**Abschlussaufgabe Development Fundamentals**

**Aaron Benesch**



Inhalt

[Einleitung 3](#_Toc181367576)

[Schritt 1 4](#_Toc181367577)

[Klonen des Repositories 4](#_Toc181367578)

[Schritt 2 5](#_Toc181367579)

[Erstellen der README-Datei in Markdown 5](#_Toc181367580)

[Schritt 3 6](#_Toc181367581)

[Dokumentation der Vorgehensweise 6](#_Toc181367582)

[Schritt 4 7](#_Toc181367583)

[Dockerize das Node.js-Projekt 7](#_Toc181367584)

[Schritt 5 8](#_Toc181367585)

[Git-Workflows 8](#_Toc181367586)

# Einleitung

Diese Woche haben wir uns verschiedene Themen wie Markdown, Git, GitHub und Docker beschäftigt. In dieser Abschlussaufgabe sollen alle gelernten Themen zusammengeführt werden, um eine ToDo-Applikation zu erstellen und diese in einem Docker-Container bereitzustellen. Dazu nutzen wir Git für die Versionskontrolle, schreiben eine Anleitung in Markdown und bereiten die Anwendung für Docker vor. Wir sollen die verschiedenen Tools in der Praxis anwenden können. Diese Dokumentation beschreibt Schritt für Schritt, wie die Aufgabe umgesetzt wird, und dient als Anleitung für einen klaren und gut strukturierten Arbeitsablauf.

# Schritt 1

## Klonen des Repositories

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

   Automatisch generierte Beschreibung**Fork erstellen:** Gehe zum Repository [docker-nodejs-sample](https://github.com/) auf GitHub und klicke oben rechts auf **Fork**, um eine Kopie des Projekts in deinem eigenen GitHub-Account zu erstellen.
2. **Repository klonen:** Öffne das Terminal (GitBash) und navigiere zu dem Verzeichnis, in dem du das Projekt speichern möchtest. Klone das geforkte Repository mit:

git clone <https://github.com/DeinBenutzername/docker-nodejs-sample.git>

Der Code muss natürlich personalisiert werden.

1. **In das Verzeichnis wechseln:** Gehe in das neu erstellte Projektverzeichnis:

cd docker-nodejs-sample

# Schritt 2

## Erstellen der README-Datei in Markdown

1. **README-Datei erstellen:**
   * Erstelle im Terminal eine neue Datei mit dem Namen README.md im Root-Verzeichnis des Projekts:

touch README.md

1. **README-Datei bearbeiten:**
   * Öffne README.md in Visual Studio Code (oder einem anderen Texteditor) und füge Informationen zum Projekt hinzu:
     + **Projekttitel** und **Beschreibung** der ToDo-Applikation.
     + **Installationsanleitung**, die beschreibt, wie man das Projekt klont und die Abhängigkeiten installiert:

## Installation

1. Repository klonen:

```bash

git clone https://github.com/DeinBenutzername/docker-nodejs-sample.git

**- In das Verzeichnis wechseln:**

cd docker-nodejs-sample

**- Pakete installieren:**

npm install

* **Docker-Setup** und **Start der Anwendung**.

1. **Speichern und Committen:**

**- Speichere die Datei, füge sie zur Git-Stage hinzu und committe sie:**

git add README.md

git commit -m "Add README with project description and installation instructions"

# Schritt 3

## Dokumentation der Vorgehensweise

**1. Neues Word-Dokument erstellen:**

- Erstelle ein neues Dokument und benenne es `Dokumentation\_DevelopmentFundamentals\_Vorname.docx`.

**2. Einleitung schreiben:**

- Eine kurze Einleitung schreiben, die sagt, für was das Projekt ist und was zum Projekt gehört

**3. Detaillierte Beschreibung jedes Schritts:**

- Für jeden Schritt eine genaue Beschreibung hinzufügen

- Eine klare Struktur benutzen, die Schritt für Schritt den Prozess erklärt und sodass es auch jeder anwenden kann

- Füge kurze Erklärungen für alle wichtigen Befehle und Dateien hinzu

**4. Screenshots einfügen:**

- Ergänze das Dokument mit Screenshots der wichtigsten Schritte und/oder Dinge die schwierig zu finden sind

- Die Screenshots sollten übersichtlich und in guter Qualität eingefügt sein

**5. Abschluss und Zusammenfassung:**

- Beende die Dokumentation mit einer kurzen Zusammenfassung, die die wichtigsten Punkte der Installation und des Docker-Einsatzes wiederholt.

**6. Speichern und Hochladen:**

- Speichere das Dokument als ` Dokumentation\_DevelopmentFundamentals\_Vorname.docx `.

- Füge die Datei zum Git-Repository hinzu und committe sie mit:

git add Dokumentation\_ToDo\_Applikation.docx

git commit -m "Add project documentation"

- Push die Datei auf GitHub:

git push origin main

# Schritt 4

## Dockerize das Node.js-Projekt

Öffne aus Visual Studio Code den Terminal ganz oben. Es soll unten aufgehen.

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

Automatisch generierte Beschreibung

Führe den Befehl «docker init» aus

Jetzt werden verschiedene Fragen in dieser Reihenfolge gestellt. Die Antworten sind blau.

---------------------------------------------------------------------------------------------------------------

What application platform does your project use? Node

What version of Node do you want to use? 21.0.0

Which package manager do you want to use? npm

What command do you want to use to start the app? node index.js

What port does your server listen on? 3000

---------------------------------------------------------------------------------------------------------------

Um die Anwendung zu starten, geben Sie docker compose up –build ein.

Wenn sie auf ihrem Browser <http://localhost:3000/> eingeben, soll ein einfache ToDo Liste erscheinen.

# Schritt 5

## Git-Workflows

Git-Workflows helfen dabei, den Fortschritt sauber und übersichtlich zu dokumentieren und regelmässig auf GitHub zu speichern. Hier sind die Schritte, um mit Git strukturiert zu arbeiten:

**1. Änderungen regelmässig committen**

Nach jeder wesentlichen Änderung im Projekt werden die Änderungen per Commit gespeichert. So bleibt der Projektverlauf nachvollziehbar und übersichtlich:

git add .

git commit -m "Beschreibung der Änderung"

**2. Änderungen auf GitHub pushen**

Lade deine Commits regelmässig auf GitHub hoch, um den aktuellen Stand des Projekts zu sichern und mit anderen zu teilen:

Git push origin main

Hier wird auf den „main“-Branch gepusht. Falls du auf einem anderen Branch arbeitest, gib den entsprechenden Namen an.

**3. Finalen Stand auf GitHub sichern**

Wenn die Arbeit an einem grösseren Abschnitt abgeschlossen ist, überprüfe den finalen Stand auf GitHub. Stelle sicher, dass alle Dateien vorhanden und aktuell sind und wichtige Änderungen im `main`-Branch zusammengeführt wurden.

# Zusammenfassung

In der Aufgabe wurde eine ToDo-Applikation erstellt, dokumentiert und in Docker containerisiert. Dazu wurden folgende Schritte durchgeführt: Erstellung eines GitHub-Forks, Anlegen einer README.md zur Installation und Konfiguration, Dokumentation der Arbeitsschritte, Docker-Containerisierung und Einsatz von Git-Workflows für regelmässige Commits und Updates auf GitHub. Die Kombination dieser Tools ermöglicht eine strukturierte, leicht zugängliche und plattformunabhängige Anwendung.