

GIT/GITLAB: Fragen und Übungen

Eric Walke, 43378
Elia Baisch, 43506
Michael Dick, 43396
Tom Flocken, 43366

.....
Vorname, Name, Matrikelnummer

1. Was genau sind die Gründe um Gitlab zu verwenden?

- Orderly collaboration in a large team (work on the same project in parallel)
- Can also be used offline (local only)
- Simple handling
- One code base

2. Welche Daten gehören (nicht) ins Repo?

- .java Files
- .xml Files
- .Jason Files
- Bilddateien für Gameprojekte
- Musikdateien für Projekte
- .project Files (Endung je nach Tool)
- UML Modelle
- Zeichnungen
- Notizen
- Dokumentation
- Configurationsfiles
- Kapitel eines Buches
- Eine Bachelorarbeit
- Passwörter für Cloud-Services
- Passwörter für lokale Services (Self-hosted)
- logfiles
- Messdaten vom Profiling

3. Was soll der „Mist“ mit den Stages (dass add/commit nur lokal wirken)?

This ensures that only the code you have previously added ends up on Git.

If every commit/add were to write directly to the remote, it could get chaotic when working in parallel.

4. Würden Sie in einer Firma Gitlab selber hosten oder GIT als Service im Netz? Begründung.

Here it clearly depends on the preferences of the company. For example, if modularity and data security is an important point, then it would be advisable to use your own Gitlab service. Personally, I would recommend a private Gitlab hosting for a company.

5. Verwenden Sie Branches im Projekt oder arbeiten alle Teammitglieder auf dem Master Branch?

Zeigen Sie Vor- und Nachteile der Verfahren

Master Advantages: clearer, only one fallow is used. This is always the newest.

Master disadvantages: if many work at the same time on the project, it can come to complicated merge conflicts.

Branch advantages: organized code base. Each branch can e.g. stand for a single feature.

Branch disadvantages: can become complex with time.

6. Wie veröffentlichen Sie Ihre Änderungen auf dem globalen Repository? Wie oft checken Sie Ihre Änderungen im globalen Repo ein? Was ist besser: Nach jeder Änderung, nach einigen Änderungen, wenn Sie ein zusammenhängendes Stück fertig haben, wenn Sie eine Änderung machen die viele Kollegen betrifft, einmal am Tag, einmal die Woche. Wieso?

When you have a coherent piece ready, since you thus do not check in too many individual changes but also not too few. Logically, the code simply belongs together.

7. Eine Version eines Files im Repo sieht so aus:

„There are two versions of GitLab: Community Edition (CE) and Enterprise Edition (EE)“

Ihre Version die Sie hochladen wollen hat an dieser Stelle:

„There are two versions of GitLab: CE and EE.“

Wie löst Gitlab diesen Konflikt?

Here a "merge conflict" has occurred. Gitlab offers the user to "merge" these two merge into one to fix the error.

8. Hausaufgabe für jedes Team: Es muss eine chain-story (Kettengeschichte) erzählt werden bei der ALLE Teammitglieder jeweils immer einen Satz erfinden und dann einchecken. Ich will von ALLEN Teammitgliedern Sätze im Repo finden! Danach sagen Sie uns wie es ging. Haben Sie über externe Kanäle kommuniziert (Slack?). Anderweitig abgesprochen?

We communicated mainly via Whatsapp and on site at the university.