

DSE22Gruppe10-Wartungshandbuch

Diego Krupitza, Jan Müller, Kian Pouresmaeil

16. Mai 2022

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | Entwicklungsumgebung | 1 |
| 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 |
| 2 | Frameworks und Libraries | 3 |
| 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 |
| 3 | Build Process | 4 |
| 4 | Test process | 4 |
| 5 | Deployment | 4 |
| 6 | API Documentation | 4 |

1 Entwicklungsumgebung

Für das Entwickeln der Java Microservices braucht man Java 17 LTS, sowie Maven (mit einer Version größer gleich 3.8.5). Bevor man die Entwicklungsumgebung starten kann muss man die *docker-compose.yml* Datei im Hauptordner des Projektes mittels *docker-compose up* starten.

Sobald Docker den rabbitmq und monodb container gestartet hat, kann man mit dem starten der Microservices beginnen.

An sich gibt es keine strikte Reihenfolge für das starten der Microservices, jedoch bietet es sich an folgenden Reihenfolge zu befolgen, um die Wartezeit zu verringern.

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

1.1 Starten der Java Services

Die Services *Flowcontrol-Service*, *Simulato-Service* und *Gateway-Service* können mittels Maven gestartet werden. Hierfür muss man in den Ordner des gewünschten Services gehen und den folgenden command ausführen:

```
mvn spring-boot:run
```

1.2 Starten der Typescript Services

JAN MAGIC

1.3 Konfiguration der Services

Einige Services der Applikation bieten Anpassungsmöglichkeit. Diese Anpassungen werden über Environment variablen gesteuert. Die Applikationen laufen auch ohne das Anpassen dieser Werte.

1.3.1 Konfiguration Entity-Service

Folgende Environment variablen sind möglich zu setzen.

| Name | Standardwert | Beschreibung |
|----------------|-----------------|--|
| MONGO_DB_HOST | localhost:27017 | Hostname / die ip wo die MongoDB Instanz läuft |
| MONGO_DB_USER | admin | Username für die Authentifizierung bei der MongoDB Instanz |
| MONGO_DB_PWD | admin | Passwort für die Authentifizierung bei der MongoDB Instanz |
| MONGO_DB_NAME | local-entity-db | MongoDB Datenbank die das Service benutzen soll |
| RABBIT_MQ_HOST | localhost | Hostname / die IP wo die RabbitMQ Instanz läuft |

Tabelle 1: Environment variablen für Entity-Service

1.3.2 Konfiguration Tracking-Service

Folgende Environment variablen sind möglich zu setzen.

| Name | Standardwert | Beschreibung |
|----------------|-------------------|--|
| MONGO_DB_HOST | localhost:27017 | Hostname / die ip wo die MongoDB Instanz läuft |
| MONGO_DB_USER | admin | Username für die Authentifizierung bei der MongoDB Instanz |
| MONGO_DB_PWD | admin | Passwort für die Authentifizierung bei der MongoDB Instanz |
| MONGO_DB_NAME | local-tracking-db | MongoDB Datenbank die das Service benutzen soll |
| RABBIT_MQ_HOST | localhost | Hostname / die IP wo die RabbitMQ Instanz läuft |

Tabelle 2: Environment variablen für Tracking-Service

1.3.3 Konfiguration Simulator-Service

Folgende Environment variablen sind möglich zu setzen.

| Name | Standardwert | Beschreibung |
|----------------------|--------------|---|
| ENTITY_SERVICE_IP | localhost | Hostname / die IP vom Entity-Service |
| TRACKING_SERVICE_IP | localhost | Hostname / die IP vom Tracking-Service |
| FLOW_SERVICE_IP | localhost | Hostname / die IP vom Flowcontrol-Service |
| SPRING_RABBITMQ_HOST | localhost | Hostname / die IP wo die RabbitMQ Instanz läuft |

Tabelle 3: Environment variablen für Simulator-Service

1.3.4 Konfiguration Flowcontrol-Service

Folgende Environment variablen sind möglich zu setzen.

1.3.5 Konfiguration Gateway-Service

Folgende Environment variablen sind möglich zu setzen.

| Name | Standardwert | Beschreibung |
|----------------------|----------------|---|
| ENTITY_SERVICE_IP | localhost | Hostname / die IP vom Entity-Service |
| TRACKING_SERVICE_IP | localhost | Hostname / die IP vom Tracking-Service |
| SIMULATOR_SERVICE_IP | localhostadmin | Hostname / die IP vom Simulator-Service |
| SPRING_RABBITMQ_HOST | localhost | Hostname / die IP wo die RabbitMQ Instanz läuft |

Tabelle 4: Environment variablen für Simulator-Service

| Name | Standardwert | Beschreibung |
|----------------------|----------------|---|
| ENTITY_SERVICE_IP | localhost | Hostname / die IP vom Entity-Service |
| TRACKING_SERVICE_IP | localhost | Hostname / die IP vom Tracking-Service |
| SIMULATOR_SERVICE_IP | localhostadmin | Hostname / die IP vom Simulator-Service |
| FLOW_SERVICE_IP | localhost | Hostname / die IP vom 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

JAN

4 Test process

KIAN

5 Deployment

DIEGO

6 API Documentation

KIAN