

Dokumentation Vorgehensweise

Minel Saipi

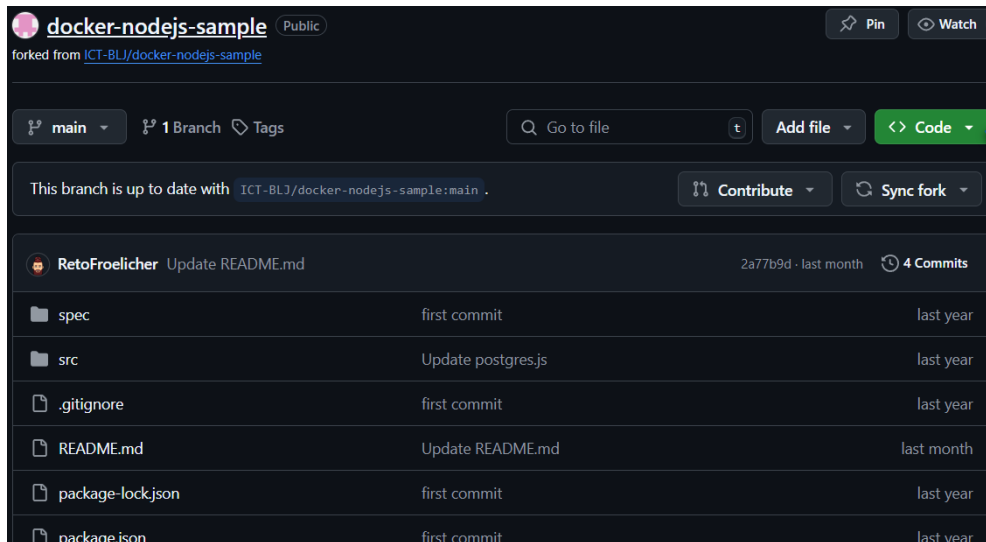
30.10.2024

Inhalt

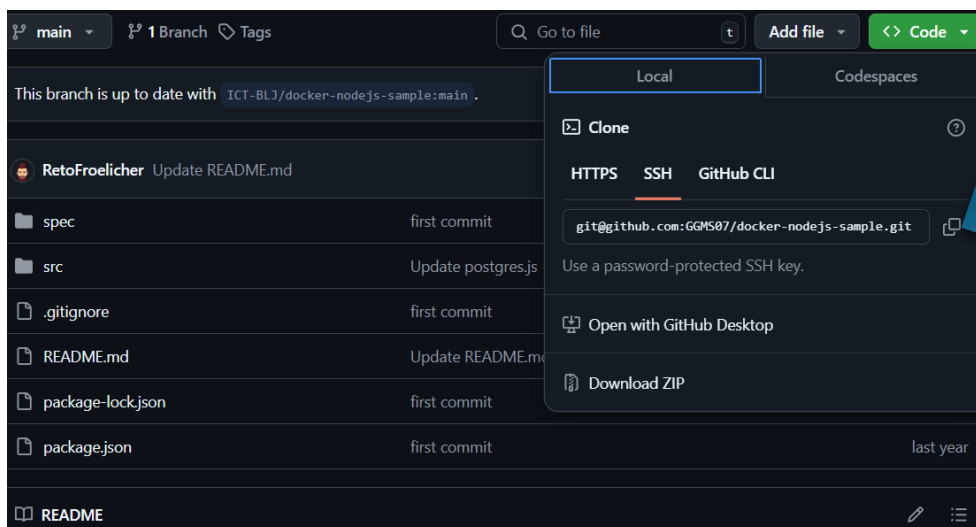
Überschrift 1.....	Fehler! Textmarke nicht definiert.
Überschrift 2.....	Fehler! Textmarke nicht definiert.
Überschrift 3.....	Fehler! Textmarke nicht definiert.
Überschrift 4.....	Fehler! Textmarke nicht definiert.
Überschrift 5.....	Fehler! Textmarke nicht definiert.
Überschrift 6.....	Fehler! Textmarke nicht definiert.
Überschrift 7.....	Fehler! Textmarke nicht definiert.
Überschrift 8.....	Fehler! Textmarke nicht definiert.
Überschrift 9.....	Fehler! Textmarke nicht definiert.
Überschrift 10.....	Fehler! Textmarke nicht definiert.
Überschrift 11.....	Fehler! Textmarke nicht definiert.
Überschrift 12.....	Fehler! Textmarke nicht definiert.
Überschrift 13.....	Fehler! Textmarke nicht definiert.

1. Klonen des Repositories

Um ein Repository zu klonen muss man im ersten Schritt zu seinem Repository navigieren. Danach klickt man oberhalb der Liste der Dateien auf « <>Code ».



Im nächsten Schritt kopiert man die URL, welche man benutzen möchte es gibt hierbei 3 Möglichkeiten: HTTPS, SSH oder GitHub CLI. Wenn man sich für SSH entscheidet braucht man einen SSH-Schlüssel damit das klonen überhaupt möglich ist.



Nachdem öffnet man Git Bash und ändert dort das aktuelle Verzeichnis mithilfe vom Befehl «cd», in das Verzeichnis, indem das ganze geklont sein soll.

Sobald man das Verzeichnis gefunden hat, gibt man diesen Befehl ein:

git clone (URL)

Bestätige das ganze mit der Eingabetaste.

2. Einrichtung der Entwicklungsumgebung

Um die Entwicklungsumgebung einwandfrei einzurichten, muss man sich folgende Dinge konfigurieren:

Docker Desktop, diesen kann man über die Webseite von Docker herunterladen.



Git, kann man ebenfalls von der online Webseite herunterladen.



GitHub, Konto erstellen.



3. Erstellung der README.md

Beim Erstellen der README.md muss man den kopierten Ordner im VS öffnen und dann unter neues Dokument eine Datei mit dem Namen README.md erstellen.

Die README.md ist mit markdown zu bearbeiten, da diese Sprache auf .md Dateien spezifiziert ist. Dabei kann man Links benutzen die Schrift **Fett** darstellen lassen oder *kursiv*.

Installation des Projekts

1. Fork erstellen und klonen

Um einen Fork zu erstellen muss man folgende Schritte beachten:

1. In das Repository gehen, welches man forken will.
2. In der oberen rechten Ecke der Seite auf Forken klicken.
3. In dem Fenster wählt man den Besitzer und gibt dem Fork einen beliebigen Namen. Bei Gelegenheit kann man auch eine Beschreibung einfügen.
4. Sobald du alles ausgefüllt hast das du brauchst, kannst du auf den grünen Knopf unten rechts gehen mit der Beschriftung "Create fork". Dann bist du ach schon fertig.

Um einen Fork zu klonen befolgen sie folgende Schritte:

1. Navigiere zu deinem Fork.
2. Klicke oberhalb der Liste der Dateien auf "<> Code".
3. Kopiere die URL für das Repository.
 - Wähle dann über welche URL du es klonen möchtest (HTTPS/ SSH/ GitHub CLI). Beachte das du beim SSH einen SSH-Schlüssel benötigst. Kopiere die URL mit der Grafik rechts neben dem Code.
4. Öffne Git Bash.
5. Ändere dein aktuelles Verzeichnis an einen Ort an dem du das geklonte Verzeichnis speichern willst.
6. Gebe folgenden Befehl in Git Bash ein:
`git clone(deine kopierte URL)`
7. Bestätige den Befehl mit der Eingabetaste.
[GitHub](#)

Hier ist ein Beispiel wie es geöffnet aussieht.

4. Verwendung von Git

Git hatte ich bei folgenden Dingen benutzt:

- Klonen von Repository auf Rechner
- Pushen von Dateien auf GitHub
- Status checken
- Verzeichnis auswählen zum Abspeichern

```
drwxr-xr-x 1 saipi 197611      0 Oct 30 16:20 ZLI/
-rw-r--r-- 1 saipi 197611    15454 Jun 28  2023 dism.log
-rw-r--r-- 1 saipi 197611 6576664576 Oct 30 08:56 hiberfil.sys
drwxr-xr-x 1 saipi 197611      0 Dec 21  2022 hp/
-rw-r--r-- 1 saipi 197611    1013290 Jun  7  2023 install.log
-rw-r--r-- 1 saipi 197611 20469080064 Oct 30 15:49 pagefile.sys
-rw-r--r-- 1 saipi 197611    16777216 Oct 30 08:56 swapfile.sys

saipi@DESKTOP-T06JVVD MINGW64 /c
$ cd ZLI/

saipi@DESKTOP-T06JVVD MINGW64 /c/ZLI
$ git clone git@github.com:GGMS07/docker-nodejs-sample.git
Cloning into 'docker-nodejs-sample'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 64 (delta 6), reused 4 (delta 4), pack-reused 27 (from 1)
Receiving objects: 100% (64/64), 1.71 MiB | 3.49 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

5. Erstellung und Benutzung von Docker Containern

Um ein Docker Container zu erstellen, muss man das Terminal als Admin öffnen. Im nächsten Schritt wechselt man in das Verzeichnis, in dem die Dateien von der Anwendung sind. Sobald man dies getan hat gibt man den Befehl: `docker init`, um einen Container zu erstellen.

Nachdem man den Container erstellt hat kann man einen weiter Befehl zum Starten der Anwendung eingeben, dieser lautet wie folgt: `docker compose up`.

Sobald man diese Schritte erledigt hat, kann man die ganze Anwendung schon starten wenn man alle Schritte davor richtig ausgeführt hat.

