Dokumentation

**Abschlussaufgabe Development Fundamentals**

Rico Krenn

Inhaltsverzeichniss

[Dokumentation 2](#_Toc212035817)

[Klonen des Repositories 2](#_Toc212035818)

[Einrichtung der Entwicklungsumgebung 2](#_Toc212035819)

[Erstellung der README.md 2](#_Toc212035820)

[Verwendung von Git (Commit, Push) 2](#_Toc212035821)

[Erstellung und Nutzung von Docker-Containern 3](#_Toc212035822)

# Dokumentation

## Klonen des Repositories

A screenshot of a computer

AI-generated content may be incorrect.Ich habe eine Fork des folgenden GitHub-Repositories erstellt:[docker-nodejs-sample](https://github.com/ICT-BLJ/docker-nodejs-sample). Dies habe ich gemacht indem ich oben rechts im Repo auf *Fork* und *Create a new Fork* gegangen bin. Wie es im Bild erichtlich ist:

## Einrichtung der Entwicklungsumgebung

Die Anwendung habe ich von GitHub geforkt und geclont. Das Deployment des Docker-Containers ist unter “Erstellung und Nutzung von Docker-Containern” dokumentiert.

Um effizient zu arbeiten habe ich mir meine Entwicklungsumgebung gut eingerichtet. Als IDE habe ich Visual Studio Code gewählt, da ich mich darin am bessten auskenne und es in vielen anwendungsfällen gut funktioniert. Um den Docker-Container zu erstelle habe ich Docker Desktop installiert, da es eine gute möglichkeit bietet.

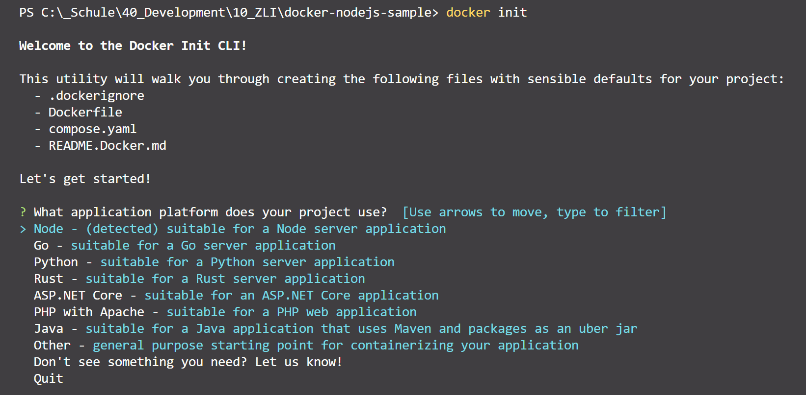
## Erstellung der README.md

Um ein README.md zu erstellen, muss man in VS-Code ein neues File erstllen und es README.md benennen.

## Verwendung von Git (Commit, Push)

Ich habe Git verwedet, um meine Änderungen zu commiten und zu tracken. Ich habe nach jeder eindeutigen änderung einen commit gemacht, damit mein Projekt übersichtlich bleibt. Git Push habe ich verwendet um es auf GitHub zu bekommen und um ich eine history zu erstellen.

## Erstellung und Nutzung von Docker-Containern

Ich bin der [Anleitung](https://docs.docker.com/guides/nodejs/containerize/) von Docker um Node.js-Anwendung zu containerisieren gefolgt. Der erst Punkt ist, dass man im directory “docker-nodejs-sample” ist, damit die assets beim richtigen Ort ertellte werden. Der erste Befehl ist “docker init”. Er stellt dir Fragen, die man beantworten muss:

What application platform does your project use? **Node**

What version of Node do you want to use? **18.0.0**

Which package manager do you want to use? **npm**

What command do you want to use to start the app: **node src/index.js**

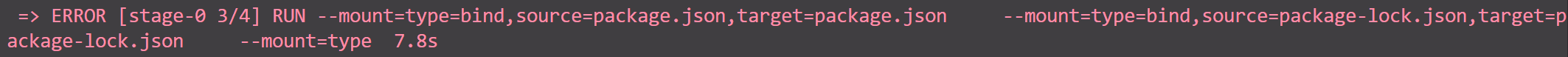
What port does your server listen on? **3000**

Wenn man die Fragen richtig beantwortet hat, wurden im Projekt neue Dateien erstellt. Hier sind die Dateien die man jetzt mindestens haben muss: A screen shot of a computer program

AI-generated content may be incorrect.

Jetzt habe ich alle Dateien die ich benötige um die Applikation zum laufen zu bringen. Jetzt fehlt nur noch eins und zwar das Starte. Dies macht man mit diesem Kommand: “docker compose up --build”.

Beim Bauen ist bei mir folgender Fehler aufgeteten:



Ich habe zuerst versucht im Internet nach hilfreichen lösungen zu suchen. Da ich mich mit dem Theam jedoch nicht wirklich auskenne, hat mir dies nicht weitergeholfen. Deswegen habe ich ChatGPT den ganzen verlauf zukommen lassen, damit er es mir erklärt.

Chat hat gesagt, dass es aufgrund von zwei dingen nicht funktionier hat. Erstens schlägt der Docker-Build fehl, da sqlite3 nativen Code kompilieren muss und zweitens, dass Image node:18-alpine nicht über die erforderlichen Build-Tools (Python, make, g++ usw.) verfügt.

Da ich jetzt die Fehler kenne, habe ich mich daran gemacht, die Fehler auszumerzen.

Um Python und die Build-Tools zu installieren musst ich im Dockerfile folgende Linie einsetztn: “RUN apk add --no-cache python3 make g++”.

Um den Docker zum laufen zu bringen habe ich dann “docker compose build --no-cache” ins VS-Code Termainal eingegeben. Dadurch werden die Docker images aufgebaut. Es sieht sich das Dockerfile an und befolgt die Anleitung Schritt für Schritt. Das “--no-cache” steht dafür, dass ein sauberen wiederaufbau gemacht wird, wenn man etwas verändert hat, man einen error von einem alten cache layer hat oder man sicher gehen will, dass man komplet frisch reinstartet. Ohne diesen Part, würde es schneller starten. Es skipt aber vieleicht Teile, die nicht geändert wurden. Dies könnte dazu führen, dass es einen unsauberen Start gibt.

Jetzt habe ich die Docker images aufgestzt. Um den Container nun vom compose file zu Starten, habe ich diesen Befehl eingegeben: “docker compose up”.