

GitItDone

Slides voor GitItDone workshop van
het IT-lab



Wat is Git?

- Git is een version control system (VCS)
 - Hou een historie bij van je project en alle wijzigingen
 - Keer eenvoudig terug naar oudere versies (handig bij fouten...)
 - Leuk om evolutie en voortgang van project te zien

Een beetje geschiedenis...

- The man, the myth, the legend: Linus Torvalds...

Git installeren

- Beschikbaar voor Windows, MacOS en Linux (uiteraard)
 - Download voor jouw systeem de juiste [versie](#)
 - Windows: [Git Bash](#) is ook een optie

Git configureren

- Email en naam goedzetten
- verschil tussen globaal en lokaal configureren (context) - zeker voor geavanceerde gebruikers interessant

Git configureren - SSH sleutels

- Waarvoor dienen SSH-sleutels? (kort)
- Hoe genereer ik zo'n sleutelpaar?
- Public key toevoegen aan GitHub (demo?)
- Extra: je kan verschillende keypairs gebruiken voor verschillende accounts/repo's! (voorbeeld Emilia)

Git - commits

- Om veranderingen bij te houden, maken we gebruik van *commits*
- Commits bevatten één of meerdere wijzigingen in onze repository
- Commits hebben altijd een bijhorende *commit message*
- Commits hebben altijd een unieke *commit hash*

Commits - tips en best practices

- Probeer je commits *atomair* te houden! (commit heeft een doel)
- Schrijf duidelijke commit messages!
- Kom tot overeenkomst/structuur met jouw team (samenwerken)

Git - wat is een .gitignore bestand?

- Soms wil je sommige zaken niet bijhouden in jouw version control (denk aan wachtwoorden, sensitieve data,...)
- .gitignore definieert wat je niet wilt tracken (voorbeeld met demo!)
- Er bestaan heel wat templates voor verschillende projecten!

Git workflow

Git - command line versus GUI

- Git kan je zowel via de CLI als GUI gebruiken
- CLI: snel en efficiënt, iets hogere learning curve (maar niet veel!)
- GUI: handig voor een visueel overzicht, maar er gebeurt veel *under the hood*

Basiscommando's - lokaal

- Een repository lokaal binnenhalen (klonen)

```
git clone <URL>
```

- Wijzigingen in working directory aan staging toevoegen

```
git add <FILE>
```

- Alles toevoegen aan staging

```
git add .
```

Basiscommando's - lokaal

- Wijzigingen in stage aan de lokale git-repository toevoegen

```
git commit
```

- Meteen een (duidelijke!) commit message toevoegen

```
git commit -m "DIT IS MIJN COMMIT MESSAGE"
```

Basiscommando's - lokaal

- Lokale git-repository naar working directory

```
git merge
```

Basiscommando's - remote

- Lokale repository naar een remote repository

```
git push
```

- Remote repository naar lokale repository

```
git fetch
```

Belangrijk! Je moet een remote repository ingesteld hebben

Basiscommando's - remote

- Checken of er een remote is ingesteld

```
git remote
```

- Remote repository naar lokale werkomgeving

```
git pull
```

Pull = fetch + merge!!

Order matters...

Demo basic workflow

- Klassieke git workflow

```
git add .
```

```
git commit -m "Fix myFunction function: supports multiple parameters now"
```

```
git pull
```

```
git push
```

Git - branches

- Ruimte voor tekst, uitleg en/of oubollige meme?

Branches - commando's

- Nieuwe branch aanmaken

```
git branch <NAAMBRANCH>
```

- Wisselen naar branch

```
git checkout <NAAMBRANCH>
```

OF

```
git switch <NAAMBRANCH>
```

Git - merge conflicten

- Wat zijn merge conflicten?
- Wanneer ontstaan deze?
- Hoe kan je deze oplossen (BELANGRIJK - demo!!)

Demo merge conflict

Laat ons eens een merge conflict veroorzaken en ook oplossen...

Git - samenwerken

- Nood aan een remote repository (niet hetzelfde als lokaal)
- GitHub is een voorbeeld van zo'n provider/platform
- Verschill tussen Git en GitHub kort toelichten

GitHub specifiek

- Remote repositories, platform om samen te werken
- Heel wat specifieke mogelijkheden (Actions, Pages,...)
- Pull requests!!

Git - advanced topics

TO DO

- git rebase: in detail eens overgaan
- Meer advanced commando's?
- GitKraken of andere GUI tool demonsteren?
- Website hosten met GitHub pages -> kan gratis voor statische sites (zie huidige slides)
- DEMO's van vanalles moeten nog toegevoegd worden! (repo's, specifieke flows, use cases,...)

Handige/interessante links

- Git(Hub) guide van HOGENT - <https://hogenttin.github.io/git-hogent-gids/>
- Leren branchen - <https://learngitbranching.js.org/>
- Templates .gitignore - <https://github.com/github/gitignore>
- Wat als het allemaal in de soep loopt? - <https://ohshitgit.com/>