ToDo-Applikation mit Docker

Inhalt

[GitHub Repository klonen 2](#_Toc177740510)

[Verwendung von Git 2](#_Toc177740511)

[Docker-Konfiguration und -Installation 2](#_Toc177740512)

[Einrichtung der Entwicklungsumgebung 2](#_Toc177740513)

[Erstellung und Nutzung von Docker-Containern 3](#_Toc177740514)

[Bearbeiten der README.md 3](#_Toc177740515)

# GitHub Repository klonen

Ich bin als Erstes auf [docker-nodejs-sample](https://github.com/ICT-BLJ/docker-nodejs-sample) und habe die eine Fork von dem Projekt gemacht. In meinem neu erstellten Repository habe ich dann unter "Code" den HTTPS-Link kopiert. Ich habe Visual Studio Code in einem neuen Ordner geöffnet und dann die Repository von GitHub auf mein lokales Gerät kopiert, indem ich in der Kommandozeile den Befehl "git clone [link]" eingegeben habe. Der Befehl klont das Repository auf mein Gerät, sodass ich es jetzt lokal bearbeiten kann.

# Verwendung von Git

Wenn Änderungen auf GitHub hochgeladen werden sollen, kann man in der Kommandozeile von der IDE (z.B. Visual Studio Code) "git commit -m [message]" und danach "git push" eingeben. Die Nachricht, die mit dem Commit mitgesendet wird, sollte simple sein, aber auch aussagekräftig sein. Neue Dateien die Lokal erstellt wurden, müssen noch mit "git add ." hinzugefügt werden.

# Docker-Konfiguration und -Installation

Da ich Docker bereits installiert hatte, musste ich es nicht mehr von [docker.com](https://www.docker.com/products/docker-desktop/) herunterladen. Mithilfe der [Docker-Dokumentation](https://docs.docker.com/guides/language/nodejs/containerize/#overview) habe ich das Programm dann zum Laufen gebracht.

# Einrichtung der Entwicklungsumgebung

Als Erstes habe ich Docker Desktop auf meinem Gerät gestartet. Man muss das Projekt, das man laufen lassen will, in Visual Studio Code öffnen und dann dort in die Kommandozeile "docker init" eingeben.

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

Automatisch generierte BeschreibungIn der Konsole muss man sich dann durch ein Konfiguration-Menü durcharbeiten, indem man die richtigen Einstellungen für sein Programm festlegt, wie zum Beispiel die Programmiersprache(Node.js, Python, PHP), deren Version und den Port, die sicherstellt, dass die Anwendung, die im Docker-Container läuft, mit deiner Aussenwelt (dem Host-System oder anderen Rechnern) kommunizieren kann.

Nachdem die Konfiguration abgeschlossen ist, werden Einstellungen gespeichert, die Projektstruktur erstellt, Abhängigkeiten installiert und möglicherweise Docker-Images gebaut.

# Erstellung und Nutzung von Docker-Containern

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte BeschreibungNachdem ich mit der Einrichtung von Docker Desktop fertig war, habe ich dann noch "docker compose up --build -d" in die Kommandozeile eingegeben. Das startet das Programm in dem erstellten Container von Docker Desktop.

Mit localhost:[port] kann man dann auf das Programm in einem beliebigen Browser zugreifen.

Im Browser sieht die Website dann so aus:

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

Automatisch generierte Beschreibung

# Bearbeiten der README.md

Ich habe dann die README.md Datei angefangen zu bearbeiten. Die README.md Datei wird auf GitHub unter dem Projekt angezeigt und dient als Dokumentation und als Anleitung zu dem Projekt. Die Datei wird in Markdown geschrieben, was keine Programmiersprache, sondern genauso wir HTML eine Auszeichnungssprache (Markdown language). In meiner README Datei habe ich eine Anleitung geschrieben, wie man das Programm selbst aufsetzen kann und auch mögliche Lösungswege dokumentiert.