

Cinema - Mario Lengauer

Erstellung des Projektes

```
mvn io.quarkus:quarkus-maven-plugin:1.0.0.CR1:create \
  -DprojectId=at.htl \
  -DprojectArtifactId=cinema \
  -DclassName="at.htl.cinema.rest.cinema" \
  -Dpath="/cinema"
```

Starten des Projektes

Hibernate

1. Dependencies eintragen

```
<dependencies>
  <!-- Hibernate ORM specific dependencies -->
  <dependency>
    <groupId>io.quarkus</groupId>
    <artifactId>quarkus-hibernate-orm</artifactId>
  </dependency>

  <!-- JDBC driver dependencies -->
  <dependency>
    <groupId>io.quarkus</groupId>
    <artifactId>quarkus-jdbc-postgresql</artifactId>
  </dependency>
</dependencies>
```

2. DB konfigurieren

in application.properties folgendes eintragen:

```
quarkus.datasource.url=jdbc:postgresql://192.168.99.100/postgres
quarkus.datasource.driver=org.postgresql.Driver
quarkus.datasource.username=postgres
quarkus.datasource.password=passme
quarkus.datasource.min-size=3
quarkus.datasource.max-size=13
quarkus.hibernate-orm.database.generation=drop-and-create
```

3. Entität erstellen

```
@XmlElement
@Entity
@NamedQuery(name = "Cinema.findAll", query = "select c from Cinema c")
public class Cinema {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String address;
    private LocalDate founded;

    @OneToMany(
        mappedBy = "cinema",
        cascade = CascadeType.ALL,
        orphanRemoval = true
    )
    private List<Hall> halls;

    public Cinema() {
    }

    public Cinema(String name, String address, LocalDate founded) {
        this.name = name;
        this.address = address;
    }
}
```

4. InitBean erstellen

```

@ApplicationScoped
public class InitBean {
    @Inject
    EntityManager em;

    @Transactional
    public void init(@Observes StartupEvent ev){
        System.err.println("-----init-----");
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
        Cinema cineplexLinz = new Cinema("Cineplex Linz", "Prinz-Eugen-Straße 22, 4020
Linz", LocalDate.parse("1999-12-08", formatter));
        List<Hall> halls = new ArrayList<>();
        halls.add(new Hall("DOLBY CINEMA", 394, cineplexLinz));
        halls.add(new Hall("Saal 2", 133, cineplexLinz));
        halls.add(new Hall("Saal 3", 132, cineplexLinz));

        Cinema cineplexGraz = new Cinema("Cineplex Graz", "Alte Poststraße 470, 8055
Graz", LocalDate.parse("2004-06-17", formatter));
        halls = new ArrayList<>();
        em.persist(cineplexGraz);
        halls.add(new Hall("Saal 1", 247, cineplexGraz));
        halls.add( new Hall("Saal 2", 247, cineplexGraz));
        halls.add(new Hall("Saal 3", 184, cineplexGraz));
        em.persist(cineplexGraz);
        for (Hall hall : halls) em.persist(hall);
    }
}

```

Docker Container Starten

```
docker run --name some-postgres -p 5432:5432 -e POSTGRES_PASSWORD=passme -d postgres
```

Rest Service im Dev Mode starten:

```
mvnw compile quarkus:dev
```