

Umzug eines Repositories

via `git push --mirror`

Der manuelle Umzug per `--mirror` ist ideal, wenn du ein reines Git-Repository inklusive aller **Branches, Tags und der kompletten Commit-Historie** übertragen möchtest – ganz ohne GitHub-Tokens oder Metadaten wie PRs und Issues.

Hier ist der genaue Ablauf über das Terminal:

1. Leeres Repository in Forgejo erstellen: Auf der Ziel-Plattform.

1. Melde dich bei deiner Forgejo-Instanz an.
2. Klicke auf das **+**-Symbol oben rechts und wähle **Neues Repository**.
3. Gib den Repository-Namen ein.
4. **Wichtig:** Initialisiere das Repository **nicht** (keine README, keine `.gitignore` und keine Lizenz anlegen lassen). Es muss vollkommen leer sein.
5. Kopiere die Git-URL des neuen Repositories (z. B. `[https://forgejo.dein-server.de/user/repo.git]` (`https://forgejo.dein-server.de/user/repo.git`) oder per SSH `git@forgejo.dein-server.de:user/repo.git`).

2. Bare-Clone des GitHub-Repositories erstellen: Lokal auf deinem Computer.

Erstelle einen temporären "Bare-Clone" des GitHub-Repositories. Dieser enthält nur die Git-Datenbank ohne Arbeitsverzeichnis:

Bash

```
git clone --bare https://github.com/nutzername/repository.git mirror-temp.git
cd mirror-temp.git
```

3. Repository zu Forgejo spiegeln:

Pushe den gesamten Stand (alle Referenzen, Branches und Tags) per Mirror-Option zu Forgejo:

Bash

```
git push --mirror https://forgejo.dein-server.de/user/repo.git
```

(Wenn du SSH nutzt, verwende die SSH-URL: `git push --mirror git@forgejo.dein-server.de:user/repo.git`)

4. Aufräumen und Remote-URL aktualisieren: Säubern und bestehende Arbeitskopie anpassen.

1. Lösche den temporären Bare-Ordner wieder:

Bash

```
cd ..  
rm -rf mirror-temp.git
```

2. Wenn du bereits einen normalen Arbeitsordner des Repositories auf deinem Rechner hast, wechsele hinein und passe die Remote-URL an:

Bash

```
cd dein-projekt-ordner  
git remote set-url origin https://forgejo.dein-server.de/user/repo.git  
git fetch origin
```

Was macht `--mirror` anders als ein normaler Push?

- `git push --all` überträgt nur alle lokalen Branches.
- `git push --mirror` spiegelt den *exakten* Zustand aller Referenzen: Lokale Branches, Remote-Tracking-Branches, Tags sowie Notizen (`refs/notes/`). Es überschreibt das Ziel-Repository komplett, damit es exakt dem Quell-Repository entspricht.