

**SOFTWARE ENGINEERING TOOLS**

# **MANUALS FOR EMBEDDED ENGINEERS**

## **GIT - AN INTRODUCTION**

**SCHOOL OF ENGINEERING AND AUTOMOTIVE**

**HAN**\_UNIVERSITY  
**OF APPLIED SCIENCES**

V1.1 Feb 2021

© HAN University of Applied Sciences

Academy Engineering and Automotive / Embedded Systems

[joan.korten@han.nl](mailto:joan.korten@han.nl)

Ruitenberglaan 26

6826 CC Arnhem

The Netherlands

Web address: [han.nl](http://han.nl)

# WHY USE GIT

Writing software isn't easy you need to think hard about solving your problems and correctly writing it down in a language of choice. And if you have solved it, also software source code is naturally prone to errors. Syntactic errors, errors in grammar and most annoyingly semantic errors. Of course you can and will need to fix these errors. Also the software building process is often done in incremental interactive steps where functionality is added time and time again.

These are a number of incentives that make good version management a necessity for everyone that works with software source code. We just name a few:

- Tracking code changes
- Allow others to comment on your code
- Allow to merge code developed as a team
- Always having a backup of your source
- (Advanced) performing automated testing on your code
- (Advanced) moving from traditional development to Continuous Integration (CI)

# 1. GIT BASICS

## repository

/rɪˈpɒzɪt(ə)ri/

noun

- a central location in which data is stored and managed.

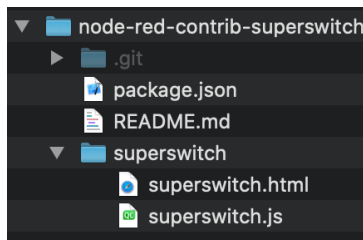
(Oxford dictionary)

First we need to tell you the basic structure of a (local) git repository. The following shows what it looks like:

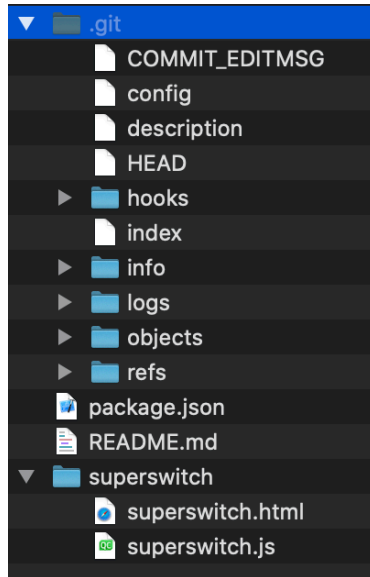
### git basic structure

- A. **working tree**  
e.g. c:\documents\HAN\Sx\cpp\projects\projectGit\  
or \$HOME/HAN/Sx/cpp/projects/projectGit/
- B. **staging area** (file with changes in your Git folder)
- C. **.git folder** (repository)

This is what a typical (local) repository could look like (macOS)

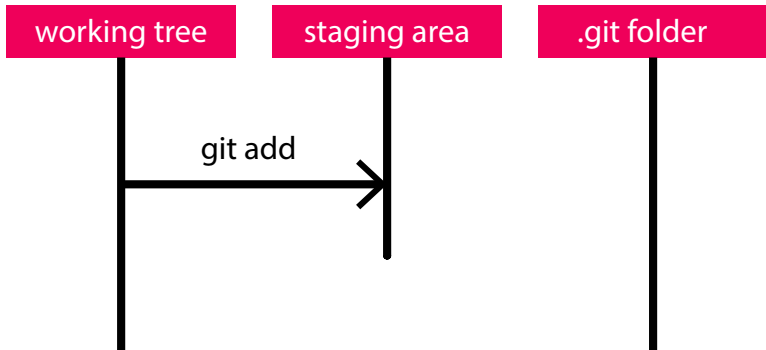


Look closely and notice a `git` hidden folder with all the local git files in it



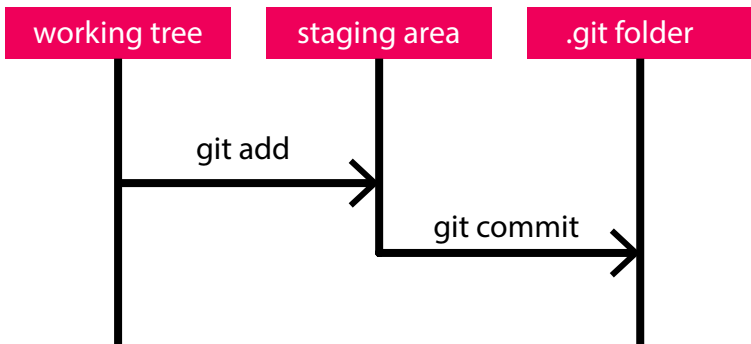
If you add a file or multiple files using `git add` followed by the filename(s) (or `.` for all files) they become *staged*.

## **git add**



Staging means that the current saved file is recorded in the stage area as is

## **git add and commit**



After staging files the next step is committing them This is done if you issue `git commit`

The staged files are then committed to the (local) repository After that you can add them to the remote repository we will come to that later

## 2. NEW LOCAL GIT REPO

### remote

/rɪ'məʊt/

adjective

- that you can connect to from far away, using an electronic link.

(Oxford dictionary)

Ok let's create a remote repository

Pick the service of your choice (e.g. GitHub, GitLab, Bitbucket)

Create an account

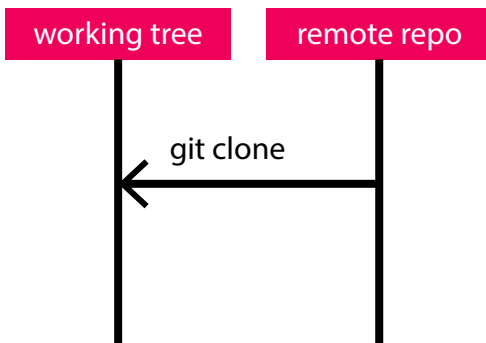
Create a new repository

Local repository

On your local computer move to a convenient folder

Issue a `git clone https://github.com/username/reponame.git`

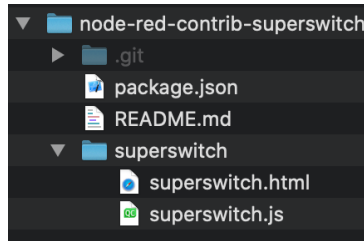
### **git clone**



So `git clone` will get all the files from the remote and clone it to your local working folder

*Note: We find this workflow to be the safest way to perform setup for git local and remote repositories.*

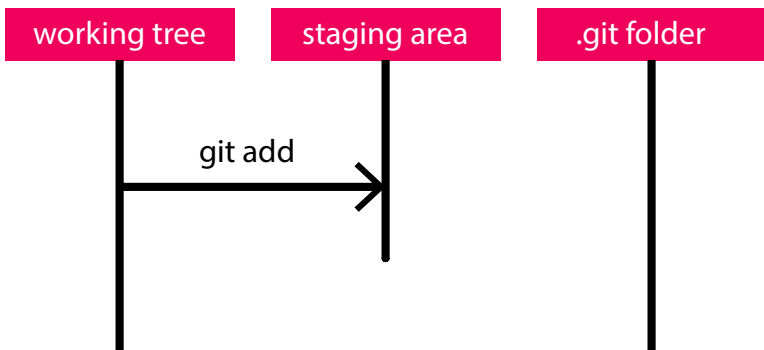
If you now look in your working folder you should see a folder with the name of the remote repository that looks somewhat like this (but if you didn't initialize it with a README.md on creation it is totally empty apart from the git files)



Now put your source files inside that folder re open them in your source editor

Remember to git add all source files to your local stage

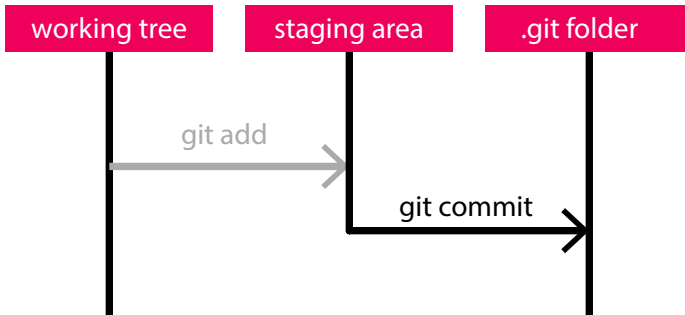
## **git add**





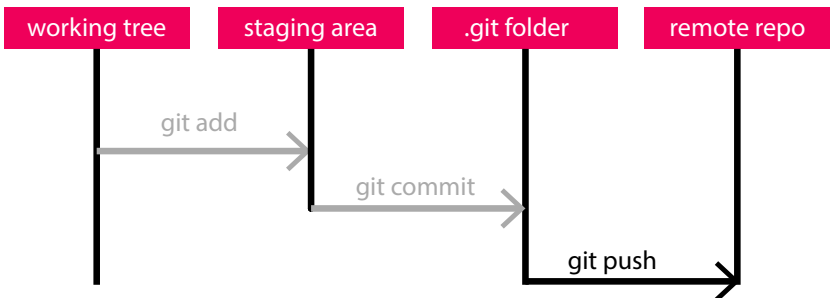
And then after larger changes commit to the repository with a useful short but descriptive message on what was changed

## 🎵 **git add and commit**



Finally don't forget to also push the local commits regularly to the remote

## 🎵 **git add, commit, push...**



## 3. WORKING TOGETHER

# Collaborate

/kə'labəreɪt/

verb

- work jointly on an activity or project.

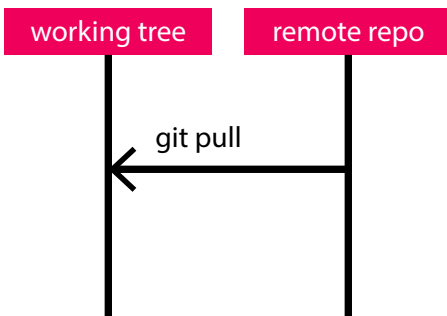
(Oxford dictionary)

You should check in the remote repository supplier of your choice (e.g. GitHub, GitLab or Bitbucket) to see how you can create teams and private repositories. Private repositories can not be seen by third parties but can be seen by members that are logged in to the environment.

If you start working on a repository for the day, remember to make sure your local and remote repositories are synchronized.

For this you use `git pull`

 **git pull**



Sometimes when you try to commit you get a message that shows your local repository is not in synch anymore with the remote. In that case do a `git pull merge`

After that you should be able to add, commit and push as usual.

You will however notice that you get merge conflicts time and again.



## BRANCHES / FORKING

A good practice is to work in your own *branch*. You do this by forking the master (remote) repository. You then do all your `add`, `commit`, `pushes` to that fork and only after a while you merge the changes on the forked repository to the master.

If you are not the owner of the master branch you can simply make a `pull request` and ask the owner to merge the changes you propose with the master branch.

## FINAL REMARKS

This tutorial is by no means exhaustive and there are numerous online resources available on git. Below just a few recommendations.

We also would like to encourage readers wanting to become professional (software) developers to

Integrate your development environment (IDE) with git.  
Getting knowledge of and experience with testing,  
continuous integration (CI), continuous deployment (CD)  
and or the DevOps process.



# USEFUL LINKS

---

Pro Git book by Scott Chacon and Ben Straub  
<https://git-scm.com/book/en/v2>

Git Crash Course: A Simple Workflow for Small Teams and Startups [shorturl.at/atKUZ](https://shorturl.at/atKUZ)

See also  
<https://github.com/Ineuhaus/pyrpl/wiki/Code-How-to-work-together-using-git>

## **Hosted Solutions:**

GitHub <https://help.github.com/en/github>

GitLab <https://about.gitlab.com/get-started>

Bitbucket <https://bitbucket.org/product/guides>

