

## TO DO APPLICATION

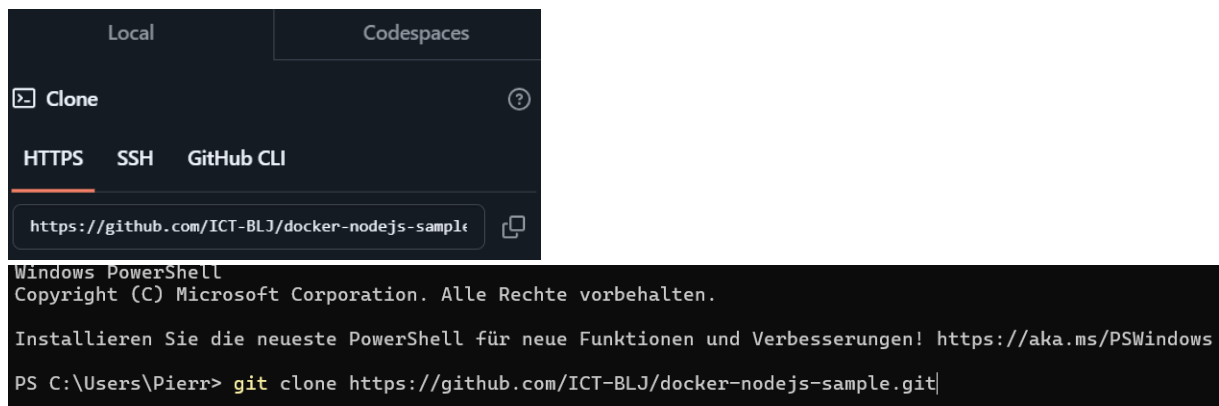


## Inhaltsverzeichnis

Klonen des Repositories .....	2
Einrichtung der Entwicklungsumgebung .....	2
Erstellung der README.md .....	2
Verwendung von Git (Commit, Push) .....	3
Erstellung und Nutzung von Docker-Containern .....	3

## Klonen des Repositories

Zunächst wird das Repository geklont. Geben Sie dazu einfach den Befehl ``git clone «Repository-URL»`` in das Terminal von Git Bash ein. Dadurch wird eine Kopie des Projekts auf Ihren Computer heruntergeladen, und alle Dateien stehen sofort zur Verfügung (z. B. ``git clone https://github.com/benutzername/repositoryname.git``).



## Einrichtung der Entwicklungsumgebung

Nun geht es an die Konfiguration der Entwicklungsumgebung. Stellen Sie sicher, dass die erforderlichen Tools vorhanden sind, wie zum Beispiel ein Code-Editor, häufig ist dies Visual Studio Code. Je nach Projekt müssen möglicherweise auch zusätzliche Bibliotheken installiert werden. Es ist ratsam, diese Schritte im README.md-Dokument festzuhalten, um später darauf zurückgreifen zu können.

## Erstellung der README.md

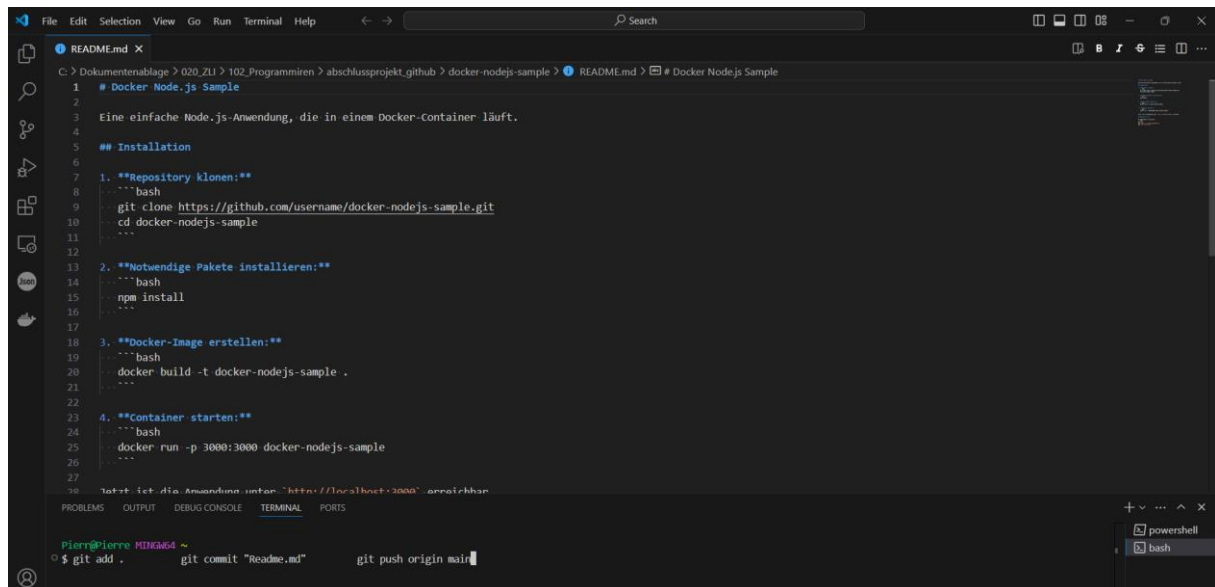
Die Datei README.md ist eine wichtige Informationsquelle für das Projekt. Sie sollte grundlegende Informationen enthalten, wie zum Beispiel:

Projektname: Der Titel des Projekts.

Installation: Eine Schritt-für-Schritt-Anleitung zur Installation des Projekts.

Nutzung: Eine Erklärung zur Verwendung des Projekts.

So haben andere jederzeit eine gute Übersicht über das Projekt.



```
1 # Docker Node.js Sample
2
3 Eine einfache Node.js-Anwendung, die in einem Docker-Container läuft.
4
5 ## Installation
6
7 1. **Repository klonen:**
8    ```bash
9    git clone https://github.com/username/docker-nodejs-sample.git
10   cd docker-nodejs-sample
11   ...
12
13 2. **Notwendige Pakete installieren:**
14    ```bash
15    npm install
16    ...
17
18 3. **Docker-Image erstellen:**
19    ```bash
20    docker build -t docker-nodejs-sample .
21    ...
22
23 4. **Container starten:**
24    ```bash
25    docker run -p 3000:3000 docker-nodejs-sample
26    ...
27
28 Jetzt ist die Anwendung unter 'http://localhost:3000' erreichbar
29
30 $ git add .
31 $ git commit -m "Readme.md"
32 $ git push origin main
```

## Verwendung von Git (Commit, Push)

Zur Versionskontrolle wird Git verwendet. Nachdem Änderungen an den Dateien vorgenommen wurden, sollten diese zunächst mit ``git add <Dateiname>`` zur Staging-Area (dem Zwischenspeicher) hinzugefügt werden. Anschließend kann mit ``git commit -m "kurze Beschreibung der Änderungen"`` ein Commit erstellt werden. Um die Änderungen ins Remote-Repository (Cloud-Zwischenspeicher) zu übertragen, geben Sie einfach ``git push`` ein. So bleibt alles auf dem neuesten Stand.

## Erstellung und Nutzung von Docker-Containern

Docker wird eingesetzt, um Anwendungen in Containern zu isolieren. Um einen neuen Container zu erstellen, benötigen Sie ein Dockerfile, das die notwendigen Anweisungen enthält. Mit dem Befehl ``docker build -t <Container-Name>`` wird der Container erstellt. Um ihn zu starten, geben Sie ``docker run <Container-Name>`` ein. Docker erleichtert die Verwaltung der Entwicklungsumgebung und sorgt für eine konsistente Umgebung.

[illegible]