



car Sharing now

Ein Auto immer in der Nähe

Readme

Inhaltsverzeichnis

1.Einstellungen anpassen.....	3
1.1 Grundeinstellungen zum ausführen in Containern.....	3
1.2 Einstellungen zum Testen.....	3
2.Erstellen der Jar File als Grundlage der Docker Container.....	4
2.1 Tests und die Jar File.....	4
2.2 Jar Datei einfach ohne Testen erzeugen.....	4
3.Erstmalige verwendung der Software.....	6

1. Einstellungen anpassen

Die Einstellungen für die Anwendung lassen sich in der datei 'application.properties' unter dem Pfad

```
.\\carSharingNow\\carSharingNow\\carSharingNow.zip_expanded\\carSharingNow\\src\\main\\resources
```

einstellen.

1.1 Grundeinstellungen zum ausführen in Containern

Die Grundeinstellungen für das erfolgreiche ausführen der Docker Container ist:

```
spring.datasource.url=jdbc:mysql://carSharingNowDB:3306/cSnDB
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
server.port=8443
vaadin.urlMapping=/*

jwt.auth.secret=J6GOtcwC2NJI1I0VkHu20PacPFGTxpirBxWwynoHjsc=

server.ssl.key-store=classpath:key-cert-mousumi.p12
server.ssl.key-store-password=changeit
server.ssl.keyStoreType=PKCS12
```

1.2 Einstellungen zum Testen

Zum ausführen der Tests muss eine bereits laufende Datenbank vorhanden sein. Diese muss von der Software erreichbar sein, das kann mit dem änder der datasource Url erreicht werden.

```
spring.datasource.url=jdbc:mysql://[hier passende adresse zur DB angeben]:3306/cSnDB
```

Weiter muss noch der für die datenbank vorgesehene Benutzer identifiziert werden:

```
spring.datasource.username=[hier Nutzernamen der Test Datenbank angeben]
spring.datasource.password=[hier Passwort des Nutzers der Test Datenbank angeben]
```

Die restlichen angaben sollten so wie sie sind belassen werden, ansonsten wird es Probleme mit der Verschlüsselung oder der Erreichbarkeit der Anwendung geben.

2. Erstellen der Jar File als Grundlage der Docker Container

Die nachfolgenden Anweisungen müssen über ein Terminal oder Kommandozeile ausgeführt werden. Dies muss im Projektordner unter

```
.\\carSharingNow\\carSharingNow\\carSharingNow.zip_expanded\\carSharingNow
```

Geschehen.

Vorraussetzungen:

Java 17 JDK korrekt installiert

Node.js

Docker Installiert vorzugsweise mit Docker Desktop

2.1 Tests und die Jar File

Das Erstellen der Jar Datei für den Docker container wird mithilfe von Maven erzeugt, dazu muss entweder eine Datenbank vorhanden sein mit der getestet werden kann (Dazu bitte Einstellungen unter 1.2 beachten). *Es wird empfohlen diese Tests nicht mit dem Erstellen der Jar Datei für das Erstellen der Container auszuführen.*

!!Die erzeugte Jar File beim Testen kann nicht als Jar zum Bauen des Containers verwendet werden!!

2.2 Jar Datei einfach ohne Testen erzeugen

Alternativ zum Erstellen der Anwendungsdateien unter Testzwang lässt sich die benötigte Jar Datei auf ohne das Durchführen von Tests erzeugen. Dies ist notwendig wenn die Anwendung beim Kunden installiert werden soll, denn ohne weitere MySQL-Datenbank für die Tests wird das Erzeugen nicht möglich sein.

Ablauf Erzeugen ohne Testen:

1. Navigieren Sie zum Projektordner

```
.\\carSharingNow\\carSharingNow\\carSharingNow.zip_expanded\\carSharingNow
```

2. Führen Sie den folgenden Befehl aus um ohne Tests eine Jar File zu erstellen

```
mvn clean package -Dmaven.test.skip=true
```

3. Nachdem Ausführen des Befehls sollte das Terminal eine Build Success Benachrichtigung ausgeben

```
ingNow-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.1:repackage (repackage) @ carSharingNow ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 45.461 s
[INFO] Finished at: 2022-07-24T12:05:57+02:00
[INFO] -----
```

4. Zum erstellen der Docker Container muss nun noch folgender Befehl ausgeführt werden

```
docker compose up -d
```





















5. Überprüfen ob die Container laufen kann man entweder über das Docker Desktop UI oder mit dem Befehl

```
docker container ls
```

Das Ergebnis sollte im Terminal wie folgt aussehen

```
E:\CodeProjects\carSharingNow\carSharingNow\carSharingNow.zip_expanded\carSharingNow>docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
2f613fbec863   openjdk       "java -jar app.jar"      23 minutes ago Up 48 seconds  0.0.0.0:8443->8443/tcp             carSharingNowSys
1a590bf92799   mysql:8       "docker-entrypoint.s..." 23 minutes ago Up 49 seconds  0.0.0.0:3306->3306/tcp, 33060/tcp  carSharingNowDB
```

und in der Docker Desktop Anwendung so

<input type="checkbox"/>		carsharingnow 2 containers	-	Running (2/2)	-	 Open				
<input type="checkbox"/>		carSharingNowDB 1a590bf92799 	mysql	Running	3306	7 minutes ago 				
<input type="checkbox"/>		carSharingNowSys 2f613fbec863 	openjdk	Running	8443	7 minutes ago 				

3. Erstmalige verwendung der Software

Nachdem erzeugen von neuen Docker Containern mit einem neuen Volume muss zunächst der Admin account erstellt werden. Dazu muss lediglich einmal ein gescheiterter Login versuch unternommen werden und es wird einmalig zu beginn (immer wenn die Datenbank keine Benutzer enthält) ein Account unter dem Nutzernamen admin, mit dem Passwort rotierendeente erzeugt.

!!Hiweis bitte direkt nach dem Ersten Einloggen das Passwort ändern.!!

Ist der Administrations Account erzeugt kann man direkt auf das Adminboard zugreifen und Fahrzeuge erstellen.

!!Dem Admin Account sollte unter keinen Umständen die Admin-Rolle entzogen werden!!