

# Erste Schritte mit dem Gitlab-Server der Fakultät E

Stefan Gast

04.12.2018

Version: bf04098

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Erstmalige Anmeldung und Freischaltung</b>	<b>2</b>
2.1	Zugriff auf bestehende Projekte oder Gruppen . . . . .	2
2.2	Eigene Projekte anlegen . . . . .	2
<b>3</b>	<b>Einrichtung des lokalen Git-Clients</b>	<b>2</b>
3.1	Name und E-Mailadresse konfigurieren . . . . .	3
3.2	SSH einrichten . . . . .	3
3.2.1	SSH-Key erzeugen . . . . .	3
3.2.2	SSH-Key auf dem Server registrieren . . . . .	4
<b>4</b>	<b>Abschließende Hinweise</b>	<b>6</b>

## 1 Einführung

Die Fakultät E der Hochschule Ravensburg-Weingarten betreibt unter folgender Adresse einen Gitlab-Server für Teamprojekte:

<https://fbe-gitlab.hs-weingarten.de>

Diese Anleitung erklärt Ihnen, wie Sie Zugriff auf den Server bekommen und wie Sie Git auf Ihrem Rechner so einrichten, dass Sie mit den Repositories auf **fbe-gitlab** arbeiten können. Dazu werden grundlegende Kenntnisse im Umgang mit Git vorausgesetzt<sup>1</sup>.

---

<sup>1</sup>Es gibt zahlreiche allgemeine Git-Tutorials im Internet, wie z.B.: <https://rogerdudler.github.io/git-guide/index.de.html>

## 2 Erstmalige Anmeldung und Freischaltung

Als neuer User können Sie lediglich ein paar öffentliche Projekte sehen. Um Zugriff auf geschützte Projekte zu erhalten, müssen Sie zunächst freigeschaltet werden. **Damit das möglich ist, melden Sie sich bitte zuerst einmalig auf fbe-gitlab über die Web-Oberfläche an.** Kontaktieren Sie **danach** den entsprechenden Ansprechpartner:

### 2.1 Zugriff auf bestehende Projekte oder Gruppen

Nehmen Sie mit dem Eigentümer des Projektes oder der Gruppe Kontakt auf, damit dieser Ihnen die benötigten Rechte einräumt. Sollten Sie unsicher sein, hilft Ihnen der Dozent / Tutor der jeweiligen Veranstaltung oder der Betreuer Ihrer Abschluss- / Projektarbeit gerne weiter.

### 2.2 Eigene Projekte anlegen

Beantragen Sie per Mail an Stefan Gast <stefan.gast@hs-weingarten.de>, dass für Ihr Benutzerkonto persönliche Projekte aktiviert werden. Die Mail sollte fbe-gitlab im Betreff haben und zusätzlich folgende Informationen enthalten:

- Hochschul-Loginname
- Zweck, also z.B. „Grafische Bedienoberflächen“ oder „Projektarbeit“
- betreuende Lehrperson
- ggf. Anzahl der Projekte, falls mehr als eines benötigt wird

Sie erhalten eine Benachrichtigung, sobald die Freischaltung erfolgt ist.

## 3 Einrichtung des lokalen Git-Clients

Zum Abschluss der Einrichtung konfigurieren Sie Ihren lokalen Git-Client. Sie müssen zumindest Ihren Namen und Ihre Mailadresse in der Git-Konfiguration hinterlegen. Wie das geht, erklärt Abschnitt 3.1.

Optional können Sie sich einen SSH-Key einrichten, sodass Sie nicht bei jedem Zugriff des Clients auf den Server Ihre Zugangsdaten eingeben müssen. Die Vorgehensweise ist in Abschnitt 3.2 beschrieben.

## 3.1 Name und E-Mailadresse konfigurieren

In jedem Git-Commit wird der Name und die Mailadresse des Urhebers gespeichert. Damit innerhalb eines Teams die Urheberschaft der Commits klar ersichtlich ist, konfigurieren Sie bitte Ihren Namen und Ihre Mailadresse in Ihrem lokalen Git-Client korrekt. Benutzen Sie als Mailadresse Ihre Hochschuladresse der Form `loginname@hs-weingarten.de` (nicht `vorname.nachname@hs-weingarten.de`).

Zur Konfiguration öffnen Sie unter Linux oder auf dem Mac ein gewöhnliches Konsolenfenster. Unter Windows starten Sie die `Git Bash` über das Startmenü oder die Kacheloberfläche. Dort geben Sie dann die folgenden beiden Befehle ein (Name und Mailadresse natürlich anpassen):

```
git config --global user.name "John Doe"
git config --global user.email "jd-123456@hs-weingarten.de"
```

## 3.2 SSH einrichten

Ihr lokaler Git-Client kann über SSH oder HTTPS auf die Repositories zugreifen. Für eigene Rechner, auf die nur Sie Zugriff haben, wird der Zugriff über SSH empfohlen. Für öffentlich zugängliche Rechner wie die Laborrechner empfiehlt sich der Zugang über HTTPS.

Im Folgenden wird erklärt, wie Sie den Zugang über SSH einrichten. Für HTTPS ist keine zusätzliche Konfiguration erforderlich, allerdings müssen Sie sich dann bei jedem Serverzugriff mit Ihren Hochschuldaten einloggen.

Sofern Sie noch keinen SSH-Key auf Ihrem Rechner haben, müssen Sie sich zunächst einen erzeugen. Die Vorgehensweise wird in Abschnitt 3.2.1 erklärt. Wenn Sie bereits einen SSH-Key haben, dann springen Sie am besten gleich zu Abschnitt 3.2.2 auf Seite 4. Dort wird erklärt, wie Sie den SSH-Key auf `fbe-gitlab` registrieren.

### 3.2.1 SSH-Key erzeugen

Sie erzeugen einen SSH-Key mit dem Kommandozeilenprogramm `ssh-keygen`. Unter Linux oder auf dem Mac genügt zum Start des Programmes ein gewöhnliches Konsolenfenster. Unter Windows wählen Sie über das Startmenü oder die Kacheloberfläche den Menüpunkt `Git Bash`, um ein geeignetes Terminalfenster zu erhalten.

Nach dem Aufruf von `ssh-keygen` stellt das Programm einige Fragen. Wenn Sie nichts eingeben und einfach nur über `↵` bestätigen, dann werden die in den Klammern angegebenen Standardeinstellungen verwendet.

Wenn Sie noch keinen weiteren SSH-Key haben, dann belassen Sie es bei der Frage nach dem Speicherort am einfachsten bei der Standardeinstellung. Das Programm teilt Ihnen dann in seiner Ausgabe mit, wo es den Key ablegt („Your public key has been saved in...“).

```

$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/doz/gastste/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/doz/gastste/.ssh/id_rsa.
Your public key has been saved in /home/doz/gastste/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:M+Mk40SXe6FqMdpY3KJaWhPDqbGgDBQHbxFBjDHAXYY gastste@ap055049
The key's randomart image is:
+---[RSA 2048]-----+
|+=B*=o           |
| o=Eo            |
| . oo           |
|...= . .        |
|+ = o + S       |
|=+ oo. * =      |
|+.++ +o o       |
| +o.*...        |
|o...            |
+-----[SHA256]-----+

```

Listing 1: ssh-keygen in Aktion

Sie können den Key für zusätzliche Sicherheit mit einem Passwort versehen. Dieses wird immer dann abgefragt, wenn Sie den Key verwenden.

Listing 1 zeigt einen beispielhaften Durchlauf.

### 3.2.2 SSH-Key auf dem Server registrieren

Nun müssen Sie Ihren SSH-Key auf **fbe-gitlab** registrieren. Loggen Sie sich dazu auf dem Server ein. Klicken Sie dann auf den User-Button rechts oben im Bild und wählen Sie in dem aufklappenden Menü den Punkt **Settings** (siehe Abbildung 1).

Auf dem nun erscheinenden Bildschirm wählen Sie in dem Menü am linken Rand den Punkt **SSH Keys**. Daraufhin sehen Sie eine Maske ähnlich wie in Abbildung 2.

In das mit Key beschriftete Feld fügen Sie nun den Inhalt der Datei mit dem **öffentlichen** Teil des Schlüssels ein. In der Regel ist das `id_rsa.pub`. Öffnen Sie dazu die Datei mit einem Texteditor oder lassen Sie sich die Datei mit `cat` auf der Konsole ausgeben (Linux / Mac). Den Inhalt können Sie dann per Copy/Paste in das Textfeld übertragen.

**Achten Sie bitte unbedingt darauf, dass Sie den öffentlichen und *nicht* den privaten Teil des Schlüssels hochladen!**

Tragen Sie unter **Title** noch eine aussagekräftige Beschreibung ein, wie z.B. „Mein privates Notebook“. Anschließend können Sie mittels **Add key** die Registrierung abschließen.

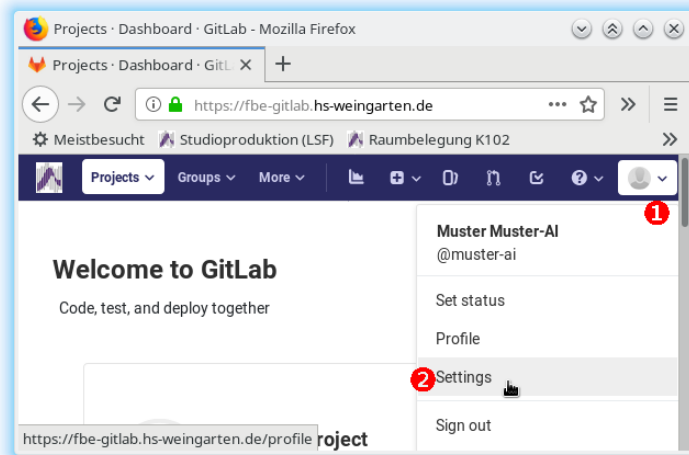


Abbildung 1: Auf die Profileinstellungen gelangen Sie, indem Sie auf den User-Button ❶ klicken und im daraufhin aufklappenden Menü den Punkt `Settings` ❷ auswählen.

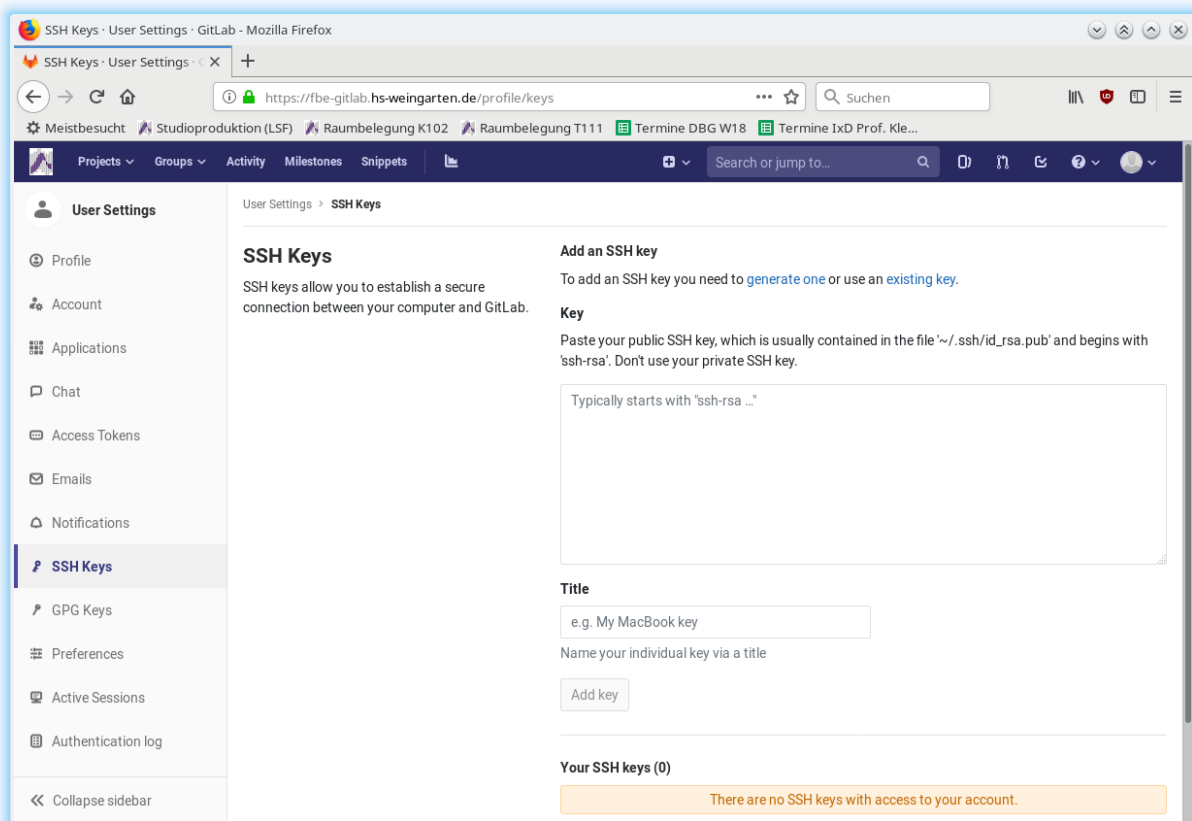


Abbildung 2: Eingabemaske für SSH-Keys

## 4 Abschließende Hinweise

Es folgen ein paar Hinweise zu Git allgemein und zum Server `fbe-gitlab` im Speziellen. Bitte berücksichtigen Sie diese, um eine gute Zusammenarbeit in Ihren Teams zu ermöglichen und eine reibungslose Funktion des Servers sicherzustellen.

- Schreiben Sie sinnvolle Commit-Messages! Eine Anleitung dafür finden Sie z.B. hier: <http://chris.beams.io/posts/git-commit/>
- **Checken Sie keine Build-Artefakte ein!** Diese verbrauchen nur unnötig Platz auf dem Server und verursachen vermeidbare Merge-Konflikte. **Auch Backup-Dateien, wie sie von manchen Editoren angelegt werden, sind in einem Git-Repository überflüssig.** Benutzen Sie in Ihren Projekten `.gitignore`-Dateien, um Ihre Repositories sauber und Ihre Patches klein zu halten.
- Der Platz auf dem Server ist begrenzt. Sprechen Sie es vorher mit der Administration des Servers ab, wenn Sie größere Projekte ablegen möchten (>100 MB).
- Die Administration behält sich vor, bei schwerwiegendem Fehlverhalten einzelne User zu blockieren oder Projekte zu löschen.
- Der Quellcode dieses Dokumentes ist auf `fbe-gitlab` verfügbar.<sup>2</sup> Verbesserungsvorschläge und sinnvolle Merge-Requests sind selbstverständlich willkommen.

---

<sup>2</sup><https://fbe-gitlab.hs-weingarten.de/adm-fbe-gitlab/tutorial>