

 **ai2-education-fiep-turma-4 / git-katas** Publicforked from [eficode-academy/git-katas](#)**Code**


Pull requests

Actions

Wiki

Security

Insights

 master ▾

...

git-katas / configure-git /

This branch is 1 commit ahead of eficode-academy:master.

 Contribute ▾**pal-pq and RandomSort** ...

on 6 Nov 2020



..



README.md

12 months ago

Git Kata: Configuring Git

 **README.md**

This kata has no `setup.sh` script. Just read and follow along.

Downloading and Installing on Windows

- Download at <https://git-scm.com/download/win> (or use [Chocolatey](#))
- Install using the pre-selected defaults
- After installation, open Git Bash for following configuration steps

Git Initial Configuration

Git wants to know who it should write as the committer of changes etc. In order to do that, configure user name and user email towards Git with the following commands:

1. `git config --global user.name "John Doe"`
2. `git config --global user.email "johndoe@example.com"`

Editor setup

Sometimes Git needs you to edit a file it creates e.g. the message of a commit you create. As default, Git is configured with VIM, but that has a steep learning curve, so you might be better off with another tool of your liking:

If you want to use the cli based editor nano:

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

For the Windows peeps:

- `git config --global core.editor notepad`

Or alternatively other tools you already are familiar with:

- `git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"`
- `git config --global core.editor "atom --wait"`
- `git config --global core.editor "code --wait"`

Aliases

You can set up aliases as such:

- `git config --global alias.lol 'log --oneline --graph --all'`

This might be useful to you when you look at the Git graph. Paste that into your terminal, and try it out with `git lol`.

More on aliases can be found in the alias kata.

SSH authentication

- See <https://help.github.com/articles/generating-an-ssh-key> for details about authenticating against SSH-enabled repositories
- Or run `ssh-keygen` to generate a SSH key pair in `%USERPROFILE%/.ssh/`:

```
ssh-keygen -t rsa -b 4096 -C "johndoe@example.com"
```

This generates public/private keys named `id_rsa.pub` / `id_rsa`, respectively)

- The public key `id_rsa.pub` needs to be uploaded to your repo server:
 - For GitHub, this is in *Settings* -> *SSH and GPG keys*
 - For BitBucket server, this is in *Manage Account* -> *SSH Keys*