



# DEVELOPMENT FUNDAMENTALS

Stokic Teodora

## Inhalt

Klonen des Repositories .....	2
Einrichtung der Entwicklungsumgebung.....	2
<b>Benutzereinstellungen</b> .....	2
<b>Theme</b> .....	3
<b>Erweiterungen hinzufügen</b> .....	3
<b>Docker-Konfiguration und -Installation</b> .....	3
Erstellung der README.md .....	4
Verwendung von Git (Commit, Push) .....	4
<b>Was ist git commit?</b> .....	4
<b>Syntax</b> .....	4
<b>Was ist git push?</b> .....	4
<b>Syntax</b> .....	5
Erstellung und Nutzung von Docker-Containern .....	5

## Klonen des Repositories

Um einen Fork auf GitHub zu erstellen, suchst du zunächst das gewünschte Repository. In unserem Fall ist das das Projekt `docker-nodejs-sample`. Öffne den entsprechenden Link und klicke oben rechts auf „Fork“. Danach wählst du „Create a new fork“, wodurch GitHub automatisch eine Kopie dieses Repositories in deinem eigenen Account anlegt.

Anschliessend öffnest du Git Bash. Dort klonst du deinen Fork, indem du den folgenden Befehl eingibst:

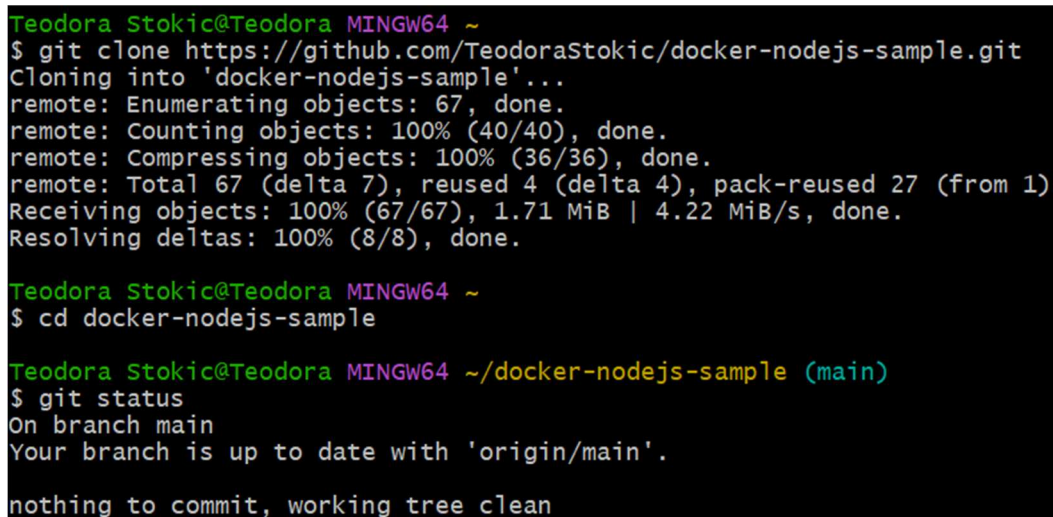
```
git clone https://github.com/[git-username]/docker-nodejs-sample.git
```

Danach wechselst du in den Ordner des geklonten Projekts mit dem Befehl:

```
cd docker-nodejs-sample
```

Um sicherzugehen, dass du dich im richtigen Verzeichnis befindest, kannst du den Status des Repositories überprüfen:

```
git status
```



```
Teodora Stokic@Teodora MINGW64 ~
$ git clone https://github.com/TeodoraStokic/docker-nodejs-sample.git
Cloning into 'docker-nodejs-sample'...
remote: Enumerating objects: 67, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 67 (delta 7), reused 4 (delta 4), pack-reused 27 (from 1)
Receiving objects: 100% (67/67), 1.71 MiB | 4.22 MiB/s, done.
Resolving deltas: 100% (8/8), done.

Teodora Stokic@Teodora MINGW64 ~
$ cd docker-nodejs-sample

Teodora Stokic@Teodora MINGW64 ~/docker-nodejs-sample (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

## Einrichtung der Entwicklungsumgebung

### Benutzereinstellungen

Die Benutzereinstellungen von VS Code lassen sich über das Kommando Preferences: Open Settings (JSON) aufrufen → Zahnrad, Settings, oben rechts Preferences: Open Settings (JSON) auswählen. Es öffnen sich zwei Editorfenster mit JSON-Dateien: Rechts wird die Datei mit den momentanen Benutzereinstellungen angezeigt und links sind die Standardeinstellungen zu sehen. Nur die rechte Seite lässt sich bearbeiten.

Folgend einige wichtige Einstellungen:

```
{  
  "files.eol": "\n",  
  "files.insertFinalNewline": true,  
  "files.trimFinalNewlines": true,  
  "editor.formatOnSave": true,  
  "editor.useTabStops": false,  
  "editor.tabSize": 2,  
  "editor.renderWhitespace": "all",  
}
```

## Theme

Mit Themes können Sie die Farben in der Benutzeroberfläche von Visual Studio Code an deine Präferenzen und die Arbeitsumgebung anpassen. Um nach Themen zu suchen, tippe *Theme* in das Suchfeld der Erweiterungsansicht (Ctrl+Shift+X) ein.

Man kann danach das aktive Farbschema mit den Benutzereinstellungen *Workbench.colorCustomizations* und *editor.tokenColorCustomizations* zusätzlich anpassen.

## Erweiterungen hinzufügen

VS Code bietet eine Vielzahl an Erweiterungen, die die Entwicklungsarbeit erleichtern.

## Docker-Konfiguration und -Installation

- Um Docker Desktop auf Windows einzurichten, lade zunächst die passende Version von der Docker-Website herunter. Vor der Installation wirst du gefragt, ob du die Version für Windows ARM oder AMD benötigst. Wenn du dir unsicher bist, kannst du dies einfach überprüfen, indem du CMD öffnest und den Befehl *echo %PROCESSOR\_ARCHITECTURE%* eingibst.
- Nach der Installation von Docker Desktop lade die Docker-Erweiterung für Visual Studio Code herunter.
- Öffne nun die PowerShell als Administrator und führe die beiden Befehle *dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart* sowie *dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart* aus.
- Starte deinen Computer neu

- Nach der Aktivierung ist noch WSL1 als Standard eingestellt, das ändert man mit Hilfe der folgenden Eingabe `wsl --set-default-version 2`.
- Anschliessend kannst du über den Microsoft Store eine Linux-Distribution deiner Wahl installieren.
- Sobald alles eingerichtet ist, kannst du testen, ob Docker korrekt funktioniert, indem du in PowerShell den Befehl `docker run hello-world` ausführst.

```
PS C:\windows\system32> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\windows\system32>
```

## Erstellung der README.md

Öffne in VS Code den Ordner, in dem du das README erstellen möchtest. Drücke auf die Tastenkombination CTRL+N. Speichere die Datei mit CTRL+S. Wähle beim Speichern den Dateityp Markdown (.md) aus und gebe der Datei einen passenden Namen.

## Verwendung von Git (Commit, Push)

### Was ist git commit?

Ein Git Commit dient dazu, Änderungen im aktuellen Repository zu sichern. Dabei wird ein Snapshot des Projekts zu einem bestimmten Zeitpunkt erstellt. Jede Commit-Nachricht beschreibt die vorgenommenen Änderungen und liefert somit eine klare Übersicht über den Fortschritt des Projekts.

### Syntax

```
git commit -m "Deine Commit-Nachricht"
```

### Was ist git push?

Mit Git Push werden die Inhalte deines lokalen Repositories in ein Remote-Repository hochgeladen. So kannst du deine Änderungen mit anderen teilen und sicherstellen, dass das Remote-Repository immer auf dem neuesten Stand ist.

## Syntax

git push origin main

## Erstellung und Nutzung von Docker-Containern

- Im Verzeichnis `docker-nodejs-sample` führe im Terminal den folgenden Befehl aus:  
*docker init*

Der Befehl `docker init` erstellt eine Standardkonfiguration für dein Projekt und stellt dir einige Fragen zu deiner Anwendung. Die du wie in der Abbildung unten beantworten sollst.

```
Let's get started!
? What application platform does your project use? N [Use arrows to move, type to filter]
? What application platform does your project use? Node
? What version of Node do you want to use? 18.0.0
? 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? [tab for suggestions] (node index.js) node src/index.js
? What command do you want to use to start the app? node src/index.js
? What port does your server listen on? 3000
? What port does your server listen on? 3000
```

- Nun führe im Terminal den folgenden Befehl aus:  
*docker compose up --build*  
Dieser Befehl startet die Anwendung und erstellt bei Bedarf die Docker-Container neu.
- Öffne anschliessend einen Webbrowser und gehe zu <http://localhost:3000>. Dort solltest du eine einfache To-Do-Anwendung sehen.

## ACHTUNG

```
-----
22 |     # into this layer.
23 | >>> RUN --mount=type=bind,source=package.json,target=package.json \
24 | >>>     --mount=type=bind,source=package-lock.json,target=package-lock.json \
25 | >>>     --mount=type=cache,target=/root/.npm \
26 | >>>     npm ci --omit=dev
27 |
-----
failed to solve: process "/bin/sh -c npm ci --omit=dev" did not complete successfully: exit
code: 1
```

Wenn so etwas erscheint, kann es daran liegen, dass die Version 18.0.0 nicht mehr aktuell ist.

Um dieses Problem zu beheben, öffnest du in Visual Studio Code die Dockerfile und änderst die Version auf 22.18.0.