Spring Data

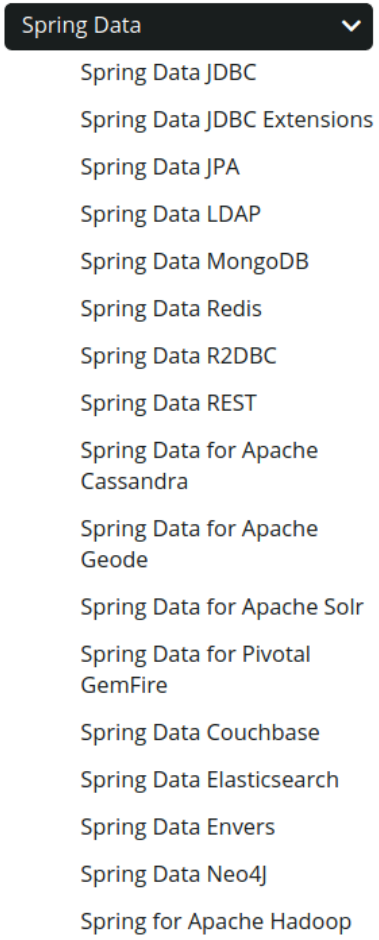
Projekt base.camp

Inhalt

- Was ist Spring Data?
- Einführung JPA
- Hibernate
- Spring Data JPA
- Demo
- Fazit

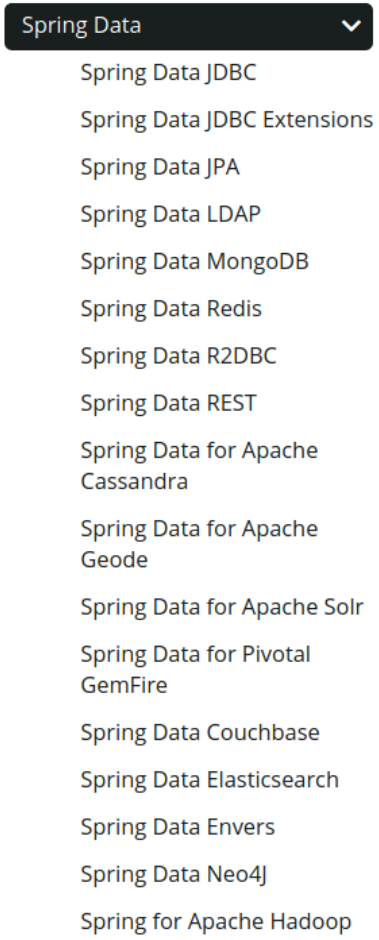
Was ist Spring Data? (I)

- Ziel: konsistentes, spring basiertes programmier Modell
- Enthält viele Subprojekte, die spezifisch für gegebene DBMS sind
  - Sql, NoSql, map reduce frameworks
  - z.B. JDBC, JPA, MongoDB, Cassandra
- CRUD Operationen und einige andere müssen nicht implementiert werden. Sie werden durch das Spring Data Repository geliefert.
- Verminderung von redundantem “boilerplate” Code



Was ist Spring Data? (II)

- Einfache Integration anderer Spring Frameworks
- Generierte Queries durch Schlüsselwort-Erkennung (z.B. in Spring Data JPA)
- Viele andere Features

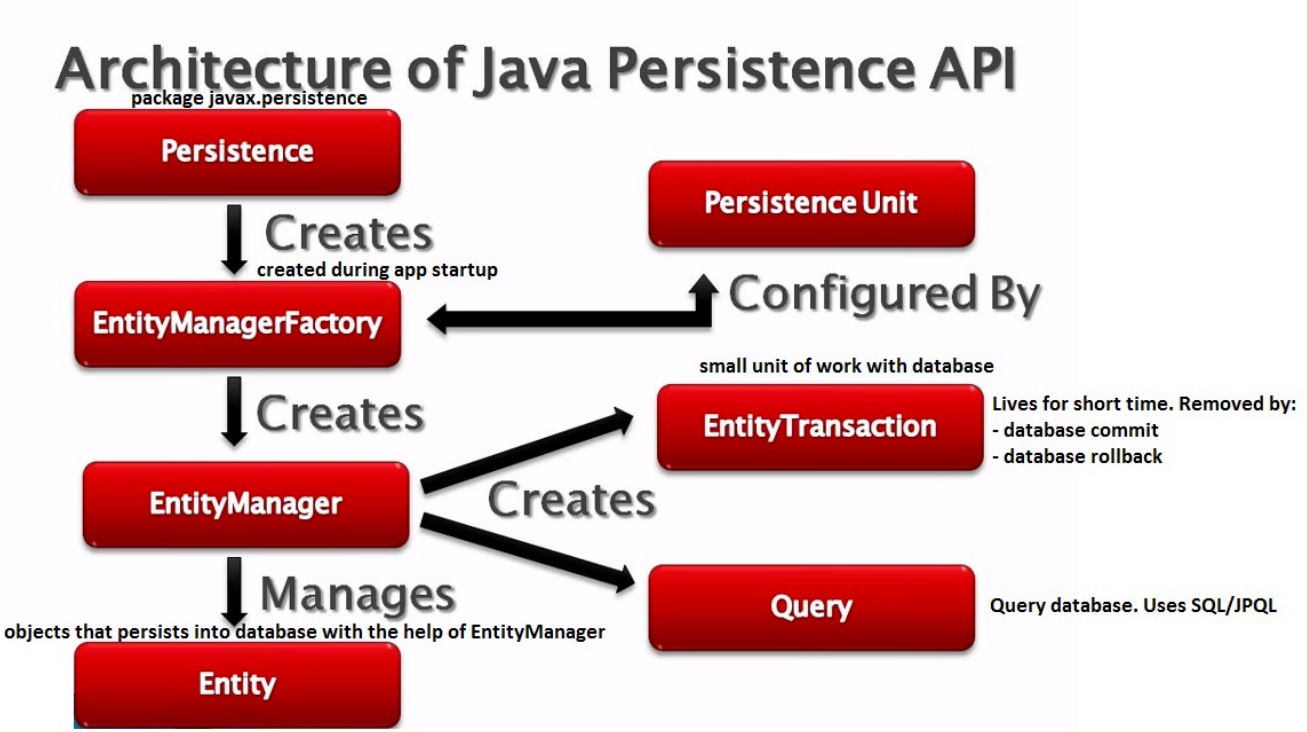


Was ist JPA? (I)

- JPA: Java Persistent API
- Problem: Programmierung ist objektorientiert, während die DBMS relational sind
- Für jede Tabelle in DBMS muss eine Entity POJO Klasse erzeugt werden
  - Mit diesen Objekten arbeitet man im Code
- API javax.persistance package
- Java Persistance Query Language (JPQL)
  - Sehr an SQL angelehnt
  - Arbeitet mit Entity-Objekten und nicht mit Relationen
- @OneToMany, @ManyToOne, @ManyToMany, @ManyToOne

Was ist JPA? (II)

- Implementation geschieht durch die JPA Provider
  - Hibernate, Eclipse Link, TopLink, ...
  - Jeder Provider bringt eigene Features mit sich



Hibernate

- Implementation von JPA
- Der populärste JPA Provider auf dem Markt
- Entity Mappings über XML
- Automatische Erzeugung des DB Schemas und der Tabellen möglich
- Eigene Sprache für Queries: HSQL
- ...

Spring Data JPA (I)

- Spring Data JPA ist kein JPA Provider
- Es arbeitet eine Abstraktionsebene über dem JPA Provider
- Man erzeugt eine Entity als POJO Klasse und dazu ein Repository, welches von “CrudRepository” erbt

```
public interface StudentRepository extends CrudRepository<Student, Long> {
}
```

- Spring Data JPA liefert die Methoden:
  - save(S entity), saveAll(Iterable<S> entities), findById(ID id), count(), delete(T entity), deleteAll(), ...

Spring Data JPA (II)

```java
public interface CustomerRepository extends CrudRepository<Customer, Long> {

    List<Customer> findByEmail(String email);

    List<Customer> findByDate(Date date);

    // custom query example and return a stream
    @Query("select c from Customer c where c.email = :email")
    Stream<Customer> findByEmailReturnStream(@Param("email") String email);

}
```

- Durch das findBy erzeugt Spring Data JPA eine Query, die Customer Objekte anhand der E-Mail findet
- Es sind aber auch Queries möglich: @Query Annotation
  - Diese Funktion ist vorallem bei komplexen Queries nötig

Spring Data JPA Demo

Hilfreiche Ressourcen

- Features von Spring Data JPA: https://thorben-janssen.com/what-is-spring-data-jpa-and-why-should-you-use-it/#repositories
- Gutes Beispiel, welches eine MySql DB benutzt: https://attacomsian.com/blog/accessing-data-spring-data-jpa-mysql
- Spring Initializr zur Erzeugung von Spring Data Projekten: https://start.spring.io/
- Offizieller Quickstart Guide von Spring Data. Es ist keine Datenbank Einrichtung nötig, da mit einer Inmemory Datenbank gearbeitet wird (gut zum Ausprobieren): https://spring.io/guides/gs/accessing-data-jpa/