

The access to github is through the github website and through a commandline interface once these are configured you are ready to go. Without revealing my detailed insight I can tell that GIT is a distributed version control system that allows multiple repositories to be synchronized. Most version control systems have a central repository and merging repositories is an elaborate task. GIT as such has some of the same conflict resolution issues, but handling these issues is much more common place as GIT is an inherent distributed version control system.

Being a novice git'er is not yet business a routine, to reconcile repositories, but at this stage my only experience is to browse / download repositories.

URL	https://github.com
user email	Some-email@address.dk
PASSWORD	Some-thing
User name	Some name (andersfuf)

From the website a new repository can be created. Github recommends to create a repository from the command line:

Quick setup — if you've done this kind of thing before https://github.com/andersfuf/uis-database.git
...or create a new repository on the command line echo "# uis-database" >> README.md git init git add README.md git commit -m "first commit" git remote add origin https://github.com/andersfuf/uis-database.git git push -u origin master
...or push an existing repository from the command line git remote add origin https://github.com/andersfuf/uis-database.git git push -u origin master
...or import code from another repository You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

My experience on this machine:

```
460 git status
503 git
504 history | grep git
505 git clone https://github.com/andersfuf/uis-database.git
506 sudo git clone https://github.com/andersfuf/uis-
database.git
510 history | grep git
```

...or create a new repository on the command line

```
echo "# uis-database" >> README.md
```

This first line creates a one line readme file. It did not work as used sudo to git clone. I had to change ownership on the folder from sudo user root to user develop: `sudo chown develop uis-database`

```
git init
```

Initialization is not necessary as it is created on the site

```
git add README.md
```

The readme file is posted. On my machine prefix with sudo to invoke as superuser: `sudo git add README.md`

```
git commit -m "first commit"
```

After registering my name and email with git config, git commit invoked with sudo post my first file.

```
git remote add origin https://github.com/andersfuf/uis-database.git
```

This is probably the command that adds a repository at your github user from the client command line. This is not needed this time.

```
git push -u origin master
```

Pushing files to the github must be invoked with sudo: `sudo git push -u origin master`

```
deBoor:uis-database develop$ sudo git commit -m "first commit"
Password:
```

```
*** Please tell me who you are.
```

```
Run
```

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'root@deBoor.
(none)')
deBoor:uis-database develop$
```

Notice above that the first commit prompts for username and email configuration on the client. In my case registered email but I could also have registered a ku-mail. The important point here is have rights to update the target repository (..target repository). Probably prefix with **sudo**.

```
git config --global user.email "yourAccount@di.ku.dk"
git config --global user.name "Your Name"
```

Setting my account preferences on the client did not require sudo. Now commit will work. The master trunc – repository on my github account is updated.

Changing the README.md file on my github account leaves my local copy in an older version. New versions can be pulled, evoked with sudo: sudo git pull

Alphabetical Index

git clone.....	2
git status.....	2
README.md.....	3
sudo chown.....	2
sudo git pull.....	3
sudo: sudo git push -u origin master.....	2
--global user.email.....	
git config.....	3
git push -u origin master.....	2