

Git for Developers

Oefening 1: Anatomy of a Repository

In deze oefening installeren we Git en maken een repository aan.

1. Installeer Git
2. Maak een repository aan
 - Open **Git Bash**
 - cd c:
 - mkdir git
 - cd git
 - git init demo
 - cd demo
3. Pas de configuratie-instellingen toe
 - git config --global user.email "..."
 - git config --global user.name "..."

Clone de tool voor het visualiseren van repositories:

1. cd ..
2. git clone <https://github.com/riezebosch/Glitter.git>
3. cd glitter
4. build.cmd

Sleep de demo repository op het canvas van de Glitter tool.

Oefening 2: Basic Snapshotting

In deze oefening voegen we wijzigingen toe aan de repository.

1. Maak een nieuw tekstbestand aan
2. Bekijk de status (status)
3. Voeg deze toe aan de repository (add)
4. Bekijk de status (status)
5. Commit de wijzigingen (het toegevoegde bestand) (commit)
6. Bekijk de status (status)
7. Wijzig de inhoud van het bestand
8. Bekijk de status (status)
9. Commit de wijzigingen
10. Wat is er gebeurd? Wat had je verwacht?
11. Voeg het gewijzigde bestand toe (add)
12. Bekijk de status (status)
13. Commit de wijzigingen (commit)
14. Bekijk de status (status)
15. Maak nog een nieuw tekstbestand aan
16. Bekijk de status (status)
17. Commit de wijzigingen (commit)
18. Wat is er gebeurd? Wat had je verwacht?

Oefening 3: Going Distributed

In deze oefening maken we een clone van onze bestaande repository en gaan we commits uitwisselen tussen de nieuwe en de oorspronkelijke repository.

Clone de huidige repository

- `cd c:/git1`
- `git clone demo demo-clone`

Bekijk de clone

- `cd demo-clone`
- `git log --oneline`

Bekijk de remote tracking branches

- `git remote`
- `git remote -v`

Voeg een nieuw bestand toe (add)

Commit deze wijziging (commit)

Navigeer naar de oorspronkelijke repository

- `cd c:/git/demo`

Controleer of de nieuwe wijzigingen doorgevoerd zijn

- `git log --oneline`

Ga terug naar de gekloonde repository

- `cd c:/git/demo-clone`

Push de wijzigingen naar de originele repository (origin)

- `git push origin master`

Waarom mag dit niet?

Ga terug naar de originele repository

- `cd c:/git/demo`

Haal de wijziging binnen (pull)

Waarom kan dit niet?

Voeg de gekloonde repository als remote toe

- `git remote add new_remote c:/git/demo-clone`

Haal de wijzigingen van de gekloonde repository binnen

- `git pull new_remote master`

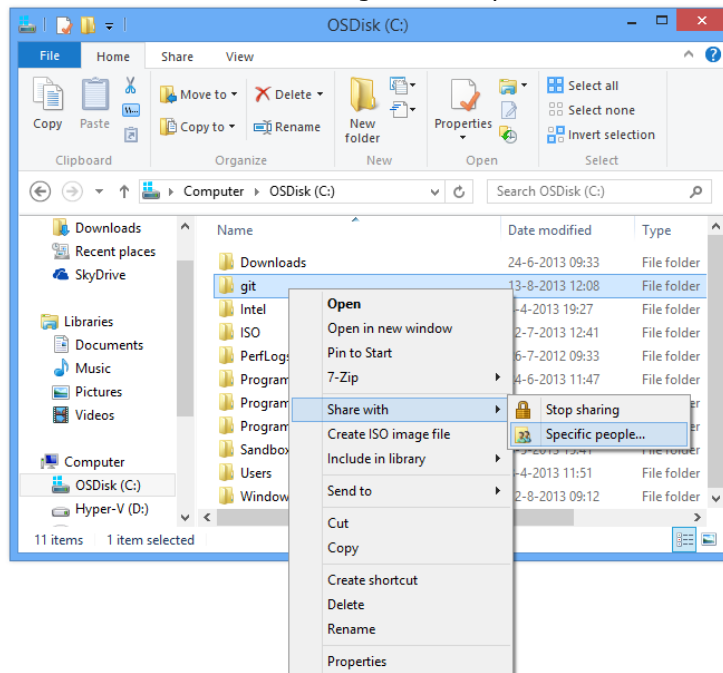
Controleer of het nieuwe bestand aanwezig is

- `git log --oneline`

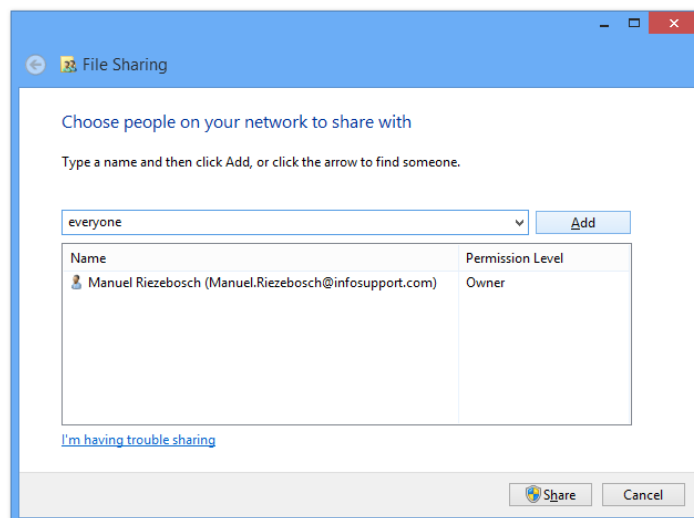
¹ **Let op!** Bash werkt met de forward slash als path separator.

We gaan nu een extra clone maken van onze repository op het netwerk

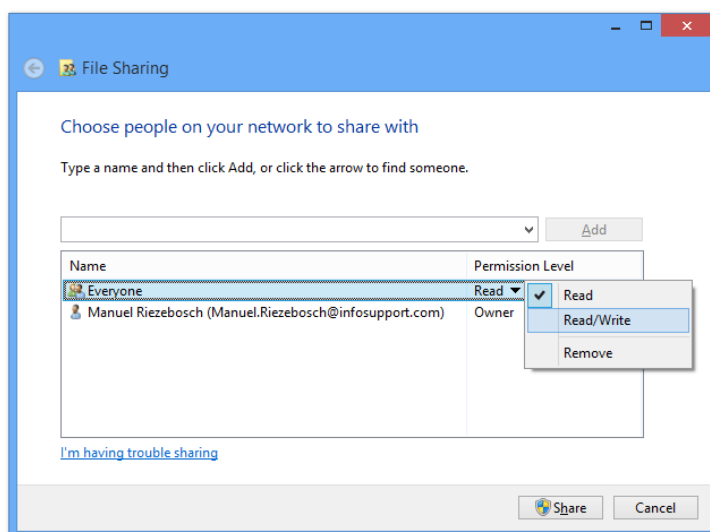
Maak een share aan voor de git directory:



Rechtermuisknop → Share with → Specific people...



Type in "everyone" en klik op "Add"



Everyone → Read → Read/Write

Maak nu een "bare" repository aan in de share van degene die naast je zit:

- `git init //cursistXY2/git/cursistXX3 --bare`

Voeg deze repository toe als remote aan jouw repository:

- `git remote add buurman //cursistXY/git/cursistXX`

Push de commits van master naar deze remote

- `git push buurman master`

Waarom kan en mag naar deze repository wel worden gepusht?

Oefening 4: Branching and Merging

1. Clone de bootstrap source code: <https://github.com/twbs/bootstrap.git> (clone)
2. Bekijk de carousel example in de browser (docs/examples/carousel)
3. Maak een branch aan met de naam 'background' (branch)
4. Checkout de nieuwe branch (checkout)⁴
5. Pas de achtergrondkleur aan in de carousel.css


```
body {
    padding-bottom: 40px;
    color: #5a5a5a;
    background-color: red;
}
```
6. Bekijk het resultaat
7. Commit de change.
8. Checkout master
9. Maak een nieuwe branch aan met de naam 'footer'
10. Checkout de nieuwe branch
11. Pas in de index.html de footer aan naar je eigen bedrijfsnaam (op de plaats van "2013 Company, Inc")
12. Bekijk het resultaat
13. Commit de change.

² cursistXY is de machinenaam van de PC van je buurman of buurvrouw

³ cursistXX is de machinenaam van je eigen PC

⁴ Het aanmaken en uitchecken van de branch kan ook in één keer met 'checkout -b'

14. Maak een nieuwe branch aan met de naam 'combined'
15. Merge de changes van de andere branch(es) naar deze branch
16. Bekijk het resultaat

Extra oefening:

De docent deelt een repository met daarin een leeg bestand waarin de namen en de e-mailadressen van de cursisten moeten komen. Iedereen krijgt leesrechten op deze repository en kan deze clonen. Slechts enkelen krijgen schrijfrechten.

De cursisten wisselen onderling commits uit door elkaar schrijfrechten op de repository te geven. Uiteindelijk pusht de cursist met schrijfrechten het resultaat terug naar de repository van de docent.

Oefening 5: Status and Diff

1. Ga terug naar de 'demo' repository in c:/demo
2. Pas een bestand in de working directory aan.
3. Voeg de wijzigingen toe aan de repository (add).
4. Wijzig hetzelfde bestand nogmaals
5. Vergelijk de
 - Workspace met index
 - Workspace met HEAD
 - Index met HEAD
6. Verwijder de bestanden uit de staging area (reset)
7. Maak de lokale wijzigingen ongedaan (checkout)
8. Ruim de working directory op (clean)

Oefening 6: Rewriting History

1. Voeg een nieuwe wijziging toe aan de laatste commit en pas de commit message aan (commit --amend)
2. Beschrijf wat er net is gebeurd in de repository. Wanneer mag dit nog en wanneer absoluut niet meer?
3. Voeg alle commits samen tot één commit (rebase)

Oefening 7: Moving Around

In deze oefening gaan we proberen de master branch van de vorige oefening te herstellen.

1. Ga op zoek naar de laatste commit voor de rebase pogingen
2. Doe een checkout van deze commit
3. Merk op dat je HEAD nu detached is
4. Maak een branch met de naam 'old' die wijst naar deze commit
5. Doe een checkout van de vorige versie
6. Controleer dat de bestand(en) in de working directory teruggezet zijn.
7. Controleer met de status