Todo-Applikation

Von Mattia Tuor

Ein Bild, das Screenshot, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

Inhaltsverzeichnis

[Klonen des Repositories 2](#_Toc212048264)

[Einrichtung der Entwicklungsumgebung 3](#_Toc212048265)

[Erstellung der README.md 4](#_Toc212048266)

[Verwendung von Git (Commit, Push) 5](#_Toc212048267)

[Erstellung und Nutzung von Docker-Containern 6](#_Toc212048268)

[Docker-Konfiguration und -Installation 6](#_Toc212048269)

[Starten der Applikation in einem Docker-Container 6](#_Toc212048270)

[Anwendung gestartet 6](#_Toc212048271)

[Anwendung im Hintergrund gestartet 6](#_Toc212048272)

# Klonen des Repositories

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.Über den Link von Moodle bin ich auf das Repository von Reto gekommen. Dort habe ich auf "Fork" geklickt, um eine Kopie des Repositories zu erstellen. Nachdem ich die Kopie erstellt hatte, habe ich auf "Code" geklickt und den Befehl unter HTTPS kopiert.

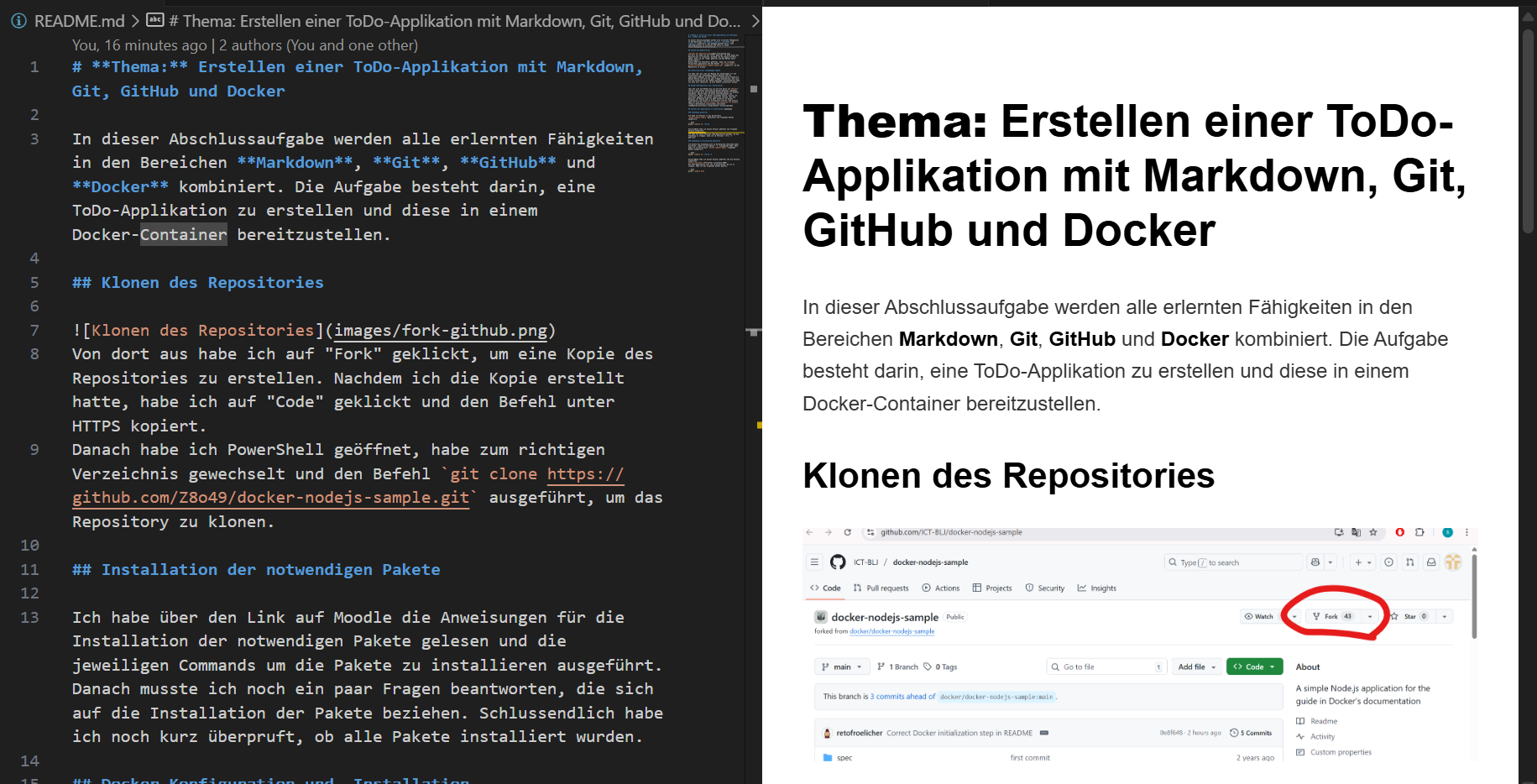
Danach habe ich PowerShell geöffnet, habe zum richtigen Verzeichnis gewechselt und dort den Befehl `git clone https://github.com/Z8o49/docker-nodejs-sample.git` ausgeführt, um das Repository zu klonen.

# Einrichtung der Entwicklungsumgebung

Ich habe mit VS Code gearbeitet. Um VS Code für das Projekt einzurichten habe ich, nach dem ich das Github Repository auf meinem Laptop hatte, die Extension für Docker heruntergeladen. Danach habe ich Docker selbst noch heruntergeladen, weil ich ohne Docker gar nicht an dem Projekt arbeiten könnte. Was auch wichtig ist für die Einrichtung, ist der Internetzugriff.

# Erstellung der README.md

Das README.md File war schon vorhanden weshalb ich dies nur noch anpassen musste. Dann habe ich mit Markdown alle Abschnitte so hinzugefügt wie es auf GitHub stand. Kleines Beispiel:



# Verwendung von Git (Commit, Push)

Ich habe Git so verwendet, dass ich immer commited habe, sobald ich mit einer Datei beziehungsweise mit einer Aufgabe fertig war. Ich habe auch probiert, einigermassen verständliche Git Messages zu verwenden.

# Erstellung und Nutzung von Docker-Containern

## Docker-Konfiguration und -Installation

Über den Link auf Moodle bin ich auf die Seite von [Docker](https://www.docker.com/products/docker-desktop/) gelangt und habe dort die passende Download-Option für Windows ausgewählt. Anschließend habe ich die Datei heruntergeladen und installiert. Bevor ich Docker verwenden konnte, musste ich WSL2 aktualisieren und meinen Laptop neu starten. Mit dem Beispiel von Moodle habe ich überprüft, ob Docker korrekt funktioniert, und konnte es erfolgreich starten. Zum Abschluss habe ich noch die [Visual Studio Code Extension für Docker](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker) heruntergeladen.

## Starten der Applikation in einem Docker-Container

### Anwendung gestartet

Ich habe im Terminal in das Verzeichnis docker-nodejs-sample gewechselt und den Befehl «docker compose up –build» ausgeführt. Anschließend habe ich meinen Browser geöffnet und die Adresse http://localhost:3000 aufgerufen. Ich konnte nun meine einfache ToDo-Applikation sehen. Um die Anwendung zu stoppen, habe ich im Terminal Strg + C gedrückt.

### Anwendung im Hintergrund gestartet

Ich konnte die Anwendung auch im Hintergrund (detached mode) ausführen, indem ich die Option -d hinzugefügt habe. Dazu habe ich im Verzeichnis docker-nodejs-sample den Befehl «docker compose up --build -d» ausgeführt. Anschließend habe ich meinen Browser geöffnet und erneut die Adresse http://localhost:3000 aufgerufen. Die ToDo-Applikation lief nun im Hintergrund. Um sie zu stoppen, habe ich den Befehl «docker compose down» verwendet.