

JPA mit Maven

1 leere Datenbank erstellen

```
mysql> CREATE DATABASE beispieldb;
```

```
mysql> CREATE USER 'benutzer'@'localhost';
```

```
mysql> ALTER USER 'benutzer'@'localhost' IDENTIFIED BY 'passwort';
```

```
mysql> GRANT ALL PRIVILEGES ON beispieldb.* TO 'benutzer'@'localhost';
```

2 Maven-Projekt erstellen

- maven-archetype-quickstart
- groupId: de.gbsschulen.jpabeispiel
- artifactId: jpabeispiel

3 Java 11 konfigurieren in pom.xml

Löschen

```
<maven.compiler.source>1.7</maven.compiler.source>  
<maven.compiler.target>1.7</maven.compiler.target>
```

Erweitern

```
<plugin>  
  <artifactId>maven-compiler-plugin</artifactId>  
  <version>3.8.0</version>  
  <configuration>  
    <release>11</release>  
  </configuration>  
</plugin>
```

4 Dependencies hinzufügen in pom.xml

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->  
<dependency>  
  <groupId>mysql</groupId>  
  <artifactId>mysql-connector-java</artifactId>  
  <version>8.0.18</version>  
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/org.eclipse.persistence/org.eclipse.persistence.jpa -->  
<dependency>  
  <groupId>org.eclipse.persistence</groupId>  
  <artifactId>eclipselink</artifactId>  
  <version>2.7.5</version>  
</dependency>
```

5 JPA-Konfigurationsdatei

src/main/resources/META-INF/persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="BeispielPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>de.gbsschulen.jpabeispiel.Person</class>
    <properties>
      <property name="javax.persistence.jdbc.url"
        value="jdbc:mysql://localhost:3306/beispieldb?serverTimezone=UTC"/>
      <property name="javax.persistence.jdbc.user"
        value="benutzer"/>
      <property name="javax.persistence.jdbc.password"
        value="passwort"/>
      <property name="eclipselink.ddl-generation"
        value="drop-and-create-tables"/>
    </properties>
  </persistence-unit>
</persistence>
```

6 de.gbsschulen.jpabeispiel.Person

```
package de.gbsschulen.jpabeispiel;

import javax.persistence.Entity;
import javax.persistence.Id;
import java.util.UUID;

@Entity
public class Person {
    @Id
    private String id;
    private String vorname;
    private String nachname;

    public Person(String vorname, String nachname) {
        this.id = UUID.randomUUID().toString();
        this.vorname = vorname;
        this.nachname = nachname;
    }

    public Person() { this("Max", "Mustermann"); }

    public String getVorname() { return vorname; }

    public void setVorname(String vorname) { this.vorname = vorname; }

    public String getNachname() { return nachname; }

    public void setNachname(String nachname) { this.nachname = nachname; }
}
```

7 Entity-Manager und Transaktion

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("BeispielPU");
EntityManager em = emf.createEntityManager();
em.getTransaction().begin();
```

```
// Hier CRUD-Aktionen ausführen
```

```
em.getTransaction().commit();
em.close();
emf.close();
```

8 CRUD-Operationen

CREATE

```
Person person = new Person();
em.persist(person);
```

READ

```
Person person = em.find(Person.class, "d4319f2b-ad8a-44bf-ac14-43d03d78920a"); // primary key
```

oder

```
Query query = em.createQuery("SELECT Person FROM Person person"); // JPQL
List<Person> list = query.getResultList();
for (Person person : list) {
    ... = person ...
}
```

UPDATE

```
person.setVorname("Anna");
```

DELETE

```
em.remove(person);
```