

DEZSYS DOM

Projekteinführung

Diese Übung soll helfen die Funktionsweise und Einsatzmöglichkeiten eines dokumentenorientierten dezentralen Systems mit Hilfe des Frameworks Spring Data MongoDB oder einem Framework Ihrer Wahl zu demonstrieren. Die Daten werden in dieser Übung in einem NoSQL Repository gespeichert und verarbeitet.

Es handelt sich um ein Lagerstandort Beispiel, wie in Aufgabe "GK8.1 Spring Data and ORM". Die Daten aller Lagerstandorte sollen in der Zentrale persistiert und in einer NoSQL Datenbank gespeichert werden. Von hier aus koennen die Daten fuer verschiedene Fragestellungen des Betriebes (Management, Einkauf, Vertrieb,...) abgefragt werden.

Projektbeschreibung

In diesem Projekt werde ich ein Databanksystem aufsetzen, indem ich einen Online Shop programmiere, bei dem ich die grundlegenden CRUD Operationen mit REST und einer MySQL Datenbank umsetzen soll.

Theorie

Wir haben bereits die Theorie in den Stunden gelernt. Die Basic Implementierungen von CRUD Operationen sind mit einem Repository umzusetzen.

Arbeitsschritte

Als erstes habe ich mir das Projekt vom Herr Professor Micheler geklont. Von diesem Stand aus hatte ich bereits ein Repository und musste nur noch die erste Funktionalität aufsetzen. Das Repository sollte für beide ungefähr so aussehen:

```
public interface WarehouseRepository extends CrudRepository<Warehouse, Id>
```

Für die gegebene Datenstruktur muss ich eine eigene Klasse erstellen, die Getter und Setter und toString Methoden implementiert. Dabei habe ich von Haus aus einen User gegeben und muss nur noch Warehouse und Produkt hinzufügen. Für jedes davon erstelle ich dann auch noch ein Repository und kann danach schon an den CRUD Operationen arbeiten. Durch die Implementierung von CrudRepository kann ich grundlegende Methoden schon von Haus aus anwenden, wenn ich nur ein Attribut in meiner Controller Klasse erstelle. Dieses kann dann mit solchen Methoden benutzen:

```
@GetMapping("/products/{id}")
public @ResponseBody ProductData getProductDataById(@PathVariable String id) {
    return productRepository.findById(id).get();
}
```

Damit ich das auch wirklich in einem MongoDB Repository abspeichern kann, brauche ich zuerst eine Datenbank die funktioniert.

Als Erstes brauche ich den Container:

- `docker pull mongo`

Danach muss ich den Container, den ich gerade heruntergeladen habe, ausführen und starten, dafür benutze ich Port 27017 für In und Out.

```
docker run -d -p 27017:27017 --name mongo mongo
```

Wenn ich den Container starten will, muss ich jetzt nur noch diesen Befehl eingeben und kann schon auf die Datenbank zugreifen.

```
docker exec -it mongo bash
```

Befehle, die für die Datenbank wichtig sind, sind zb. `show dbs`

`localdb.startup_log.count();` oder `db.<name>.find().pretty()` .

Jetzt muss ich nur noch die Verbindung herstellen. Dafür schreibe ich

```
spring.data.mongodb.uri=mongodb://localhost:27017/mydatabase
```

in meine Application.properties Datei. Nun sollte ich meine ersten Daten generieren und ausgeben können. Jetzt sollten, nachdem ich die Verbindung hergestellt habe, die Funktion einwandfrei funktionieren.

Zusammenfassung

Ich kann jetzt auch MongoDB mit meinem Webshop verbinden und die gesamten Vorteile von MongoDB benutzen.

Quellen

<https://www.mongodb.com/docs/manual/reference/command/>