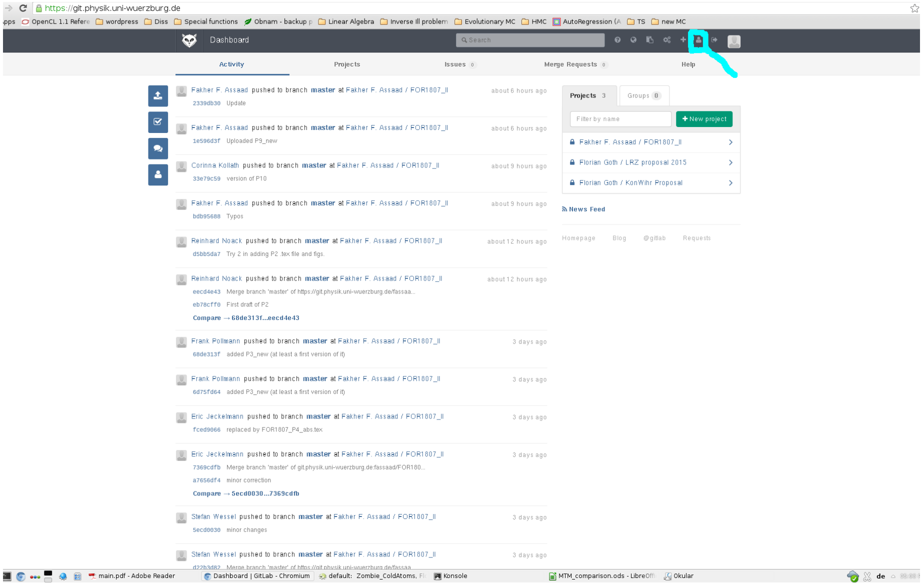
git Intro

Florian Goth

March 16, 2015

1 Setup Gitlab account

After login you will find this screen it is called the dashboard:



Fakher F. Assaad pushed to branch master at Fakher F. Assaad / FOR1807_II about 6 hours ago
2339db30 Update

Fakher F. Assaad pushed to branch master at Fakher F. Assaad / FOR1807_II about 6 hours ago
1e596d3f Uploaded P9_new

Corinna Kollath pushed to branch master at Fakher F. Assaad / FOR1807_II about 9 hours ago
33e79c59 version of P10

Fakher F. Assaad pushed to branch master at Fakher F. Assaad / FOR1807_II about 9 hours ago
bdb95688 Typos

Reinhard Noack pushed to branch master at Fakher F. Assaad / FOR1807_II about 12 hours ago
d5bb5da7 Try 2 in adding P2 .tex file and figs.

Reinhard Noack pushed to branch master at Fakher F. Assaad / FOR1807_II about 12 hours ago
eecd4e43 Merge branch 'master' of https://git.physik.uni-wuerzburg.de/fassaa...
eb78cff0 First draft of P2
Compare → 68de313f...eecd4e43

Frank Pollmann pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
68de313f added P3_new (at least a first version of it)

Frank Pollmann pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
6d75fd64 added P3_new (at least a first version of it)

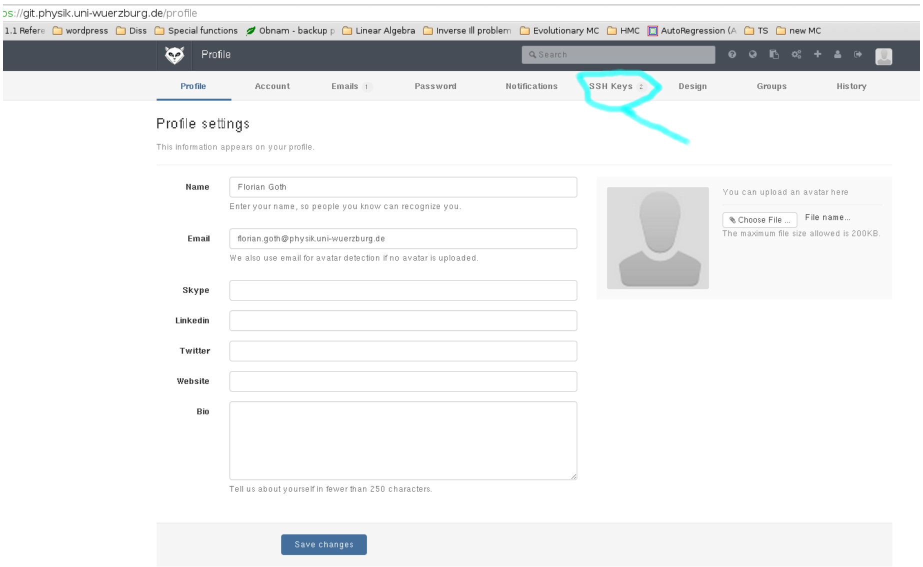
Eric Jeckelmann pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
fced9066 replaced by FOR1807_P4_abs.tex

Eric Jeckelmann pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
7369cdfb Merge branch 'master' of git.physik.uni-wuerzburg.de:fassaad/FOR180...
a7656df4 minor correction
Compare → 5ecd0030...7369cdfb

Stefan Wessel pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
5ecd0030 minor changes

Stefan Wessel pushed to branch master at Fakher F. Assaad / FOR1807_II 3 days ago
d22b3d82 Merge branch 'master' of https://git.physik.uni-wuerzburg.de/fassaa

As a first step we need to insert the SSH keys from the devices that will be used to access the repositories. As a benefit this will enable password less authentiation. To that end we need to visit the user area. I have highlighted the button via this funny cyan box in the top right corner. After that we are in the user screen. Here we need to access the section for managing SSH keys. I have again highlighted the relevant button in cyan.



Profile settings

This information appears on your profile.

Name: Florian Goth

Enter your name, so people you know can recognize you.

Email: florian.goth@physik.uni-wuerzburg.de

We also use email for avatar detection if no avatar is uploaded.

Skype:

Linkedin:

Twitter:

Website:

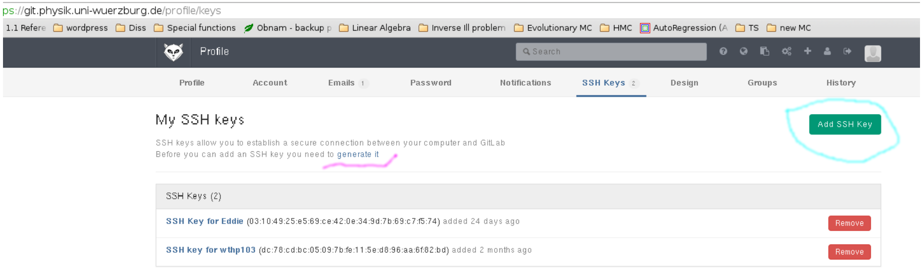
Bio:

Tell us about yourself in fewer than 250 characters.

You can upload an avatar here

The maximum file size allowed is 200KB.

After clicking the SSH key button you are in this screen:



My SSH keys

SSH keys allow you to establish a secure connection between your computer and GitLab
Before you can add an SSH key you need to generate it

SSH Keys (2)

SSH Key for Eddie (03:10:49:25:e5:69:ce:42:0e:34:9d:7b:69:c7:f5:74) added 24 days ago

SSH key for wthp103 (dc:78:cd:bc:05:09:7b:fe:11:5e:d8:96:aa:6f:82:bd) added 2 months ago

In this screen we can add an SSH key for every device that should be able to access the repository. Hints for the creation of an SSH key can be found at link marked in magenta. I have marked the button where you can upload an SSH key in cyan.
Once we have uploaded our keys we can start to checkout our project.

2 Some basic usage

The command is(the required path can be found on the overview page for a project):

```
git clone git@git.physik.uni-wuerzburg.de:fassaad/FOR1807_II.git
```

The most fundamental difference between git(or mercurial, bazaar) and tradtional version control systems is the fact that git is a *distributed* version control system. git has by design no notion of a central repository. The repository on your laptop is just as good as any other repository out there and could be used as reference by somebody else. Some basic commands are:

-
  ```
  git log
  ```
  This gives your current log.

-
  ```
  git status
  ```
  This gives an overview of the status of your files, e.g., are they modified or unknown to git

-
  ```
  git add < file >
  ```
  This marks a file for addition in the next commit

-
  ```
  git commit < file >
  ```
  This commits a file to YOUR repository. A space separated list of files is also possible

-
  ```
  git commit -a
  ```
  This commits everything.

-
  ```
  git pull
  ```
  This brings in changes from a REMOTE working copy. By default it is the one you cloned from. In our case this is the gitlab site

-
  ```
  git push
  ```
  This pushes all YOUR changes to the REMOTE site.

Except for push and pull none of these commands require a network connection. A simple workflow within your working copy would now look like this:

- What have the others done?
  ```
  git pull
  ```
- Work....
- Commit my work
  ```
  git commit file1.tex file2.tex
  ```
- make it available to the others
  ```
  git push
  ```

Good Luck.