

Übung - Vorbereitung

GitLab

Für den Austausch von Daten mit den Tutoren wird das GWDG GitLab <https://gitlab.gwdg.de/> benutzt.

Legen Sie im GWDG GitLab ein eigenes Projekt mit *Visibility Level* **Private** an, das Sie ausschließlich für die Übungen zu *Allgemeines Programmierpraktikum* im SoSe 2023 verwenden.

Öffnen Sie in diese Projekt *Settings*→*General*→*Visibility, project features, permissions* und überprüfen Sie Folgendes

- *Visibility Level* ist **Private**.
- *Repository* ist aktiviert für **Only Project Members**.
- Alles andere ist deaktiviert oder ebenfalls auf **Only Project Members** eingeschränkt.

Beantragen Sie die Mitgliedschaft in der Gruppe <https://gitlab.gwdg.de/app/2023ss/testat> Den Link *Request Access* finden Sie auf der Übersichtsseite hinter der *Group ID* und **nicht** unter *Members*. Die Mitgliedschaft wird manuell bewilligt, deshalb erfolgt die Eintragung nicht sofort.

Sobald Sie Mitglied von [app/2023ss/testat](https://gitlab.gwdg.de/app/2023ss/testat) sind, fügen Sie diese Gruppe ihrem Projekt hinzu. Benutzen Sie dazu in Ihrem Projekt den Tab *Invite group* (**nicht** *Invite member*) unter *Projekt Information*→*Members*. Wählen Sie *Developer* als *Max access level*.

Durch dieses Vorgehen bekommen die Tutoren Zugriff auf Ihr Repository. Alle anderen Teilnehmer, die - wie Sie - Gäste in der Gruppe [app/2023ss/testat](https://gitlab.gwdg.de/app/2023ss/testat) und damit auch Gäste in Ihrem Projekt sind, haben keinen Zugriff auf Ihr Repository. Gäste können aber z.B. im Wiki lesen oder das Issues Tracking benutzen, wenn Sie das nicht möchten, können Sie einzelne Dienste unter *Settings*→*General*→*Visibility, project features, permissions* deaktivieren.

Git

Haben Sie Ihr GitLab Projekt erfolgreich angelegt, können Sie für die Lösung der Übungen eine lokale Arbeitskopie erstellen. Die Studentin **worker** mit dem Projekt **app-dev** erledigt das wie folgt.

```
git clone https://gitlab.gwdg.de/worker/app-dev.git
```

Dadurch entsteht ein Verzeichnis `app-dev`.

Hinterlegen Sie als erstes folgende Datei `.gitignore`, die dafür sorgt, dass keine Objekt-Files etc. ins Repository übernommen werden. Diese Datei können und sollten Sie an Ihre Bedürfnisse anpassen.

```
# .gitignore
# compiled class files
*.class
# object files
*.o
# log files
*.log
# archive files
*.tar
*.gz
*.tgz
# editor backups
*~
\#*\#
```

Legen Sie in Ihrer lokalen Arbeitskopie ein Unterverzeichnis für die aktuelle Übung, z.B. `uebung01`, an. Speichern Sie darin alle Lösung für die aktuelle Übung, wenn nötig in weiteren Unterverzeichnissen.

Um die Dateien in GitLab zu übertragen können Sie u.a. folgende Anweisungsfolge verwenden.

```
git add .
git commit -a -m "Kommentar"
git push
```

Abgabe (*tagging*)

Sind Sie mit der Bearbeitung der aktuellen Übung in Ihrer lokalen Arbeitskopie fertig und haben alle **Änderungen ins GitLab übertragen**, kennzeichnen Sie das mit dem *annotated tag* `exerciseXX` (z.B. `exercise01`) wie folgt.

```
git tag -a exercise01 -m "Kommentar"
git push --tag
```

Das **tagging** muss spätestens am **Abgabetermin** erfolgen.

Sie können danach in der Arbeitskopie weiterarbeiten, **der Tutor wird beim Testat seine lokale Kopie auf den Zeitpunkt des *tagging* zurücksetzen**.

Hinweis

Scott Chacon, Ben Strau; **Pro Git**; *Git Basics - Tagging*
<https://git-scm.com/book/en/v2/Git-Basics-Tagging>