



Dokumentation zum Abschlussauftrag Development Fundamentals

Inhalt

[Klonen des Repositories 2](#_Toc177741189)

[Wiso 2](#_Toc177741190)

[Schritte zum Klonen des Repositories 2](#_Toc177741191)

[Schritt 1: Fork des Repositories erstellen 2](#_Toc177741192)

[Schritt 2: Repositorie Klonen 2](#_Toc177741193)

[Einrichtung der Entwicklungsumgebung 2](#_Toc177741194)

[Konfiguration nach der Installation von Docker 2](#_Toc177741195)

[1. Aktivierung des Windows-Subsystems für Linux (WSL) 2](#_Toc177741196)

[2. Aktivierung der Virtual Machine Platform 2](#_Toc177741197)

[3. Festlegen von WSL 2 als Standard 2](#_Toc177741198)

[Einrichtung der Entwicklungsungebung mit Docker 2](#_Toc177741199)

[Einführung 2](#_Toc177741200)

[Einrichtung 2](#_Toc177741201)

[Erstellung der README.md 2](#_Toc177741202)

[Vorbereitung 2](#_Toc177741203)

[Schreiben 2](#_Toc177741204)

[Verwendung von Git (Commit, Push) 2](#_Toc177741205)

[Einführung 2](#_Toc177741206)

[Verwendung 2](#_Toc177741207)

[Dateien zum Commit hinzufügen 2](#_Toc177741208)

[3.2. Änderungen hochladen (Push) 2](#_Toc177741209)

[Erstellung und Nutzung von Docker-Containern 2](#_Toc177741210)

[Einstellungen 2](#_Toc177741211)

[Erstellung 2](#_Toc177741212)

# Klonen des Repositories

## Wiso

Auf deinem Laptop kann man durch ein Reposetorieklon Entwicklungswerkzeuge wie Docker und Node.js konfigurieren und verwenden, um die Applikation zu entwickeln, zu testen und in einem Container bereitzustellen. Die Verbindung zum Fork auf GitHub stellt sicher, dass du deine Änderungen später durch Git commiten wieder in dein GitHub-Repository hochladen kannst.

## Schritte zum Klonen des Repositories

### Schritt 1: Fork des Repositories erstellen

Öffne GitHub und gehe zum ursprünglichen Repository, das geklont werden soll, z.B. das Beispielprojekt Docker-nodejs-sample.

Klicke auf den "**Fork**" Button oben rechts auf der Repository-Seite.

Dies erstellt eine Kopie des Repositorys in deinem eigenen GitHub-Konto, sodass du Änderungen vornehmen kannst, ohne das Original zu beeinflussen.

### Schritt 2: Repositorie Klonen

Um das Repositorie auf den Laptop zu Klonen muss man zu Beginn den SSH-Schlüssel auf GitHub finden und kopieren. Diesen Schlüssel findet man, wenn man in das Reposetorie geht und dort auf Code clickt.  
Anschliessend muss man diesen Schlüssel zusammen mit dem folgenden Befehl in die Git Bash eingeben.

**Achtung**: Schauen sie das in der Git Bash Konsole der gewünschte Speicherort ausgewählt ist.

**«git clone SSH-Schlüssel»**

**PS**: Falls sie noch keinen Schlüssel haben können sie sich einen unter Einstellungen in GitHub erstellen.

# Einrichtung der Entwicklungsumgebung

## Konfiguration nach der Installation von Docker

Nach der Installation von Docker sind noch einige optionale Konfigurationen notwendig, um das volle Potenzial von Docker in Verbindung mit dem Windows-Subsystem für Linux (WSL) auszuschöpfen.

### 1. Aktivierung des Windows-Subsystems für Linux (WSL)

Zunächst sollte das optionale Feature **Windows-Subsystem für Linux** aktiviert werden. Führen Sie dazu die folgende Anweisung in der **PowerShell** (mit Administratorrechten) aus:

**dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart**

### 2. Aktivierung der Virtual Machine Platform

Um WSL 2 zu verwenden, muss zusätzlich die Virtual Machine Platform aktiviert werden. Verwenden Sie hierzu den folgenden Befehl in der PowerShell:

**dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart**

### 3. Festlegen von WSL 2 als Standard

Um WSL 2 als Standardversion festzulegen, sollte der folgende Befehl ausgeführt werden:

**wsl --set-default-version 2**

## Einrichtung der Entwicklungsungebung mit Docker

### Einführung

Die Einrichtung der Entwicklungsumgebung wird mithilfe von **Docker** eingerichtet. Docker ermöglicht es, Applikationen in Containern auszuführen.

### Einrichtung

Alle schritte sollten im richtigen ordner durch Powershell oder Visual studio code ausgeführt werden.

Wechseln Sie in das Verzeichnis docker-nodejs-sample und führen Sie im Terminal den folgenden Befehl aus:

**docker compose up --build**

Öffnen Sie einen Browser und rufen Sie die Anwendung unter [**http://localhost:3000**](http://localhost:3000) auf. Sie sollten eine einfache ToDo-Anwendung sehen.

Um die Anwendung zu stoppen, drücken Sie im Terminal

**Strg + C**.

# Erstellung der README.md

## Vorbereitung

Zu Beginn muss die vorhandene README.md Datei zurückgesetzt werden. Dazu öffnet man die Datei, bearbeitet den Inhalt und löscht anschließend den gesamten Text.

## Schreiben

Die README.md Datei muss in der Markdown-Sprache verfasst werden, um eine strukturierte und lesbare Dokumentation für das Projekt zu erstellen. Dabei werden verschiedene Markdown-Elemente verwendet, um die Datei übersichtlich zu gestallten.

# Verwendung von Git (Commit, Push)

## Einführung

Git ist ein verteiltes Versionskontrollsystem, das die Nachverfolgung von Änderungen im Quellcode ermöglicht. Es bietet eine effiziente Möglichkeit, den Projektfortschritt zu dokumentieren,

## Verwendung

Dateien zum Commit hinzufügen  
Zunächst werden die Dateien, die geändert oder hinzugefügt wurden, für den Commit vorgemerkt.

Um alle Änderungen auf einmal hinzuzufügen:

**git add .**

Commit ausführen  
Nach dem Hinzufügen der Dateien wird ein Commit erstellt. Dabei sollte eine aussagekräftige Commit-Nachricht angegeben werden, die den Zweck der Änderungen beschreibt:

**git commit -m**

### 3.2. Änderungen hochladen (Push)

Sobald der Commit fertig ist, müssen die Änderungen in das Remote-Repository hochgeladen werden. Dafür wird der Push-Befehl verwendet:

Push auf das Remote-Repository  
Mit folgendem Befehl werden die lokalen Commits auf das Remote-Repository übertragen:

**git push**

# Erstellung und Nutzung von Docker-Containern

## Einstellungen

Zu Beginn wird dieser Befehl ausgeführt, um Einstellungen vorzunehmen.

**docker init**

Danach erscheint ein code mit dem man nacheinander wichtige dinge einstellen kann wie zum beispiel die version.

Bei diesem Projekt wurden folgende einstellungen verwendet.

? 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

## Erstellung

Um den Container schließlich zu erstellenmuss folgender Befehl eingegeben werden:

**docker compose up --build**

Dieser Befehl baut das Docker-Image neu und startet den Container. Nachdem der Befehl ausgeführt wurde, ist der Container betriebsbereit und die Anwendung läuft.