

1 Einführung in die Arbeit mit Git und GitLab

Git ist eine populäre Software zur (verteilten) Versionsverwaltung von Dateien. **GitLab** ist (ähnlich wie GitHub) eine Weboberfläche zum Verwalten von Git-Repositories. Das folgende Dokument führt in die für dieses Praktikum benötigte Infrastruktur ein.

1.1 Benötigte Software für die Arbeit mit Git

Wenn Sie lokal auf Ihrem Rechner arbeiten möchten, benötigen Sie für dieses Praktikum

- NetBeans 8.1 oder 8.2 mit einem Gradle-Plugin¹ und JDK8 (andere IDEs sind grundsätzlich möglich, werden von uns aber nicht im Praktikum unterstützt)
- Wir empfehlen, zusätzlich zum in NetBeans integrierten Git-Client die Software TortoiseGIT zu installieren: <https://tortoisegit.org/>
TortoiseGIT integriert sich in den Windows-Explorer, sodass alle Funktionen im Explorer über das Kontextmenü (rechte Maustaste) aufrufbar sind.
- Zur Nutzung von TortoiseGIT benötigen Sie außerdem GitForWindows:
<https://git-for-windows.github.io/>
- Falls Sie Linux verwenden, installieren Sie ein Git package Ihrer Wahl.

1.2 GitLab

Registrierung im GitLab

Wichtig: Dieser Schritt ist bereits **vor** dem 1. Praktikumstermin zu erledigen.

Das GitLab für diese Lehrveranstaltung steht Ihnen auf dem Server `dakala.fbi.h-da.de` zu Verfügung. Beim Aufruf der GitLab-URL werden Sie automatisch auf die Login/Registrierungs-Seite weitergeleitet.

Im Registrierungsformular geben Sie bitte unter **Name** Ihren korrekten Vor- und Nachnamen an, um später eine Zuordnung zu ermöglichen! Wählen Sie dann noch einen beliebigen Benutzernamen sowie ein beliebiges Passwort, tragen eine korrekte Email-Adresse ein und registrieren sich. **Merken sich diese Daten bitte gut!**

Hinweis: Ihre Daten sind sicher. Nur die Datenbanken 2 Dozentinnen und Dozenten können auf Ihre Email-Adresse zugreifen.

Zuordnung eines Projektes im GitLab

Nachdem Sie sich registriert haben, müssen wir Sie einem Repository zuordnen. Bitte teilen Sie uns dafür spätestens zum Beginn des ersten Praktikum-Termins Ihren Namen, den Namen Ihres Praktikums-Partners und Ihren Praktikumsblock mit.

Wir ordnen Ihrer Gruppe dann ein gemeinsames Repository mit den benötigten Rechten zu. Die Repositories haben die gleiche Namensgebung wie die Oracle-Datenbank-Accounts:

http://dakala.fbi.h-da.de/sose17/stud****

¹Tools->Plugin->Available Plugins->Gradle Support->Install

Clonen eines Projektes

Um Ihnen die Arbeit zu erleichtern, haben wir bereits einen Projektrahmen (wird unten näher erläutert) vorbereitet. Als erstes müssen Sie sich eine lokale Kopie des Projektes erzeugen - dieser Schritt heißt bei Git Clonen. Sie können das Clonen mit den installierten Git-Tools (siehe Abschnitt 1.1) über das Dateisystem (`git clone`) oder über Netbeans ausführen:

- Team->Git->Clone oder Team->Remote->Clone
- Unter Repository URL die URL Ihres Repositories:
`http://username@dakala.fbi.h-da.de/sose17/stud****`
eingeben, wobei *username* Ihr registrierter Benutzername im GitLab und *stud***** Ihr zugeordneter Oracle-Datenbank-Account ist.
Anmerkung: Sie können diese URL auch in der Projektansicht im GitLab kopieren (ggf. von SSH auf HTTP umschalten)
- Unter Specify Destination Folder ein geeignetes Verzeichnis auf Ihrem Rechner auswählen.
- Alle weiteren Angaben im Netbeans können Sie ohne Veränderung übernehmen.

Anmerkung: Sie können mit GitLab über HTTP unter Verwendung des Passwortes arbeiten oder über SSH. Falls Sie mit SSH arbeiten möchten, müssen Sie einen SSH Key im GitLab hinterlegen:
<https://docs.gitlab.com/ee/gitlab-basics/create-your-ssh-keys.html>

Projektrahmen

Das vorbereitete Projekt kann direkt in Netbeans ohne weitere Anpassungen verwendet werden. Das eingesetzte Build-Automatisierungssystem heißt Gradle und muss von Ihnen in der Regel nicht angepasst werden, da bereits alles lauffähig konfiguriert ist.

Der Projektrahmen ist in folgende Bereiche aufgeteilt:

- In `src/main/java` werden die Java-Sources abgespeichert (Package `de.h-da.fbi.db2`)
- In `src/main/resources` werden die zum Projekt gehörenden Ressource-Dateien abgespeichert.

Wichtiger Hinweis! Alle von Ihnen entwickelten Java-Klassen und -Packages dürfen nur als Unter-Packages von `de.h-da.fbi.db2.stud` abgelegt werden, damit der automatisierte Build-Prozess (siehe Abschnitt 2) funktioniert! Beispiel: `de.h-da.fbi.db2.stud.entity`

1.3 Wichtige Git-Kommandos

Add (Selektieren Ihrer Änderungen zum Commit) `git add`

Registriert Dateien im Git für den nächsten Commit. Dabei ist zu beachten, dass bei Anwendungen nur der Quellcode, nicht aber die fertigen Libraries/Programme hinzugefügt werden dürfen!

Bitte adden und committen Sie Ihre Projekt-Dateien ausschließlich über NetBeans! Diese Funktion ist nur für zusätzliche Dateien außerhalb des Projektordners gedacht.

Commit (Hinzufügen Ihrer Änderungen zum lokalen! Repository) `git commit`

Dieses Kommando fügt Ihre Änderungen zum lokalen Repository hinzu. Sie müssen hierbei eine Commit-Message angeben, welche die Änderungen beschreibt. Eine Übertragung zum Server findet, im Gegensatz zu SVN, beim Commit nicht statt!

Push (übertragen Ihrer Änderungen auf den Server) `git push`

überträgt Ihre Änderungen zum Git-Server. Ggf. muss vorher ein Pull gemacht werden.

Pull (Abrufen von Änderungen vom Server) `git pull`

Lädt Änderungen vom Server herunter. Ggf. müssen diese Änderungen mit den lokalen Änderungen zusammengefügt (merge) werden.

Für Studierende, die über keine Erfahrung mit Git verfügen, empfehlen wir, nicht gleichzeitig an mehreren Rechnern zu arbeiten, und jeweils unmittelbar **vor** der Arbeit ein Pull und unmittelbar **nach** der Arbeit ein Push zu machen (Commit vor dem Push nicht vergessen!).

Dokumentation

Eine gute Einführung in die Git-Workflows findet sich hier:
<https://www.atlassian.com/git/tutorials/comparing-workflows>.

Eine Online-Version des Buches *Pro Git* finden Sie unter
<https://git-scm.com/book/en/v2>. Dieses Buch enthält eine umfassende Dokumentation aller Kommandos bzw. Möglichkeiten, die Git bietet.

2 Weitere Hinweise zur Arbeit mit dem GitLab

CI – Continuous Integration

Das GitLab führt automatisch Builds und Tests Ihrer Commits durch. Damit können Sie und wir schon im Vorfeld prüfen, ob Ihr aktueller Stand den Anforderungen entspricht. Wir oben beschreiben, setzen wir dabei voraus, dass Sie Ihr Projekt in dem von uns vorgesehenem Package entwickeln: `de.h_da.fbi.db2.stud`. Wenn Sie sich nicht an diese Anforderung halten, erhalten Sie kein Testat. Den Stand dieser Builds und Tests können Sie über
http://dakala.fbi.h-da.de/sose17/stud****/pipelines abrufen.

Source Code Qualität

Im GitLab wurden Reports zur automatischen Kontrolle der Qualität Ihres Source Code integriert. Dieses können und sollten Sie unter
<http://dakala.fbi.h-da.de/reports/> anschauen.

Weitere Hinweise dazu finden Sie in einem separaten Dokument.