NOVI HOGESCHOOL

Frontend & backend leerlijn

Git commandos

Deel 1

Installatie

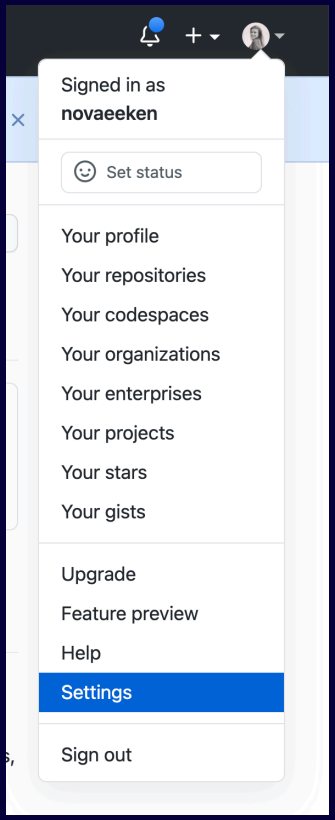
Deze stappen zijn eenmalig

- GitHub access token aanmaken
- Installatie Git
- Instellen van Git globals

GitHub email-adres

Heb jij jouw @novi-education.nl email gebruikt om je te registreren bij GitHub? **Dat is niet zo handig!** Je GitHub account neem je altijd met je mee, terwijl jouw Novi-email na afstuderen wordt opgeheven.

Pas dit gelijk even aan naar je privé-email adres bij *Settings > Email > "Add email address"*



Access token genereren

1. Ga op GitHub naar Settings > Developer Settings > Personal Access Tokens

2. Klik op “Generate a new token”
3. Geef de token een naam (bijv. “Pieters token”)

4. Selecteer alleen **repo** bij de permissies
5. Klik op “Generate token”

6. **Kopieer de token en plaats deze tijdelijk in je notities** (Heb je straks nodig!)

https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token

Installatie

GIT is standaard geïnstalleerd op nieuwere Mac versies, maar **niet** op Windows en Linux

Check of GIT geïnstalleerd is door jouw IDE te openen, de terminal te openen en dan het volgende commando te typen:

```
git --version
```

Geen versienummer? Dan moet het nog handmatig geïnstalleerd worden

Installatie Windows

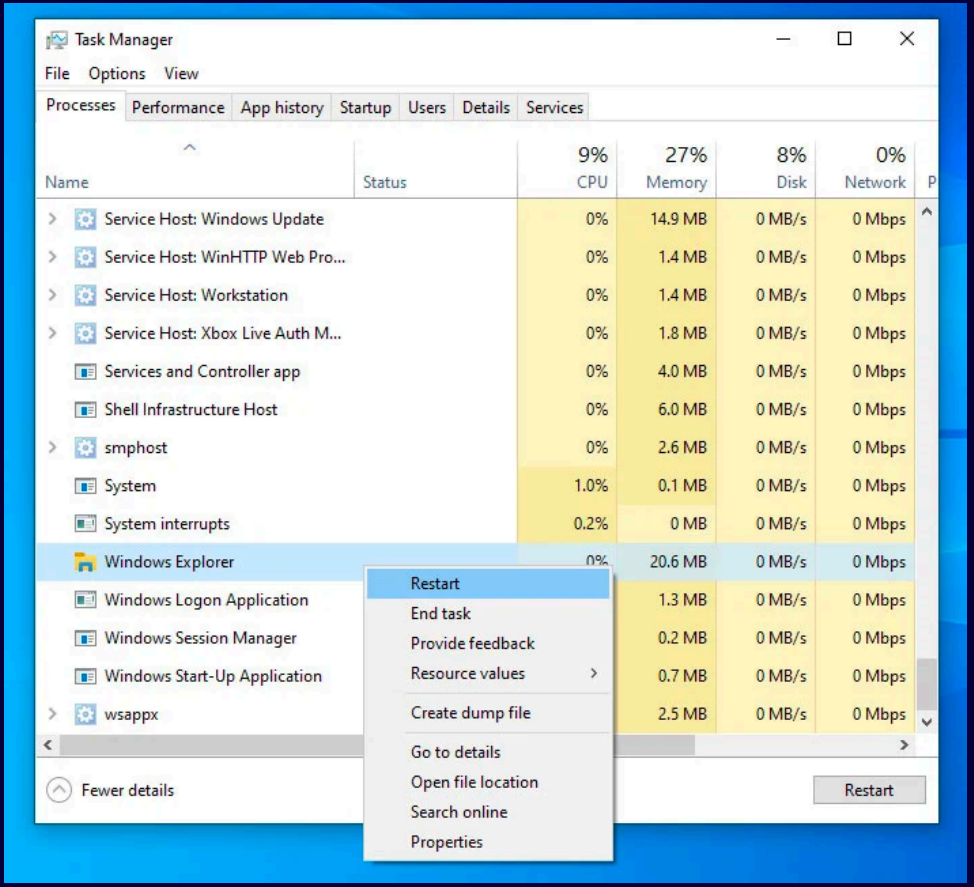
https://git-scm.com/downloads

* Dus **niet** de GitHub Desktop Manager!

Bij het installeren mag je overal op ‘volgende’ klikken en de recommended opties kiezen. Lekker doorklikken dus!

GIT hernoemt sinds een tijdje alle master branches naar main. Bij het installeren kan het zijn dat gevraagd wordt of dit automatisch mag. Je mag kiezen voor *“Let Git decide!”*

Installatie Windows



Daarna rechtermuisknop op de start**balk** > Taakbeheer > Windows Explorer opnieuw opstarten (belangrijk!)

Sluit ook **al** je running IDE's af (zoals WebStorm of IntelliJ)

Installatie Mac

https://git-scm.com/downloads

* Dus **niet** de GitHub Desktop Manager!

Als je de melding krijgt dat installatie niet lukt omdat Xcode niet aanwezig is, open je de App Store en download je **Xcode**. Daarna kun je opnieuw proberen om Git te installeren.

GIT hernoemt sinds een tijdje alle **master** branches naar **main**. Bij het installeren kan het zijn dat gevraagd wordt of dit automatisch mag. Je mag kiezen voor “Let Git decide!”

Testen of het werkt

Start één van je IDE's en open de terminal. Voer daar het volgende commando in:

```
git help
```

Als je informatie te zien krijgt, is het gelukt! 🙌

Staat er *"Git is not recognised as an internal command"* is het niet goed gegaan. Probeer de installatie dan opnieuw!

Globals instellen

GIT wil graag weten met welk GitHub account wij werken. Log in op GitHub en check wat jouw **gebruikersnaam** en **email** zijn.

Daarna open je jouw IDE en voer je de volgende commando's één voor één in in jouw terminal:

```
git config ––global user.name “pietertje”

git config ––global user.email “pieter.test@gmail.com”
```

Globals instellen

Check of het gelukt is met het volgende commando:

```
git config ––list
```

Als het goed is zie je nu jouw email en gebruikersnaam staan!

Eigen project

- Een eigen project maken
- Git opzetten en de files *ignoren*
- Commits maken

Git in een nieuw project

Wanneer je zelf een nieuw project opzet, initialiseer je git met:

```
git init
```

Maak een nieuw bestand en noem deze `.gitignore`. Hierin zet plaats je de verwijzing naar de *.idea*-map:

```
/.idea
```

Maak nu een aantal bestanden aan om mee te kunnen oefenen, zoals bijvoorbeeld een HTML- en CSS-bestand.

Commit maken

Maak een aantal wijzigingen in jouw bestanden. Typ dan in de terminal:

```
git status
```

Je ziet de bestanden die je hebt aangepast (**rood**). Dan ga je ze **stagen**:

```
git add .
```

Check of ze *gestaged* zijn (**groen**) zijn:

```
git status
```

Commit deze bestanden met:

```
git commit –m “Bedenk een bericht”
```

Checken of het gelukt is

Je kunt checken of het committen gelukt is door het commando:

```
git log
```

Dan krijg je alle gemaakte commits te zien. Drukt telkens op “enter” om verder naar beneden te gaan. Ben je bij het eind en wil je het afsluiten? Dan druk je:

```
:q
```

Naar remote

- Een remote repository maken
- Token instellen **(eenmalig)**
- Locale commits pushen naar remote

Global token instellen

Wanneer je de token eenmalig correct ingesteld hebt wordt er (als het goed is) niet meer om jouw wachtwoord gevraagd!

Met het volgende commando vragen we jouw machine de eerstvolgende ingevoerde credentials op te slaan:

```
git config ––global credential.helper cache
```

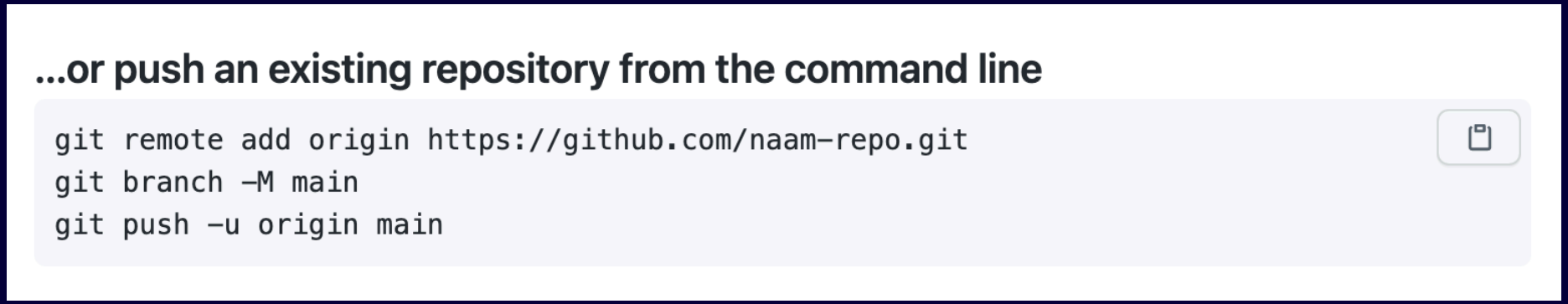
Voer daarna de stappen op de volgende slide uit. Wanneer je pusht, zal de terminal jou om inloggegevens vragen. Soms opent er een extra scherm.

Voer dan je **gebruikersnaam** in en jouw **token** als wachtwoord. Vanaf nu staan deze standaard opgeslagen op je computer!

Remote repo koppelen + pushen

Ga naar GitHub en maak daar een nieuwe repository aan. Je krijgt dan een scherm met meerdere soorten instructies te zien.

Koppel deze aan jouw oefenproject door de **2e set instructies** op GitHub één voor één te kopiëren en plakken in jouw terminal:



Huiswerk

- Huiswerkopdrachten (clonen en verwijderen van remote’s)

Huiswerk clonen

Wanneer je oefenopdrachten in de les maakt, gebruik je altijd een project met code **die al bestaat**. Die clone je dan naar jouw locale machine door via jouw IDE te kiezen voor *VCS > Get from version control > URL invullen*

Clone de volgende repository om te oefenen:
`https://github.com/hogeschoolnovi/git-workshop-fake-homework.git`

Daarna maak je een aantal veranderingen in het project en commit je deze lokaal:

```
git add .
git status
git commit –m “Super goede aanvulling”
```

Huiswerk pushen

Wanneer je oefenopdrachten in de les maakt, gebruik je altijd een project met code die al bestaat. Je wil jouw veranderen niet naar de originele repository pushen, maar naar een **eigen** repository.

> Maak een nieuwe repository aan op GitHub voor jouw nep-huiswerk.

Het project is nog wel gekoppeld aan de originele huiswerk repository. Voordat je de nieuwe remote kunt koppelen, moet je dus eerst **de oude remote verwijderen!**

```
git remote remove origin
```

> Daarna kun je jouw repository koppelen zoals op slide 18