Vorgehensweise zum Aufsetzen einer React-Entwicklungsumgebung in Docker

# Vorbereitung: Projektstruktur für neue Create-React-App

Das folgende Vorgehen basiert im Wesentlichen auf „*Developing React Inside Docker“* (<https://willschenk.com/howto/2020/developing_react_inside_docker/#bootstrap>).

1. Erstelle neues Verzeichnis „app“: mkdir app
2. Wechsel in das neue Verzeichnis: „cd app“
3. Erstelle eine „Dockerfile.initial“:  
   A screenshot of a computer

   Description automatically generated
4. Öffne ein Terminalfenster und wechsel in das neue Verzeichnis „app“
   1. Erstelle ein Docker-Image mittels Dockerfile.initial:   
      „docker build . -f Dockerfile.initial -t testapp“
   2. Starte einen Container, der den neu zu erstellenden Ordner „gui“ gemountet hat:  
      „docker run -it --rm -v ${PWD}/gui:/app app npx create-react-app .“
   3. Nun sollte ein „create-react-app“ erzeugter Ordner im lokalen „gui“-Verzeichnis existieren.
   4. Falls die Ordner- und Dateiberichtigung nur dem „root-User“ zugeordnet sind, dann müssen diese auf den User erweitert werden. Dies ist je nach Betriebssystem unterschiedlich durchzuführen. Es wird auf die Literatur bzw. Internet verwiesen
5. Lösche den Ordner „node\_modules“  
   „rm -rf node\_modules“  
   Für eine Begründung wird auf https://willschenk.com/howto/2020/developing\_react\_inside\_docker/#setting-up-for-development verwiesen:   
   A white paper with black text

   Description automatically generated

# Aufsetzen der Entwicklungsumgebung

Das folgende Vorgehen basiert im Wesentlichen auf „*Properly Setting Up React Development Environment Using Docker*“ (<https://medium.com/@sudiptob2/properly-setting-up-react-development-environment-using-docker-a2de46464d0b>)

## Dockerfile

Erzeuge das folgende „Dockerfile“ im Ordner „gui“, um die App selbst außerhalb des lokalen Verzeichnisses laufen zu lassen:

A screenshot of a computer program

Description automatically generated

**Bemerkung**: Falls die json-Dateien jemals geändert werden, dann müssen die folgenden Schritte zur Erstellung des Docker-Containers wiederholt werden.

## Docker-Compose

Erzeuge die folgende Datei namens „docker-compose.yml“ im Projektverzeichnis „gui“  


Danach führe „docker-compose build“ und „docker-compose up aus, um die Entwicklungsumgebung zu starten.

# Überprüfung der Entwicklungsumgebung

Die Entwicklungsumgebung sollte über die URL <http://localhost:3000> sichtbar sein:  
A blue and white logo

Description automatically generated

## Ändere die Datei „app.js“ im Ordner „src“

A screen shot of a computer program

Description automatically generated

Ändere z.B. in Zeile 10 den Code, indem am Ende des Satzes der Zusatz „XXX“ hinzugefügt wird:



Speichere diese Daten ab und die Änderung sollte in der laufenden App sichtbar werden:

A blue and white atom symbol

Description automatically generated

## Hinzufügen einer Bibliothek

Eine React-Bibliothek kann über den folgenden Befehl im Terminalfenster hinzugefügt werden:

„docker-compose exec web npm install package-name“.

Zum Beispiel soll die App verändert werden, indem die Dateie „App.js“ ersetzt wird durch die folgende Datei:

A computer screen shot of a program code

Description automatically generated

Hierfür muss zuerst die Bibliothek „react-globe.gl“ hinzugefügt werden. Öffne ein neues Terminalfenster im Verzeichnis „app/gui“ und führe folgenden Befehl aus:

„docker-compose exec web npm install react-globe.gl“

Dann ändere die Datei „App.js“ und speicher die Änderungen ab. Danach sollten die Änderung in der laufenden App sichtbar werden:

A planet with colorful lines around it

Description automatically generated

# Workflow für Entwickler

Das folgende Vorgehen basiert im Wesentlichen auf „*Properly Setting Up React Development Environment Using Docker*“ (<https://medium.com/@sudiptob2/properly-setting-up-react-development-environment-using-docker-a2de46464d0b>):

A screenshot of a computer

Description automatically generated

A white rectangular object with a black border

Description automatically generated with medium confidence