

Dokumentation Abschlussarbeit

30.10.2024

Arion Kaleci

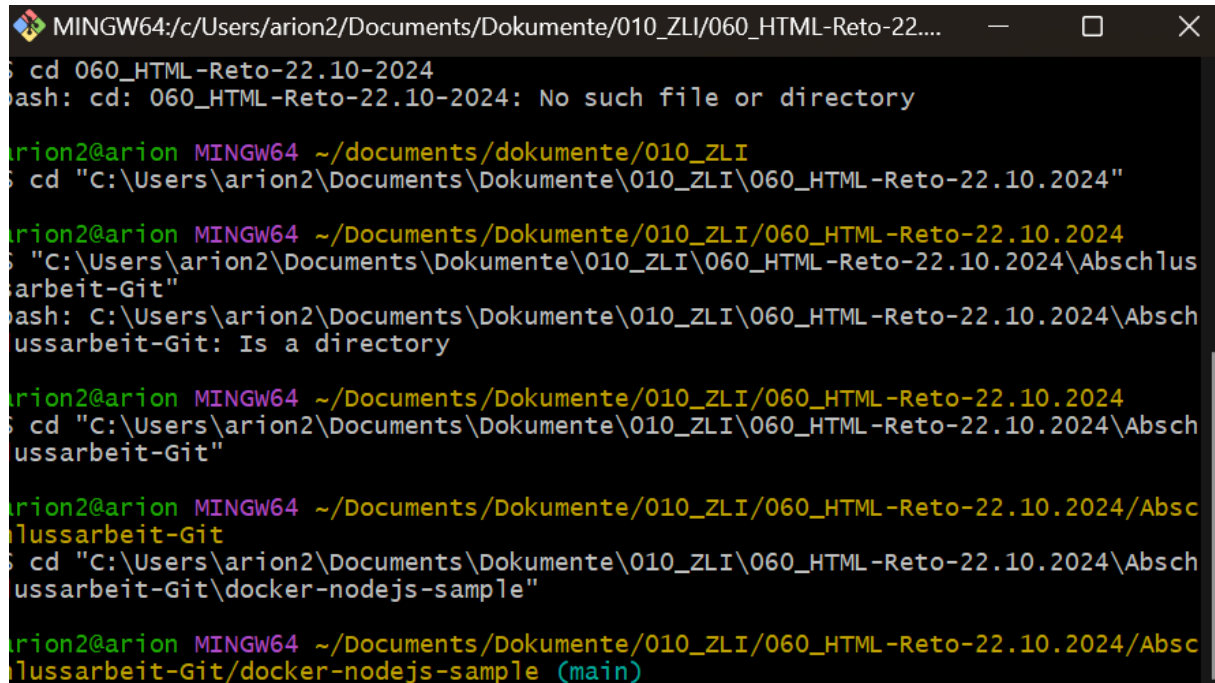


Inhalt

Dokumentation Abschlussarbeit	1
Klonen des Repository.....	3
Einrichtung der Entwicklungsumgebung	3
Erstellung der README.md.....	4
Verwendung von Git (Commit, Push)	4
Erstellung und Nutzung von Docker-Containern	5

Klonen des Repository

Bevor ich das geklont habe, habe ich zuerst den Fork erstellt. Dann habe ich Git Bash geöffnet und bin mit „cd“ in meinen Ordner „docker-nodejs-sample“ gegangen. Dann habe ich einfach "git clone" eingegeben und das war's.



```
MINGW64:/c/Users/arion2/Documents/Dokumente/010_ZLI/060_HTML-Reto-22....
$ cd 060_HTML-Reto-22.10-2024
bash: cd: 060_HTML-Reto-22.10-2024: No such file or directory

arion2@arion MINGW64 ~/documents/dokumente/010_ZLI
$ cd "C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024"

arion2@arion MINGW64 ~/Documents/Dokumente/010_ZLI/060_HTML-Reto-22.10.2024
$ "C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git"
bash: C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git: Is a directory

arion2@arion MINGW64 ~/Documents/Dokumente/010_ZLI/060_HTML-Reto-22.10.2024
$ cd "C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git"

arion2@arion MINGW64 ~/Documents/Dokumente/010_ZLI/060_HTML-Reto-22.10.2024/Abschlussarbeit-Git
$ cd "C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git\docker-nodejs-sample"

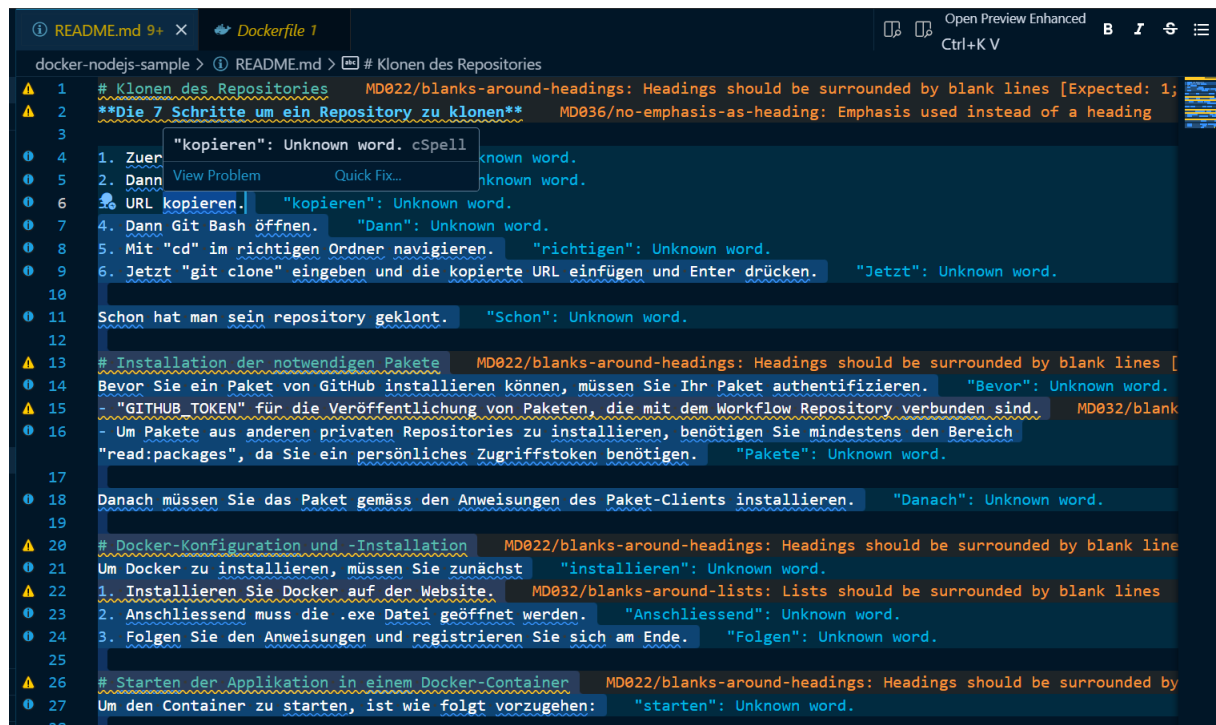
arion2@arion MINGW64 ~/Documents/Dokumente/010_ZLI/060_HTML-Reto-22.10.2024/Abschlussarbeit-Git/docker-nodejs-sample (main)
```

Einrichtung der Entwicklungsumgebung

Für die Installation habe ich einfach die Ordner angelegt, in denen ich alles haben wollte, und Docker installiert.

Erstellung der README.md

Da wir bereits eine README.md hatten, mussten wir nur den Inhalt löschen und mit Aufgabe 2 fortfahren.



```
docker-nodejs-sample > README.md > # Klonen des Repositories
1 # Klonen des Repositories MD022/blanks-around-headings: Headings should be surrounded by blank lines [Expected: 1;
2 **Die 7 Schritte um ein Repository zu klonen** MD036/no-emphasis-as-heading: Emphasis used instead of a heading
3
4 1. Zuerst die URL kopieren. "kopieren": Unknown word. cSpell
5 2. Dann die URL in Git Bash einfügen. "Dann": Unknown word.
6 3. URL kopieren. "kopieren": Unknown word.
7 4. Dann Git Bash öffnen. "Dann": Unknown word.
8 5. Mit "cd" im richtigen Ordner navigieren. "richtigen": Unknown word.
9 6. Jetzt "git clone" eingeben und die kopierte URL einfügen und Enter drücken. "Jetzt": Unknown word.
10
11 Schon hat man sein repository geklont. "Schon": Unknown word.
12
13 # Installation der notwendigen Pakete MD022/blanks-around-headings: Headings should be surrounded by blank lines [
14 Bevor Sie ein Paket von GitHub installieren können, müssen Sie Ihr Paket authentifizieren. "Bevor": Unknown word.
15 - "GITHUB TOKEN" für die Veröffentlichung von Paketen, die mit dem Workflow Repository verbunden sind. MD032/blank
16 - Um Pakete aus anderen privaten Repositories zu installieren, benötigen Sie mindestens den Bereich
17 "read:packages", da Sie ein persönliches Zugriffstoken benötigen. "Pakete": Unknown word.
18 Danach müssen Sie das Paket gemäss den Anweisungen des Paket-Clients installieren. "Danach": Unknown word.
19
20 # Docker-Konfiguration und -Installation MD022/blanks-around-headings: Headings should be surrounded by blank line
21 Um Docker zu installieren, müssen Sie zunächst "installieren": Unknown word.
22 1. Installieren Sie Docker auf der Website. MD032/blanks-around-lists: Lists should be surrounded by blank lines
23 2. Anschliessend muss die .exe Datei geöffnet werden. "Anschliessend": Unknown word.
24 3. Folgen Sie den Anweisungen und registrieren Sie sich am Ende. "Folgen": Unknown word.
25
26 # Starten der Applikation in einem Docker-Container MD022/blanks-around-headings: Headings should be surrounded by
27 Um den Container zu starten, ist wie folgt vorzugehen: "starten": Unknown word.
28
```

Verwendung von Git (Commit, Push)

Ich habe Git am Anfang und am Ende benutzt. Zuerst musste ich es mit Git Bash klonen, um es bearbeiten zu können. Danach brauchte ich es nicht mehr, da ich nur noch die README.md editieren und die Docker-Aufgabe erledigen musste. Zum Schluss musste ich noch «git add .», «git commit» und «git push» machen, damit alles auf Github ist.

```
PS C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git\docker-nodejs-sample> git add .
warning: in the working copy of '.dockerignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Dockerfiletemp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'README.Docker.md', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git\docker-nodejs-sample> git commit -m "README.md hochladen"
[main 2c19b76] README.md hochladen
5 files changed, 158 insertions(+), 41 deletions(-)
create mode 100644 .dockerignore
create mode 100644 Dockerfile
create mode 100644 Dockerfiletemp
create mode 100644 README.Docker.md
PS C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git\docker-nodejs-sample> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
```

Erstellung und Nutzung von Docker-Containern

Um einen Docker Container zu erstellen, öffnet man ein Terminal wie z.B. cmd. Dort muss man zuerst dock init eingeben. Dort mussten wir dann angeben, wie der Container aussehen soll. Zum Beispiel mussten wir den Port 3000 eingeben und auch «docker compose up -build». Eine wichtige Sache, die ich gelernt habe, ist, dass ein Account den Docker stören kann. Ich weiß das, weil es funktionierte, nachdem ich mich ausgeloggt hatte. Wenn ich mich eingeloggt habe, kam immer ein Fehler und ich konnte nicht bauen.

```
C:\Users\arion2\Documents\Dokumente\010_ZLI\060_HTML-Reto-22.10.2024\Abschlussarbeit-Git\docker-nodejs-sample>docker compose up -
-build
[+] Building 1.7s (3/3) FINISHED                                docker:desktop-linux
=> [server internal] load build definition from Dockerfile      0.0s
=> => transferring dockerfile: 1.37kB                            0.0s
=> ERROR [server] resolve image config for docker-image://docker.io/docker/dockerfile:1 1.6s
=> [server auth] docker/dockerfile:pull token for registry-1.docker.io 0.0s
-----
> [server] resolve image config for docker-image://docker.io/docker/dockerfile:1:
-----
failed to solve: failed to resolve source metadata for docker.io/docker/dockerfile:1: failed to authorize: failed to fetch oauth token: unexpected statu
s from GET request to https://auth.docker.io/token?scope=repository%3Adocker%2Fdockerfile%3Apull&service=registry.docker.io: 401 Unauthorized
```

Abbildung 1, Fehler, da ich angemeldet war.

```
=> [server internal] load .dockerignore                          0.0s
=> => transferring context: 663B                                  0.0s
=> [server stage-0 1/5] FROM docker.io/library/node:18.0.0-alpine@sha256:469ee26d9e00547ea91202a34ff2542f984c2c60a2edbb40 4.7s
=> => resolve docker.io/library/node:18.0.0-alpine@sha256:469ee26d9e00547ea91202a34ff2542f984c2c60a2edbb4007558ccb76b56df 0.0s
=> => sha256:759c4cf4639ec0cc2311e8f83065d24506c9c389c98be50750fa2a70af844c64 449B / 449B 0.2s
=> => sha256:ab29cc3c22467e202fbd625cb34d96af7579d7e146c3df4f9fa394dd97fea079 2.35MB / 2.35MB 0.4s
=> => sha256:4fababc43238082e7fe01f475aeddde0d59ede614ee7bb4a1c27a2e9a86927be 46.66MB / 46.66MB 3.3s
=> => sha256:df9b9388f04ad6279a7410b85cedfdbc2208c0a003da7ab5613af71079148139 2.81MB / 2.81MB 0.9s
=> => extracting sha256:df9b9388f04ad6279a7410b85cedfdbc2208c0a003da7ab5613af71079148139 0.1s
=> => extracting sha256:4fababc43238082e7fe01f475aeddde0d59ede614ee7bb4a1c27a2e9a86927be 1.2s
=> => extracting sha256:ab29cc3c22467e202fbd625cb34d96af7579d7e146c3df4f9fa394dd97fea079 0.1s
=> => extracting sha256:759c4cf4639ec0cc2311e8f83065d24506c9c389c98be50750fa2a70af844c64 0.0s
=> [server internal] load build context                          0.3s
=> => transferring context: 4.87MB                                0.2s
=> [server stage-0 2/5] RUN apk add --no-cache python3 make g++ 7.9s
=> [server stage-0 3/5] WORKDIR /usr/src/app                    0.1s
=> [server stage-0 4/5] RUN --mount=type=bind,source=package.json,target=package.json --mount=type=bind,source=packa 94.4s
=> [server stage-0 5/5] COPY . .                                0.1s
=> [server] exporting to image                                  11.5s
=> => exporting layers                                           9.1s
=> => exporting manifest sha256:fcda42b5ee1f610d571347dfb82fe8e2e250f30633ebba591ffa4c9c52be4eb5 0.0s
=> => exporting config sha256:7fc88b524a5a159c0c229767d120cff3e3555160f3b7b2228d07da6946cef014 0.0s
=> => exporting attestation manifest sha256:f17c84f2cd0b5fe40b83b06ed48eaf63e3484e42db75455d6358f10394ce485b 0.0s
=> => exporting manifest list sha256:25fecel1fd04acdda7db8311f41306fd76b9c1cfc073994e5c71c364c8dc5752 0.0s
=> => naming to docker.io/library/docker-nodejs-sample-server:latest 0.0s
=> => unpacking to docker.io/library/docker-nodejs-sample-server:latest 2.3s
=> [server] resolving provenance for metadata file              0.0s
[+] Running 2/2
✔ Network docker-nodejs-sample_default Created                0.0s
✔ Container docker-nodejs-sample-server-1 Created              0.3s
Attaching to server-1
server-1 | Using sqlite database at /tmp/todo.db
server-1 | Listening on port 3000
```

Abbildung 2, Erstellung, ohne Anmeldung

Das Ergebnis, eine To-Do-Liste.

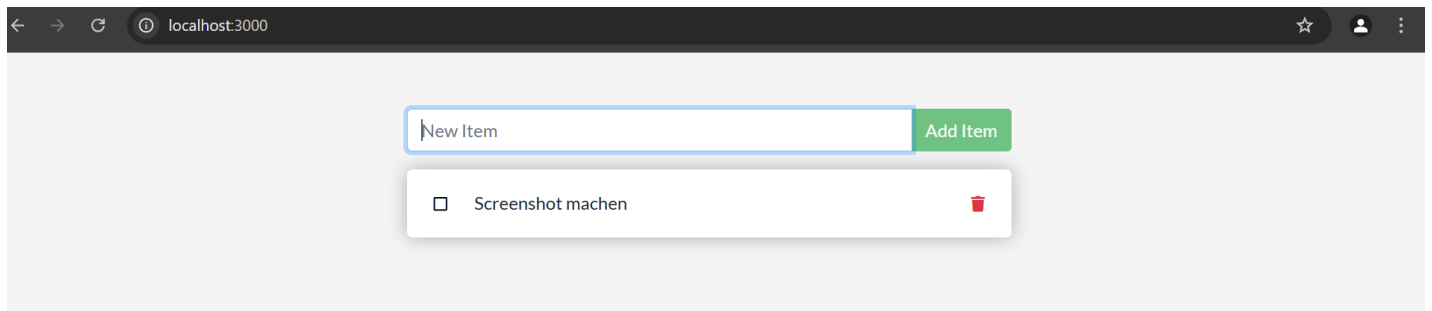


Abbildung 3, Ergebnis