JPA-Grundlagen

1 Bibliotheken

- EclipseLink(JPA 2.0), MySQL JDBC-Treiber

2 MySQL-Datenbank

- Datenbank: `beispieldb`, Benutzer: `root`, Passwort: `mysql`
- Tabelle: `PERSON`, Spalten: `ID: VARCHAR`, `VORNAME: VARCHAR`, `NACHNAME: VARCHAR`

3 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>beispiel.Person</class>
    <properties>
      <property name="javax.persistence.jdbc.url"
                value="jdbc:mysql://localhost:3306/beispieldb"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.password" value="mysql"/>
    </properties>
  </persistence-unit>
</persistence>
```

4 beispiel/Person.java

```
@Entity
public class Person {
    @Id
    private String id;

    private String vorname;
    private String nachname;

    public Person() {
        this.id = UUID.randomUUID().toString();
    }

    public Person(String vorname, String nachname) {
        this();
        this.vorname = vorname;
        this.nachname = nachname;
    }

    // getter und setter  ...
}
```

5 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();
```

6 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);
```