



“Inhalt der Checkliste

Inhalt der Checkliste.....	1
Projektanforderung Truck Signs API.....	2
1. Repository.....	2
.gitignore Datei.....	2
Dockerfile.....	2
README.md.....	2
2. Dokumentation.....	3
3. Hinweise.....	3

Projektabgabe - Truck Signs API

Bitte erfülle alle Punkte auf dieser Liste, bevor du das Projekt einreichst. Solltest du weitere Extras eingebaut haben, erwähne das kurz, damit sich die Mentoren dies bei Bedarf anschauen können.

1. Repository

Vorhandene Dateien

- ☐ Es wurde eine Dockerfile Datei angelegt, die das entstehende Container-Image beschreibt. Das Dockerfile genügt den nachstehend formulierten Anforderungen
- ☐ Eine Datei namens README.md ist vorhanden und entsprechend der Kriterien unten erweitert worden
- ☐ Es befinden sich keine weiteren Dateien im Repository, ohne dass diese explizit in der README.md benannt und beschrieben werden.

Dockerfile

- ☐ Du hast ein passendes Base-Image als Grundlage deines eigenen Container Images definiert
- ☐ Du hast einen Port des Containers exposed, sodass der Container vom Internet aus erreichbar ist
- ☐ Du hast alle benötigten Schritte als Teil des Image-Builds Prozesses definiert, sodass die Anwendung ohne zusätzlichen Aufwand gestartet wird
 - ☐ Dein Entrypoint erledigt die migrations, collectstatic, sowie createsuperuser Befehle

README.md

- ☐ Die README sollte ein Inhaltsverzeichnis a.k.a. eine Table-of-Contents (ToC) enthalten
 - ☐ Die einzelnen Sektionen sind in der ToC verlinkt
- ☐ Eine Sektion mit einer Beschreibung des Repositories muss vorhanden sein. In dieser Beschreibung sollte genannt werden was die wesentlichen Inhalte sind, was der Zweck des Repositories ist
- ☐ Eine Sektion "Quickstart" sollte als Teil der README enthalten sein. Hier sollen kurz die Voraussetzungen genannt und eine Schnellstart-Anleitung beschrieben sein.
 - ☐ es sollte hierbei eine Sektion how-to-build-the-image geben
- ☐ Es soll eine ausführliche Variante der vorgenannten Sektion als "Usage" enthalten sein. Hier soll genauer auf die Konfiguration und Konfigurierbarkeit eingegangen werden, d.h. es soll auch erklärt werden, wie relevante Passagen modifiziert werden können, um andere Resultate zu erzielen.

- ☐ Es muss dokumentiert sein, wie ein Container image erzeugt werden kann
- ☐ Der **docker run** Befehl muss dokumentiert sein - env-variablen oder andere sensitive information sollte durch Platzhalter ersetzt werden

2. Dokumentation

Die Dokumentation des Codes, sowie des Projektes soll im Repository in Form einer README Datei stattfinden.

Die Dokumentationssprache für alle Projekte (und zugehörige Unterlagen) ist englisch.

3. Hinweise

Allgemeine Hinweise

- ☐ Zusätzlich zu deinem GitHub Repository solltest du ein kurzes Loom Video (maximal 5min.) aufnehmen und bereitstellen, indem du kurz deine Abgabe zeigst und vorstellst was du getan hast - dabei musst du nicht alle Details erwähnen, jedoch sollst du auf alle relevanten Schritte kurz eingehen und diese zeigen.
- ☐ Die Container sollen hierbei **NICHT** mit `docker compose` gestartet werden.
- ☐ Die Datenbank sollte auch in einem Container betrieben werden
 - ☐ Das Backend sollte mit der Datenbank kommunizieren können, dazu müssen die Container im selben Netzwerk (network) sein
 - ☐ Container im selben Netzwerk können sich gegenseitig mittels des hostnames/container namens erreichen

Sicherheitshinweise

- ☐ Speichere keine SSH-Keys im Workspace deines Git-Repositories
- ☐ Speichere keine Passwörter, Tokens, oder Benutzernamen in deinem Code. Verwende hierfür stattdessen Environment-Variablen
- ☐ Speichere keine IP-Adressen, oder sonstigen sensiblen Informationen in einem Git Repository

Code-Konventionen

- ☐ Für build-args, environment Variablen und Shell-Variablen gilt folgende Namenskonvention: UPPER_CASE_WITH_UNDERSCORE
- ☐ Bei einer Referenz auf eine Variable sollte immer die `{}`-Notation verwendet werden um Fehler in der Interpretation zu vermeiden: `${SOME_VAR_VALUE}`, statt: `$SOME_VAR_VALUE`
- ☐ Es sollten für Build-args, oder Environment Variablen "Default"-Werte konfiguriert werden, allerdings nur dann, wenn dies Sinn ergibt.



- ☐ Kritische Konfiguration wie Tokens, Passwörter oder ähnliches sollte nicht im Code-Repository gespeichert sein, sondern bspw. durch die Verwendung eines .env-files in einen Container hineingegeben werden

Testing

Bevor Du dein Projekt einreichst, solltest du die folgenden Dinge sicherstellen und getestet haben:

- ☐ Die Truck Signs API ist erreichbar unter der IP-Adresse deiner Cloud-VM auf Port 8020
- ☐ Dein ENTRYPOINT startet die WSGI Applikation, KEINEN dev-server.
- ☐ Nach einem Neustart des Setups, sind die konfigurierten Daten noch vorhanden und werden nicht gelöscht oder überschrieben
- ☐ Die Container werden neu gestartet, sobald ein Fehler passiert, der zum Terminieren des Containers führt.