

# DSE22Gruppe10-Wartungshandbuch

Diego Krupitza, Jan Müller, Kian Pouresmaeil

17. Mai 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Entwicklungsumgebung</b>	<b>1</b>
1.1	Starten der Java Services . . . . .	1
1.2	Starten der Typescript Services . . . . .	1
1.3	Konfiguration der Services . . . . .	1
1.3.1	Konfiguration Entity-Service . . . . .	1
1.3.2	Konfiguration Tracking-Service . . . . .	2
1.3.3	Konfiguration Simulator-Service . . . . .	2
1.3.4	Konfiguration Flowcontrol-Service . . . . .	2
1.3.5	Konfiguration Gateway-Service . . . . .	2
<b>2</b>	<b>Frameworks und Libraries</b>	<b>3</b>
2.1	Entity-Service . . . . .	3
2.2	Tracking-Service . . . . .	3
2.3	Cockpit . . . . .	3
2.4	Flocontrol-Service . . . . .	3
2.5	Simulator-Service . . . . .	3
2.6	Gateway-Service . . . . .	3
<b>3</b>	<b>Build Process</b>	<b>4</b>
<b>4</b>	<b>Test process</b>	<b>4</b>
<b>5</b>	<b>Deployment</b>	<b>4</b>
<b>6</b>	<b>API Documentation</b>	<b>4</b>

# 1 Entwicklungsumgebung

Für das Entwickeln der Java Microservices werden Java 17, Maven 3.8.5 sowie Yarn 1.22.18 oder neuer benötigt. Bevor man die Entwicklungsumgebung starten kann, müssen eine Datenbank sowie eine „MoM“ mithilfe von Docker Compose gestartet werden. Dazu kann der Befehl `docker-compose up -d` im Hauptverzeichnis des Projektes ausgeführt werden.

Es gibt keine strikte Reihenfolge für das Starten der Microservices. Die folgende Reihenfolge bietet sich jedoch an, um die Wartezeit zu verringern.

1. Entity-Service
2. Tracking-Service
3. Flowcontrol-Service
4. Simulator-Service
5. Gateway-Service
6. Cockpit

## 1.1 Starten der Java Services

Die Services *Flowcontrol-Service*, *Simulator-Service* und *Gateway-Service* können mittels Maven gestartet werden. Hierfür muss im Verzeichnis des gewünschten Dienstes der Befehl `mvn spring-boot:run` ausgeführt werden.

## 1.2 Starten der Typescript Services

Die TypeScript Services sowie das Cockpit lassen sich über den Befehl `yarn dev` im jeweiligen Verzeichnis starten. Zuvor muss der Befehl `yarn install` verwendet werden, um benötigte Bibliotheken zu installieren.

## 1.3 Konfiguration der Services

Einige Services der Applikation bieten Anpassungsmöglichkeiten über Umgebungsvariablen, wobei Standardwerte eine Konfiguration optional machen.

### 1.3.1 Konfiguration Entity-Service

Folgende Umgebungsvariablen sind anpassbar.

Name	Standardwert	Beschreibung
MONGO_DB_HOST	localhost:27017	Adresse der MongoDB Instanz
MONGO_DB_USER	admin	Username der MongoDB Instanz
MONGO_DB_PWD	admin	Passwort der MongoDB Instanz
MONGO_DB_NAME	local-entity-db	Name der MongoDB Datenbank
RABBIT_MQ_HOST	localhost	Adresse RabbitMQ Instanz

Tabelle 1: Environment variablen für Entity-Service

### 1.3.2 Konfiguration Tracking-Service

Folgende Umgebungsvariablen sind anpassbar.

Name	Standardwert	Beschreibung
MONGO_DB_HOST	localhost:27017	Adresse der MongoDB Instanz
MONGO_DB_USER	admin	Username der MongoDB Instanz
MONGO_DB_PWD	admin	Passwort der MongoDB Instanz
MONGO_DB_NAME	local-entity-db	Name der MongoDB Datenbank
RABBIT_MQ_HOST	localhost	Adresse RabbitMQ Instanz

Tabelle 2: Environment variablen für Tracking-Service

### 1.3.3 Konfiguration Simulator-Service

Folgende Umgebungsvariablen sind anpassbar.

Name	Standardwert	Beschreibung
ENTITY_SERVICE_IP	localhost	Adresse des Entity-Service
TRACKING_SERVICE_IP	localhost	Adresse des Tracking-Service
FLOW_SERVICE_IP	localhost	Adresse des Flowcontrol-Service
SPRING_RABBITMQ_HOST	localhost	Adresse der RabbitMQ Instanz

Tabelle 3: Environment variablen für Simulator-Service

### 1.3.4 Konfiguration Flowcontrol-Service

Folgende Umgebungsvariablen sind anpassbar.

### 1.3.5 Konfiguration Gateway-Service

Folgende Umgebungsvariablen sind anpassbar.

Name	Standardwert	Beschreibung
ENTITY_SERVICE_IP	localhost	Adresse des Entity-Service
TRACKING_SERVICE_IP	localhost	Adresse des Tracking-Service
SIMULATOR_SERVICE_IP	localhost	Adresse des Simulator-Service
SPRING_RABBITMQ_HOST	localhost	Adresse der RabbitMQ Instanz

Tabelle 4: Environment variablen für Simulator-Service

Name	Standardwert	Beschreibung
ENTITY_SERVICE_IP	localhost	Adresse des Entity-Service
TRACKING_SERVICE_IP	localhost	Adresse des Tracking-Service
SIMULATOR_SERVICE_IP	localhost	Adresse des Simulator-Service
FLOW_SERVICE_IP	localhost	Adresse des Flowcontrol-Service

Tabelle 5: Environment variablen für Gateway-Service

## 2 Frameworks und Libraries

### 2.1 Entity-Service

JAN

### 2.2 Tracking-Service

JAN

### 2.3 Cockpit

JAN

### 2.4 Flocontrol-Service

DIEGO

### 2.5 Simulator-Service

DIEGO

### 2.6 Gateway-Service

DIEGO

### 3 Build Process

Die Java-Dienste lassen über den Befehl `mvn package -DskipTests` bauen. Analog dazu baut der Befehl `yarn build` die TypeScript-Dienste.

Letztendlich findet dies nur beim Bauen der Docker-Images statt, für welche pro Verzeichnis ein Für eine kürzere Bauzeit wurden die Dockerfiles so programmiert, dass diese mehrere Ebenen verwenden und Dependencies zwischenspeichern. Um das Bauen der Docker-Images weiter zu vereinfachen, können die `dockerize`-Skripte der Datei `package.json` verwendet werden. Der lokale Docker-Nutzer muss dafür Zugriffsrechte auf die Docker Hub Repositories besitzen.

### 4 Test process

KIAN

### 5 Deployment

DIEGO

### 6 API Documentation

KIAN