



✨ Forbind dit projekt til GitHub ✨

Lidt forskellige måder – by Tara



Den lette



Den overskuelige



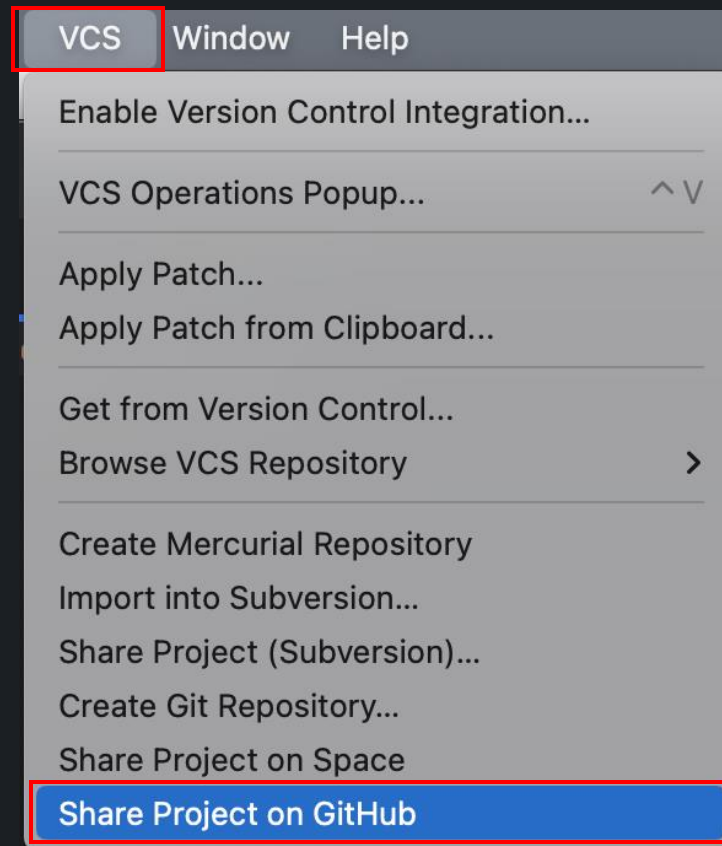
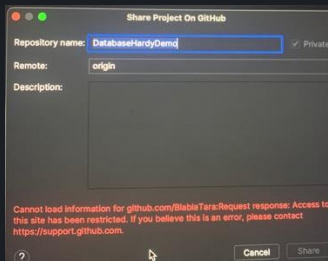
Den nørdede



# Den lette – step-by-step

- Åben dit projekt.
- Tryk på VCS øverst
  - hvis den hedder Git, har dit projekt registreret Git som dit aktive versionsstyringssystem
- Tryk på Share Project on GitHub

- **OBS:** Hvis fejl vises – Gå til slide 7





# Den lette – step-by-step

- Repository navn
- Tryk privat/public
- Evt. udfyld projekt beskrivelse
- Tryk Share

Share Project On GitHub

Repository name:  ☒ Private

Remote:

Description:

? Cancel Share

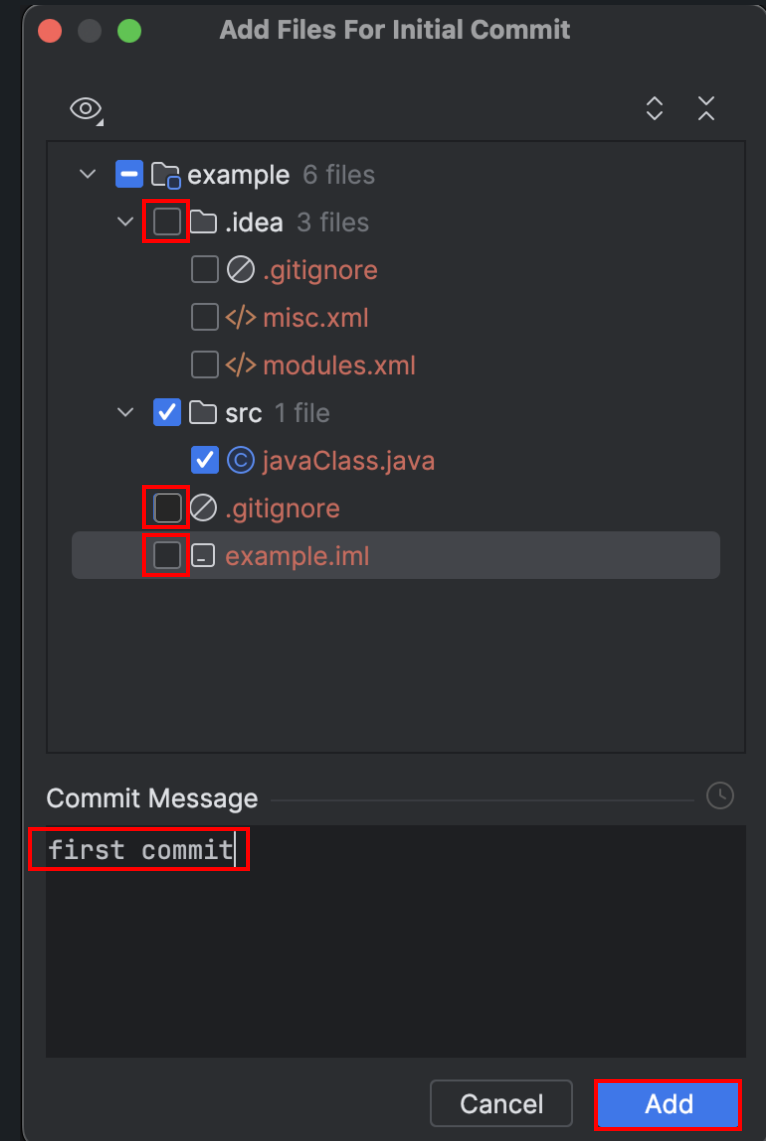


# Den lette – step-by-step

- Fravælg ”fylde-filer” – KUN til aflevering
- Skriv din Commit besked
- Tilføj dit projekt til det oprettede repository

• Juhuu! Succes 🥳

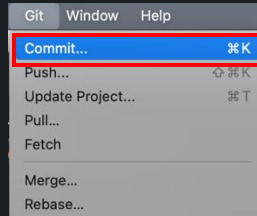
Successfully shared project on GitHub  
example



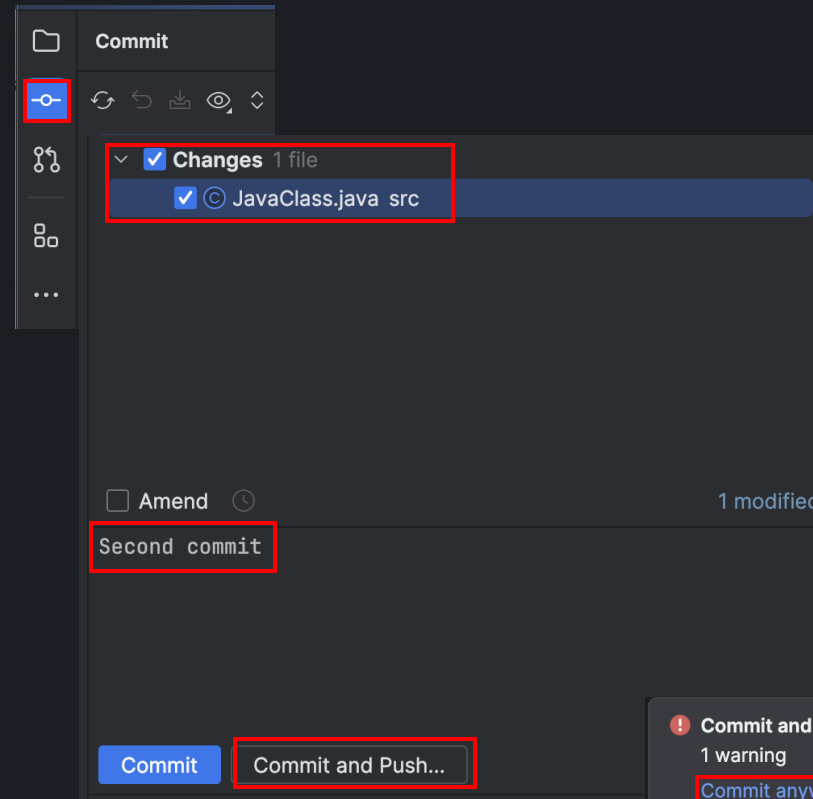


# Den lette – step-by-step

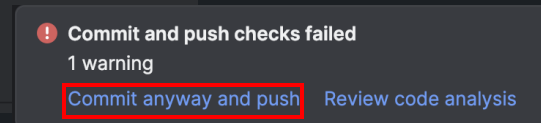
- Fra Git i toppen (gamle VCS)



- Eller... Fra siden.
  - Tryk commit and push anyway til WARNING



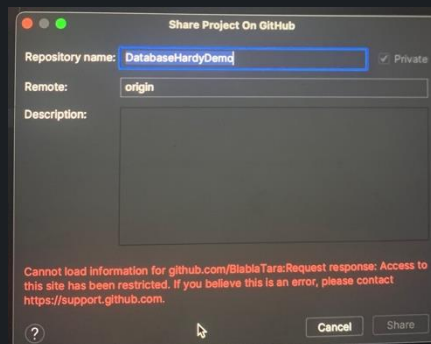
- Eller... Fra terminal. (bliver vist senere)



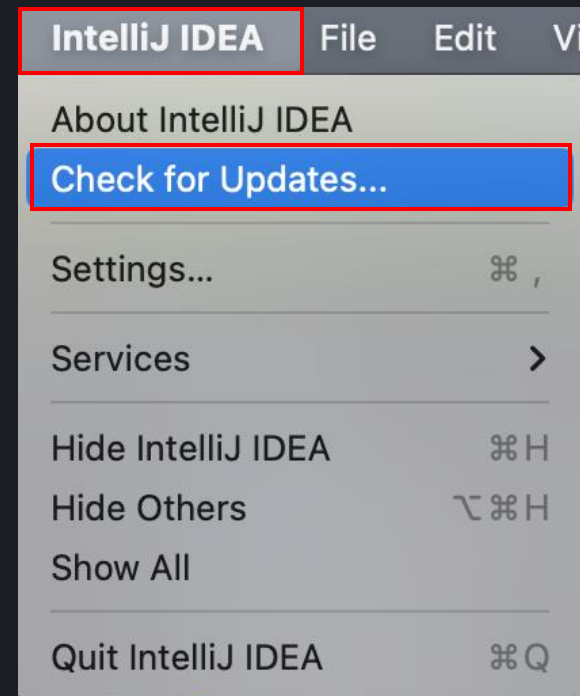


# Den lette – FEJL

- Hvis du får denne fejl



- Skal du opdatere din IntelliJ
  - Tager ca. 10 min.
  - Rimelig intuitiv – ja til alt (hvis jeg husker rigtigt)

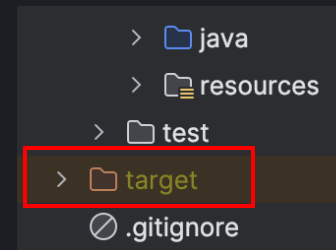
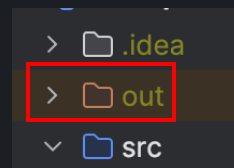




# Den overskuelige – step-by-step

- Opret dit projekt i IntelliJ **INDEN** du laver repo
- Lav main, og kørs en simpel kode
  - Java - Skaber out
  - SpringBoot - Skaber target
- Skabt .idea mappe to-do
  - Java – skal manuelt lægges i gitignore
  - SpringBoot – Lægges selv i gitignore

```
public class Hello {  
  
    public static void main(String[] args) {  
        System.out.println("Hej");  
    }  
}
```



gitignore i SpringBootser således ud:

```
### IntelliJ IDEA ###  
.  
*.iws  
*.iml  
*.ipr
```





# Den overskuelige – step-by-step

- Ind på Github
- Opret repository ( Dette er fra forsiden)
- Husk public eller privat
- Tryk Create

## Home

<> Start writing code

### Start a new repository for BlablaTara

A repository contains all of your project's files, revision history, and collaborator discussion.

#### Repository name \*

✔ Example is available.

☒ **Public**

Anyone on the internet can see this repository

☐ **Private**

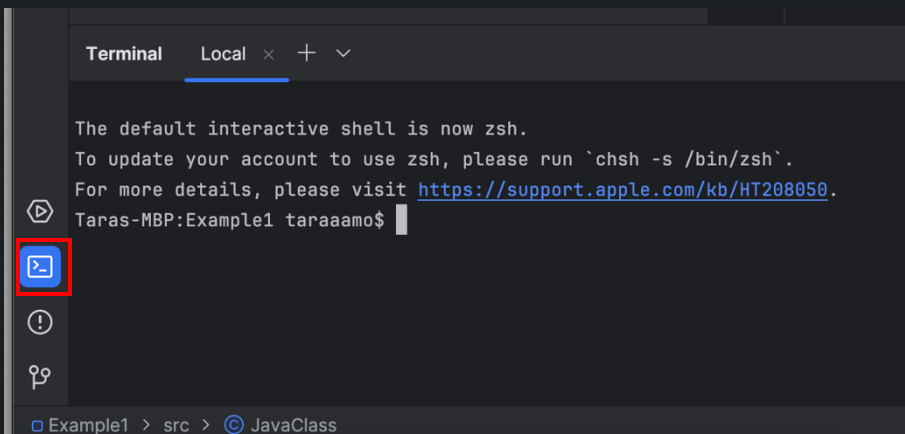
You choose who can see and commit to this repository

Create a new repository



# Den overskuelige – step-by-step

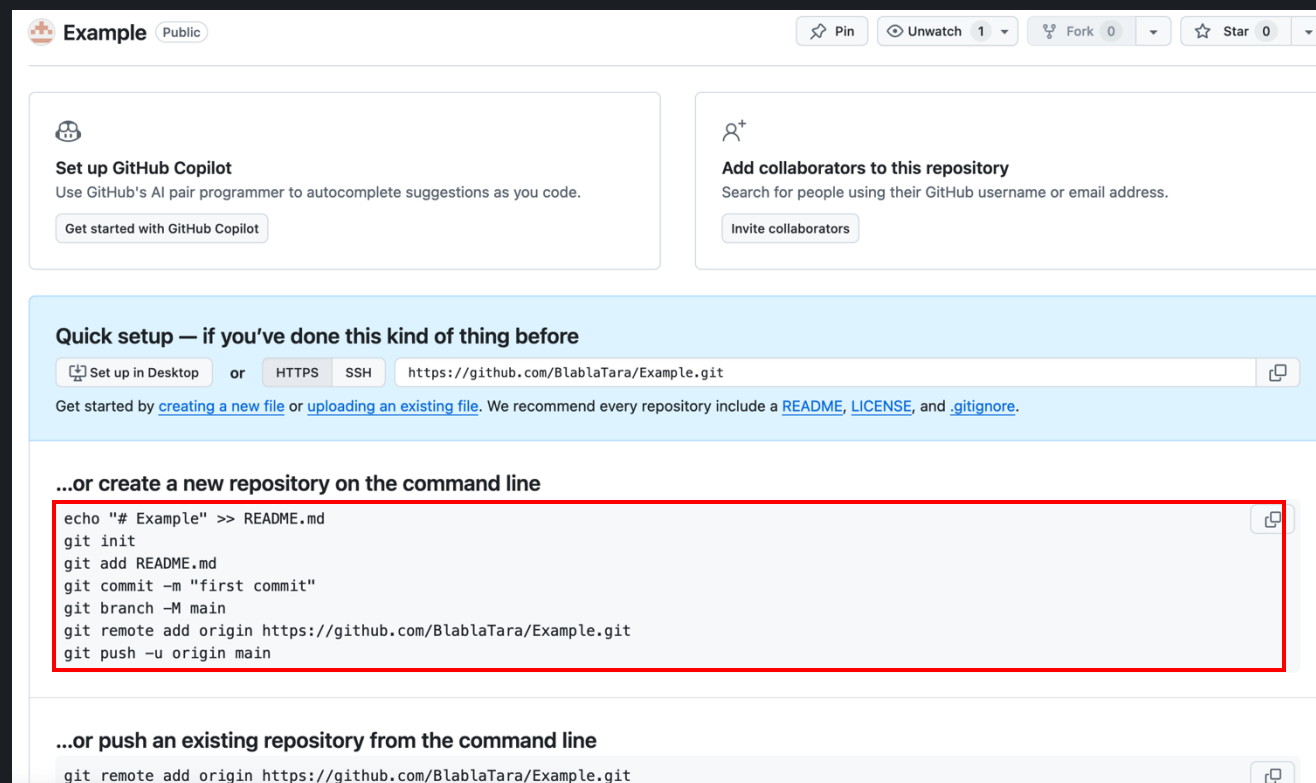
- En tom repository ser således ud:
- Denne kode skal skrives i Terminalen  
- (tilføj README eller ej)
- Det forgår i bunden af projektet



```
Terminal Local x + v

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Taras-MBP:Example1 taraaamo$
```

Example1 > src > JavaClass





# Den overskuelige – step-by-step

- Du kan med det samme opdatere dit repo og se om (og hvad) fra dit projekt der er lagt op
- **Åh åh:** (lavede repo og comittede du kørte dit program)
  - HVIS du i dit java-projekt får sendt idea og iml fil med til dit repo, kan du efterfølgende ligge dem i gitignore og så skrive disse komandoer i terminalen:

```
git rm --cached myproject.iml
```

```
git commit -m "besked"
```

```
git push (origin main)
```



# Den nørdede – step-by-step

- Opret projekt i IntelliJ. – Læg ting i gitignore.
- Åben terminal i IntelliJ og skriv:
- `git init`
- `git add .`
- `git commit -m "besked"`
- Opret repository på github
- Tilbage i terminalen i IntelliJ og skriv:
- `git remote add origin (github link)`
- `git push -u origin master`



# Git kommandoer:

- `git init`
  - Laver et git repo, hvor data kan blive gemt (.git)
- `git add .`
  - Tilføjer alle ændringer i den aktuelle mappe
- `git commit -m "besked"`
  - Laver et "snap-shot" af dit projekt. "gemmer"
- `git commit -a -m "besked"`
  - Med `-a` flaget, tilføjer du 'git add' i kommandoen
- `git remote add origin (github link)`
  - Forbinder din lokale git repo, til Githubs repo.
- `git push -u origin master`
  - Pusher & laver relationen ml. lokal branch og remote
- `git push`
  - Når relationen er skabt, kan du pushe med denne
- `git status`
  - Viser ændringer. Ingen ændringer – viser intet
- `git log`
  - Viser log historie – forskellige commits og aktiv branch
- `git checkout (commitID)`
  - Går til den givne commit.
- `git switch -`
  - Fortryder at være gået tilbage til den givne commit.