Git Basics

Justas Mundeikis

2019-03-06

Contents

1 Basic settings

Many students working on the same computers cause, that the *Git config* has multiple diferent settings set. So there are basically two options: either unset set values, or edit the Git config file itself

```
# with this comand, user can unset the set values
$ git config --global --unset user.name
$ git config --global --unset user.email

# or open the nano editor (if set as standard editor for Git)
# and edit (delete) all entries, save and exit
$ git config --global --edit
```

You are setting Git on your own computer that noby else uses, then following settings should be set

```
$ git config --global user.name "Firstname Lastname"
$ git config --global user.email fistnamelastname@provider.geo
# preferebly set the email the same one, with which you created your GitHub account
```

In Windows 10, Windows tries to save your passwords. Which is from security point of view, not the best idea. Further the appearing pop-up to enter the password itself is annoying. Thus following settings should prevent Windows from saving your password and from opening a pop up for the password entry. Instead the password will be entered directly after entering your username after the command (for example `git push`)

```
$ git config --global core.pager cat
$ git config --global core.askPass ""
```

Depending on the computer operating system, after installing Git, the core editor might be *nano* or *Vim*. Personally I preffer *nano*. To make sure, you never end up in *Vim*, you can mannualy set your core editor to *nano*. Or if you have *Sublime* installed, set core editor su *Sublime*, by appending the execution file (.exe). I strongly suggest using *Sublime*, given its simplicity and many features

```
$ git config --global core.editor nano.exe

# here the absolute link to the execution file might be different on every computer!
$ git config --global core.editor /c/Users/USER/Download/sublime/sublime_text.exe
```

2 Git Basic Commands

Here is a short list of main commands and their breaf description

2.1 Initiating Git in a folder:

```
$ git init #initiates a repository
```

2.2 Adding failes to staging area

```
# adding files to staging area
# adds spesicif file
$ git add filename

# adds all files to staging area
$ git add .

# adds all files to staging area
$ git add -A

# updates the index (use only if no new files were created)
$ git add -u
```

2.3 Commiting files that are in staging area

```
# commiting files
# commits with message, without -m and "mesage text",
# editor would appear (nano or Submlime)
$ git commit -m "mesage text"

# here -a stands for all changed files, so use only if no new files
# were created, else stage with git add .
$ git commit -am "mesage text"
```

2.4 Branching and merging

```
# creating new branch (branchname = NewBranch |...)
$ git branch branchnname

# switching to the branch (branchname = master | NewBranch |...)
$ git checkout branchname

# merging happens into the branch, you are on,
# by calling the branchname to be merged with
$ git merge branchname
```

2.5 Reverting

```
# allways check git log
$ git log

# shorter git log version
$ git log --oneline
```

```
# reverting a commit by using its HASH
# (if necessary edit the merging errors by hand, save and commit -m "...")
$ git revert HASH

# reseting (reverts and DELETS everything from that commit)
$ git reset --hard HASH
```

2.6 Setting remotes

```
# setting remote directory (first create one in Github, WITHOUT Readme.md!)
# "origin" is just a name for the remote directory,
# you can have many remotes (with different http://... paths)
$ git add remote origin https://.....git

# to check if reomte is set
$ git remote -v

# removes the remote, for example the NAME=origin
$ git remote rm destination NAME

# pushes local repo to remote repo
$ git push -u origin master

# if you umat to update local repo from remote remote
$ git pull origin master

# if you the files in remote are outdated (or include Readme.md),
# simple pushing will not overwrite, thus
$ git push -f origin master

# cloning (downloading repo from GitHub) your own or some elses repo
$ git clone https://...git
```