Dokumentation: ToDo-Applikation

Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

Automatisch generierte BeschreibungIn dieser Dokumentation erkläre ich, wie ich eine ToDo-App gebaut und zum Laufen gebracht habe. Mit dieser App kann man Aufgaben erstellen, bearbeiten und löschen. Ziel war es, die App in einem Docker-Container laufen zu lassen und mit Git, GitHub und Markdown alles gut zu dokumentieren und zu teilen.

Bild: Meine ToDo-Applikation

**Inhaltsverzeichnis**

[1. Einführung 3](#_Toc181357812)

[2. Repository erstellen und klonen 3](#_Toc181357813)

[3. README.md schreiben 4](#_Toc181357814)

[4. Arbeiten mit Git 5](#_Toc181357815)

[5. Docker-Container erstellen und starten 5](#_Toc181357816)

[6. Fazit und Zusammenfassung 7](#_Toc181357817)

## 1. Einführung

Die ToDo-App ist eine kleine Webanwendung, die ich gebaut habe, um zu lernen, wie man eine App richtig dokumentiert und sie auf verschiedenen Systemen laufen lassen kann. Docker hilft dabei, die App auf jedem Computer gleich aussehen zu lassen, und mit Git und GitHub kann ich den Code einfach speichern und teilen.

Ziele dieses Projekts:

* **App in Node.js bauen**
* **Alle Schritte dokumentieren** in Markdown und Word
* **Git und GitHub nutzen** für die Versionskontrolle
* **App in Docker laufen lassen** – damit kann sie jeder auf seinem Computer testen

## 2. Repository erstellen und klonen

**Repository «forken»**

Als Erstes habe ich das Original-Repository docker-nodejs-sample auf GitHub "geforkt" (das bedeutet kopiert), damit ich eine eigene Version habe, an der ich arbeiten kann.

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

Automatisch generierte Beschreibung

Bild: Github (Screenshot) **Repository klonen**

Dann habe ich das geforkte Repository auf meinen PC geladen. Das nennt man "klonen". Das ging mit diesem Befehl:

|  |
| --- |
| git clone (hier kam die SSH URL des Repositorys)  cd docker-nodejs-sample |

So hatte ich jetzt die ganzen Dateien lokal und konnte an der App weiterarbeiten.

## 3. README.md schreiben

Die README.md-Datei ist wie eine Anleitung für das Projekt. Darin habe ich alle wichtigen Infos zusammengefasst, damit jeder weiß, wie er die App zum Laufen bringt. Die README.md enthält:

* **Klonen des Repositories**: Wie man das Projekt runterlädt.
* **Installation der Pakete**: Welche Befehle nötig sind, um die Abhängigkeiten zu installieren.
* **Docker einrichten**: Wie man die App in Docker laufen lässt.
* **App nutzen**: Eine kurze Erklärung, wie man Aufgaben hinzufügt, bearbeitet und löscht.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte BeschreibungBild: Meine README Datei

## 4. Arbeiten mit Git

Während des Projekts habe ich Git genutzt, um jede Änderung am Code festzuhalten und sie auf GitHub zu speichern. Dadurch konnte ich immer zu früheren Versionen zurückspringen, falls mal etwas schiefging. Hier ein paar der wichtigsten Git-Befehle, die ich verwendet habe:

* **Änderungen hinzufügen**:

|  |
| --- |
| git add . |

* **Commit erstellen** (damit speichert man die Änderungen mit einer kurzen Nachricht):

|  |
| --- |
| git commit -m "(Hier kommt die Beschreibung der Änderungen" |

* **Auf GitHub hochladen**:

|  |
| --- |
| git push origin main |

Mit diesen Befehlen hatte ich nach dem Ausführen immer die aktuelle Version auf GitHub.

## 5. Docker-Container erstellen und starten

Um die App mit Docker zum Laufen zu bringen, habe ich eine Dockerfile-Datei erstellt. Sie sagt Docker, wie es die App installieren und starten soll.

**Dockerfile erstellen**

Im Dockerfile habe ich festgelegt:

* welches Node.js-Bild Docker verwenden soll,
* dass die Projektdateien in den Container kopiert werden,
* Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

  Automatisch generierte Beschreibungund dass die App automatisch gestartet wird, wenn der Container läuft.

Bild: Mein Dockerfile

**Docker-Container erstellen und starten**

1. **Image bauen** – damit erstellt Docker eine Art "Bild" von der App:

|  |
| --- |
| docker build -t todo-app . |

1. **Container starten** – damit läuft die App und ist im Browser sichtbar:

|  |
| --- |
| docker run -p 3000:3000 todo-app |

Ein Bild, das Text, Elektronik, Screenshot, Display enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte BeschreibungJetzt kann ich http://localhost:3000 im Browser aufrufen und die App nutzen, um Aufgaben hinzuzufügen, zu bearbeiten oder zu löschen.

Bild: Git Bash (Screenshot)

## 6. Fazit und Zusammenfassung

Dieses Projekt hat mir gezeigt, wie wichtig es ist, alle Schritte ordentlich zu dokumentieren. Die README.md ist eine Art Anleitung, die jedem hilft, die App auf seinem Computer zum Laufen zu bringen. Außerdem habe ich viel über Docker gelernt – es macht es richtig einfach, die App auf verschiedenen Computern gleich laufen zu lassen.

Mit Git und GitHub konnte ich alle meine Änderungen festhalten und sicherstellen, dass ich immer eine aktuelle Version habe. Insgesamt war das Projekt eine coole Gelegenheit, die Basics der Entwicklung und Bereitstellung von Apps zu lernen.