

Devin Vögele

Docker & Git

Abschlussarbeit

30.10.2024

Inhalt

Dokumentation des Abschlussprojektes	2
Klonen des Repositories	2
Einrichtung der Entwicklungsumgebung	2
Erstellen der README	2
Verwenden von Git (Commit, Push)	2
Erstellung und Nutzung von Docker Container	3

Dokumentation des Abschlussprojektes

Klonen des Repositories

Zunächst erstellte ich einen Fork aus dem Repository. Anschliessend habe ich danach die URL kopiert, und öffnete Git Bash. In Git Bash gab ich anschliessend den Path an, wo ich das Repository klonen wollte. Danach nutzte ich git clone "Die URL des Repositories", um es zu klonen.

```
devin@devin-MINGW64 ~  
$ cd C:/users/devin  
  
devin@devin-MINGW64 /c/users/devin  
$ git clone https://github.com/Devin-create/lighthouse  
Cloning into 'lighthouse'...  
remote: Enumerating objects: 93413, done.  
remote: Counting objects: 100% (101/101), done.  
remote: Compressing objects: 100% (99/99), done.  
error: RPC failed; curl 92 HTTP/2 stream 5 was not closed cleanly: CANCEL (err 8)  
error: 10 bytes of body are still expected  
fetch-pack: unexpected disconnect while reading sideband packet  
fatal: early EOF  
fatal: fetch-pack: invalid index-pack output  
  
devin@devin-MINGW64 /c/users/devin  
$
```

Beispiel für das Klonen von einem Repository

Einrichtung der Entwicklungsumgebung

Um den Auftrag zu lösen, müssen zuerst Bedingungen erfüllt werden:

Git sollte vorher installiert sein, zudem sollte auch die SSH-Verbindung aufgebaut sein. Auch Docker sollte installiert sein, und eingerichtet werden. Auch ein GitHub Konto sollte eingerichtet werden. Um den Prozess einwandfrei zu beenden.

Auch Vorkenntnisse mit Git sollten vorhanden sein.

Erstellen der README

In dem Repository hatte es am Anfang ein README.md. Dieses war jedoch falsch, weshalb ich in VS Code wechselte und den Kontext löschte. Anschliessend begann ich mit dem Erstellen des READMEs. Ich erstellte das README in 3 verschiedenen Sprachen, Deutsch, Englisch, und auch in Französisch.

Verwenden von Git (Commit, Push)

Um die lokal bearbeitete Repository auf GitHub zu "pushen", nutzte ich Git Bash. In dem Terminal von VS Code nutzte ich als erstes "git status", um zu prüfen, ob es noch unerledigte commits gibt. Nacher nutzte ich "git add ." um die Änderungen hinzuzufügen. Anschliessend nutzte ich git commit -m "individuelle Message, bspw. was geändert wurde", um den Commit zu speichern. Anschliessend nutzte ich git push, um die Änderungen auf mein GitHub-Repository zu pushen.

```
C:\Github\docker-nodejs-sample>git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
nothing to commit, working tree clean  
  
C:\Github\docker-nodejs-sample>git add .  
  
C:\Github\docker-nodejs-sample>git commit -m "Update README"  
On branch main  
Your branch is up to date with 'origin/main'.  
  
nothing to commit, working tree clean  
  
C:\Github\docker-nodejs-sample>git push  
Everything up-to-date  
  
C:\Github\docker-nodejs-sample>
```

Erstellung und Nutzung von Docker Container

Mit "docker init" konfigurierte ich mein Projekt. Ich musste dabei ein paar Fragen beantworten.

- What application platform does your project use?
- What version of Node do you want to use?
- Which package manager do you want to use?
- What command do you want to use to start the app ?
- What port does your server listen on?

Danach benutzte ich den Command «docker compose up --build», um die Applikation auf dem local:host3000 zu starten. Dies kann auch als Hintergrundprozess gemacht werden. Dies macht man mit «docker compose up --build -d».