

Git standaard workflow – Data Technologies

R.M. van Hoek, Versie: 1.0, Datum: 14 december 2025

Dit document is een **persoonlijk notitieblok** met de vaste Git-commando's die ik gebruik voor het vak **Data Technologies**.

Workflow:

1. Werk gebeurt in de **development** branch
2. Wijzigingen worden gemerged naar **main**
3. **main** wordt gepusht naar:
 - de **private repository (origin)**
 - de **public repository (public)**

1. Overzicht branches

```
git branch --list
```

Controleer op welke branch je zit:

```
git status
```

2. Development branch bijwerken

Ga naar de development-branch:

```
git checkout development
```

Haal de laatste wijzigingen op:

```
git pull
```

Let op: als Git meldt dat de branch *behind* is en kan worden *fast-forwarded*, dan is dit de juiste stap.

3. Merge van development → main

Ga terug naar **main**:

```
git checkout main
```

Merge de development-branch:

```
git merge development
```

Controleer de status:

```
git status
```

4. Push naar private GitHub repository

Push **main** naar de private repository (**origin**):

```
git push origin main
```

(of kort:)

```
git push
```

5. Push naar public GitHub repository

Controleer eerst de remotes:

```
git remote -v
```

Verwacht resultaat:

- **origin** → private repository
- **public** → publieke repository

Push **main** ook naar de public repository:

```
git push public main
```

6. Handige controle-commando's

Laat recente Git-commando's zien:

```
history | grep "git"
```

Hulp bij checkout:

```
git checkout --help
```

Hulp bij branches:

```
git branch --help
```

Samenvatting (kort)

```
git checkout development
git pull
git checkout main
git merge development
git push origin main
git push public main
```

Dit is de vaste routine voor het bijwerken en publiceren van de Data Technologies lesstof.