

Git integratie in JupyterLab

criterium 3,10,11?

In dit bewijsstuk laat ik zien hoe ik Git en Github heb gekoppeld aan mijn JupyterLab-omgeving. Ik werk lokaal aan code en analyses en wil deze veilig opslaan. Ook wil ik dat mijn werk reproduceerbaar is en versiebeheer heeft. Door mijn lokale map te koppelen aan een repository op Github kan ik: - al mijn code centraal bewaren en terugvinden, - veranderingen stap voor stap bijhouden (versiebeheer), - mijn werk veiligstellen, - en samenwerken of mijn project eenvoudig delen.

Situatie en keuze

Ik had al een lokale map (`ml_oncoreponse`) met bestanden waarin ik werkte. Op het gedeelde schoolaccount (`ProjecticumDlerpDs`) heb ik een lege repository met dezelfde naam (`ml_oncorepsonse`) aangemaakt op Github. Bij het koppelen van een lokale map aan Github zijn er twee mogelijkheden: 1. **git clone**: dit haalt een bestaande repository van Github naar je computer. Dit is handig als je wilt beginnen vanuit een repository die al gevuld is met bestanden. In mijn situatie werkte dit niet goed omdat ik al bestanden lokaal had en op een gedeelde schoolaccount beperkte rechten had, kreeg ik foutmeldingen bij het klonen. 2. **git init**: dit zet een bestaande lokale map om in een Git-repository, waarna je de remote (GitHub) kunt koppelen. Dit paste beter bij mijn situatie aangezien ik lokaal al bestanden had en mijn Github-repository leeg was.

Daarnaast had ik bewust geen README toegevoegd bij het aanmaken van de nieuwe repository op Github om conflicts te voorkomen. Ik weet nog van project DlerpA dat dit problemen opleverde. Gezien mijn situatie heb ik dus gekozen voor de tweede optie, **git init**.

Lokale map koppelen aan Github.

1. Maak een lege map voor Git:

```
cd ~  
mkdir ml_oncoreponse  
cd ml_oncoreponse
```
2. Initialiseer Git en hernoem branch naar main:

```
git init  
git branch -M main
```
3. Voeg de remote GitHub-repo toe: `git remote add origin https://github.com/ProjecticumDlerpDs`

```
git remote -v
```
4. Kopieer bestaande bestanden vanuit backup-map:

```
cp -r ../ml_oncoreponse_backup/* ./
```

5. Voeg alle bestanden toe aan Git:
`git add --all`
6. Maak de eerste commit:
`git commit -m "Eerste commit: bestaande bestanden toegevoegd"`
7. Push naar Github:
`git push -u origin main`
8. Authenticatie:
 - Maak een Personal Access Token (PAT) aan in je Github-account.
 - Gebruik je Github-gebruikersnaam als username en de PAT als wachtwoord bij push.

Problemen die zijn opgetreden en hoe deze zijn opgelost

Clone in bestaande map:

Foutmelding: “destination path already exist and is not an empty directory”

Oplossing: ik heb een backup map gemaakt met de bestanden, en er voor gezorgd dat de map `ml_oncoresponse` leeg is.

Submodule-fout bij git add: Foutmelding: error: “‘ml_oncoresponse/’ does not have a commit checked out” Oplossing: gebruiken `git add -all` in plaats van de mapnaam

Authenticatieprobleem bij push:

Foutmelding: “Password authentication is not supported for Git operations”

Oplossing: PAT aanmaken en gebruiken in plaats van wachtwoord.

Conflict door README (evaring uit vorig project): Probleem: Bij een eerder project liep ik vast toen ik in Github bij het aanmaken van een repository een README toevoegde. Hierdoor had de remote al een commit, waardoor ik mijn lokale bestanden niet direct kon pushen (foutmelding: "failed to push some refs"). Oplossing/Voorkomen: Bij dit project heb ik expres **geen README toegevoegd** bij het aanmaken van de repository, zodat de remote leef bleef en ik mijn repo zonder problemen kon koppelen.

Conclusie

- De lokale map is nu volledig gekoppeld aan de Github-repo op het gedeelde schoolaccount.
- Alle bestanden zijn succesvol gepusht.
- Dagelijkse workflow: `git add -all -> git commit -> git push`
- Probleemoplossend vermogen getoond door submodule-issues, clone-conflict, en authenticatieproblemen op te lossen.
- Bewuste keuze gemaakt voor `git init` in plaats van `git clone`, passend bij de situatie van een bestaande lokale map en beperkte rechten op het schoolaccount.

- Geleerd van een eerder project: door bij het aanmaken van de repo geen README toe te voegen, voorkwam ik push-conflicten die ik eerder wel had ervaren.