3. Paste into the 'key' box and click 'Add key'.
4. You can now use the SSH clone URL when cloning to your local machine, bypassing the need to enter a username/password

```
# MAC, LINUX AND WINDOWS
git config --global user.name "Your name here"
git config --global user.email "your_email@example.com"
```

Git & Github terminal guide

New project: Create from local computer

Scenario

- Project does not exist on Github
- Project folder does exist on local computer but is not yet a git repository

```
# Change to your project folder
cd /project
# Create a git repository
git init
# Create empty repository on Github (alternatively use the web site)
curl -u 'USERNAME' https://api.github.com/user/repos -d '{"name":"PROJECTNAME"}'
# Link your local repository to the Github repository
git remote add origin git@github.com:USERNAME/PROJECTNAME.git
# Verify link
git remote -v
# Upload copy of local repository to the cloud repository
git push origin master
```

New project: Clone to local computer

Scenario

- Project exists as a repository on Github
- Project does not exist on local computer

```
# Move to the folder that will contain a folder for your project.
# Do not manually create your project folder
cd /projects
# Create a local repository based on the cloud copy
# Will create a new child folder
git clone https://github.com/USERNAME/REPOSITORY.git
# Example
git clone https://github.com/paulbaumgarten/example-project.git
```

Git & Github terminal guide

Upload from local computer to Github

Scenario

- You have made changes to your project on the local computer and wish to “back it up” to Github

```
# Change to the project folder
cd /project
# Add changes in local project to the local git repository and commit them
git commit -a -m "message describing change"
# Synchronise the local git repository with the github cloud repository
git push
```

The purpose of the message is to make it easier to find this particular change when searching through your git history later, in case you want to revisit or undo this change later. The presence of a message is mandatory.

Download to local computer from Github

Scenario

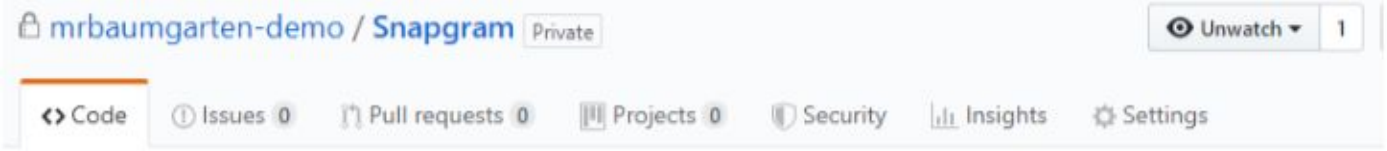
- Github has a more recent copy of your project than the local computer
  (perhaps you made some changes on a different computer which you uploaded to Github)

Always download any changes to your local machine before working on the local files. Otherwise you will get merge conflicts!

```
# Change to the project folder
cd /project
# Download changes from cloud repository to local repository
git pull
```

Share a project with another Github user

- Open your project on github website



- Click Settings
- Click Collaborators
- Enter the github username of your collaborator … my username is **paulbaumgarten**
- Click Add collaborator