

GITLAB DER FACHSCHAFT INFORMATIK – EINE EINFÜHRUNG

FRIEDRICH SCHÖNE | FSR INFORMATIK

WORUM GEHT ES

Das GitLab der Fachschaft Informatik ermöglicht es folgende Aufgaben zu erfüllen:

1. Vorlesungsmaterialien können zentral zusammengetragen werden und sind über eine Plattform gebündelt verfügbar.
2. Studenten wird ermöglicht, ihre Dokumente auszutauschen. Darunter fallen Mitschriften, Zusammenfassungen, Lösungen, etc.
3. Lernmaterialien aus vorherigen Semestern können gespeichert und so zukünftigen und aktuellen Studenten zur Verfügung gestellt werden.

Diese Einführung soll Studenten ohne Erfahrung mit Git die Arbeit mit dem System ermöglichen und auf weiterführende Informationen verweisen.

WAS IST GIT

Git ist eine Software zur Versionsverwaltung. Es dient der Überwachung von Änderungen in Dateien und Ordnerstrukturen. Außerdem koordiniert es die dezentrale Bearbeitung dieser Dateien.

Die Datenstruktur, in der ein Projekt auf einer Maschine gespeichert wird, heißt *repository*. Dieses besteht aus den Dateien und der gesamten Versionsgeschichte.

Die Speicherung einer Dateiänderung nennt sich *commit*. Über die commit-Geschichte ist es möglich zu einzelnen Speicherpunkten im Projekt zurückzukehren.

Um ein repository mit anderen repositorys zu synchronisieren nutzt man *pull* (Änderungen anfordern) und *push* (Änderungen senden).

Mehr über Git auf der offiziellen About-Seite:

<https://git-scm.com/about>

WAS IST GITLAB

GitLab erweitert Git um verschiedene Möglichkeiten des Projekt- und Usermanagements. Die zusätzlichen Funktionen sind über eine Weboberfläche nutzbar.

Am besten lässt sich GitLab durch eine Kombination aus Weboberfläche und Kommandozeile nutzen.

GitLab: <https://about.gitlab.com/>

BEDIENUNG ÜBER DIE WEBOBERFLÄCHE

Der Login funktioniert über den LDAP-Account. Dieser wird den Informatikstudenten für die Nutzung der Computerpools zur Verfügung gestellt.

Alle vorhandenen Module haben ihr eigenes repository. Diese sind auf der Startseite der Weboberfläche aufgelistet. Sollte ein Modul noch nicht angelegt worden sein, kann dieses mit einer Email an fheusel@informatik.uni-leipzig.de angefragt werden.

Auf der Übersichtsseite der einzelnen Module wird ein Beschreibungstext angezeigt. Dieser kann durch die Änderung der *Readme.md*-Datei im Wurzelverzeichnis bearbeitet werden.

Unter dem Navigationspunkt Files kann die Dateistruktur angesehen werden. Außerdem können Dateien runtergeladen, bearbeitet und hochgeladen werden.

BEDIENUNG ÜBER DIE KOMMANDOZEILE

Für eine bessere Bedienung empfiehlt sich die Nutzung des Kommandozeilenprogramms Git. Es ermöglicht das Verwalten der Dateien über das eigene Dateisystem und ein einfacheres Hoch- und Herunterladen.

Installationsdateien von der offiziellen Webseite:

<https://git-scm.com/downloads>

NOTWENDIGE FUNKTIONEN

CLONE

```
$ git clone <adresse>
```

Der Befehl kopiert das entsprechende repository auf den lokalen Computer. Damit ist der aktuelle Stand im Dateisystem und die Versionsgeschichte in den Git-Metadaten gespeichert.

Die Adresse eines repositorys findet man auf der Übersichtsseite des jeweiligen Moduls.

Alle weiteren Befehle müssen im Ordner des geklonten repositorys ausgeführt werden.

<https://git-scm.com/docs/git-clone>

PULL

```
$ git pull
```

Die letzten commits, die im repository auf dem Server gespeichert wurden, werden mit diesem Befehl geladen. Es empfiehlt sich vor jeder Änderung einen pull auszuführen um eventuelle Konflikte zu vermeiden.

<https://git-scm.com/docs/git-pull>

ADD

```
$ git add <dateiname|ordnername>
```

Der Befehl bereitet die Dateien für den commit vor. Es werden so die Dateien markiert, deren Änderungen im nächsten commit gespeichert werden sollen.

<https://git-scm.com/docs/git-add>

COMMIT

```
$ git commit -m "<kommentar>"
```

Erstellt einen commit mit den Änderungen.

Git fordert nicht zwingend einen Kommentar, dieser sollte jedoch in jedem Fall hinterlegt werden damit einfach erkennbar ist, was zu diesem Zeitpunkt geändert wurde.

Einige Hinweise zu commit-Nachrichten: <https://chris.beams.io/posts/git-commit/>

<https://git-scm.com/docs/git-commit>

PUSH

```
$ git push
```

Sendet erstellte commits an den Server und synchronisiert so die repositorys.

<https://git-scm.com/docs/git-push>

HILFREICHE FUNKTIONEN

STATUS

```
$ git status
```

Der Befehl bezieht sich auf den letzten commit und den aktuellen Stand des Dateisystems. Es werden alle Dateien angezeigt, die nicht geändert wurden.

<https://git-scm.com/docs/git-status>

LOG

```
$ git log
```

Mit diesem Befehl lässt sich die commit-Geschichte anzeigen.

<https://git-scm.com/docs/git-log>

HINWEISE

- Dateien die in einem Git-repository gespeichert werden, befinden sich für immer in der Versionsgeschichte und lassen sich daraus extrahieren.
- Im FSR-GitLab werden die Materialien nach Semester geordnet. Die Ordner der einzelnen Semester heißen „WiSeXX“ oder „SoSeXX“ wobei „XX“ die entsprechende Jahreszahl ist, in der das Semester begonnen hat. Eine weitere Sortierung in Unterordnern wie „Übungen“, „Skript“ oder „Klausuren“ ist sinnvoll.
- Auf der Übersichtsseite zu jedem repository wird eine Readme.md-Datei angezeigt. Diese in Markdown verfasste Datei liegt im Wurzelverzeichnis und kann einen Überblick über die vorhandenen Dateien geben.

Hinweise zu Markdown in GitLab:

<https://docs.gitlab.com/ee/user/markdown.html>

- Um bei push oder pull nicht immer sein Passwort eingeben zu müssen, ist es möglich in GitLab seinen SSH-Key zu hinterlegen.

Erklärung zu SSH-keys:

https://wiki.archlinux.org/index.php/SSH_keys

Hinterlegen des keys in GitLab:

<https://docs.gitlab.com/ce/ssh/README.html>

- Das FSR-GitLab lebt von den Studenten und wird besser durch deren Unterstützung. Grundsätzlich bereichern alle Dokument die hochgeladen werden das GitLab, das bedeutet auch Vorlesungsfolien, Mitschriften, Cheatsheets, Gedankenprotokolle und Klausuren.

WEITERFÜHRENDE LINKS

Um Git zu erlernen stellt Github ein Onlinetraining bereit.

Das Onlinetraining: <https://try.github.io/>

Einzelne Funktionen von Git und deren Gebrauch kann man auf verschiedenen Cheatsheets nachsehen.

Git Cheatsheet:
<https://services.github.com/on-demand/downloads/de/github-git-cheat-sheet/>

Natürlich stellen die Institutionen GitLab und GitHub auch ein Cheatsheet zur Verfügung.

Cheatsheet von GitLab:
<https://gitlab.com/gitlab-com/marketing/raw/master/design/print/git-cheatsheet/print-pdf/git-cheatsheet.pdf>

Cheatsheet von Github:
<https://rogerdudler.github.io/git-guide/index.de.html>
