

## Arbeitsblatt 6 – Integration MongoDB

### Ziele

Sie haben die Datenbank MongoDB in ihre Applikation integriert und die Entitäten "Questionnaire" werden in dieser Datenbank persistent gespeichert.

### Ausgangslage

Sie haben das Arbeitsblatt 5 korrekt gelöst.

### Aufgabe 1: Entität "Questionnaire" erstellen

Die Klasse "Questionnaire" muss zu einer Datenbank-Entität umgeschrieben werden. Ersetzen sie "Questionnaire" durch den Inhalt aus Listing 1.

```
@Document(collection="questionnaires")
public class Questionnaire {
    @Id #1
    private String id; #2
    private String title;
    private String description;

    public void setId(String id) {
        this.id = id;
    }

    public String getId() {
        return id;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getTitle() {
        return title;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getDescription() {
        return description;
    }
}
```

Listing 1: Klasse "Questionnaire" als MongoDB Entität

### Hinweise:

#1: Mit dieser Annotation wird folgende Property als ID festgelegt

#2: Die ID einer MongoDB ist immer vom Typ String

### Aufgabe 2: Zugriff auf die MongoDB herstellen

Die Klasse "QuestionnaireRepository" abstrahiert den Zugriff auf die Datenbank MongoDB. Sie muss u.a. die CRUD Operationen bereitstellen, um Entitäten in der Datenbank zu erzeugen (Create), zu lesen (Read), zu ändern (Update) und zu löschen (Delete).

Das Spring Framework stellt dazu eine umfangreiche Unterstützung bereit, so dass der Programmieraufwand sehr gering wird.

Ersetzen sie die Klasse "QuestionnaireRepository" durch das Interface aus Listing 2.

```
public interface QuestionnaireRepository extends  
    MongoRepository<Questionnaire, String> {  
    List<Questionnaire> findByTitle(String title);  
}  
#1  
#2
```

Listing 2: "QuestionnaireRepository" neu als MongoRepository

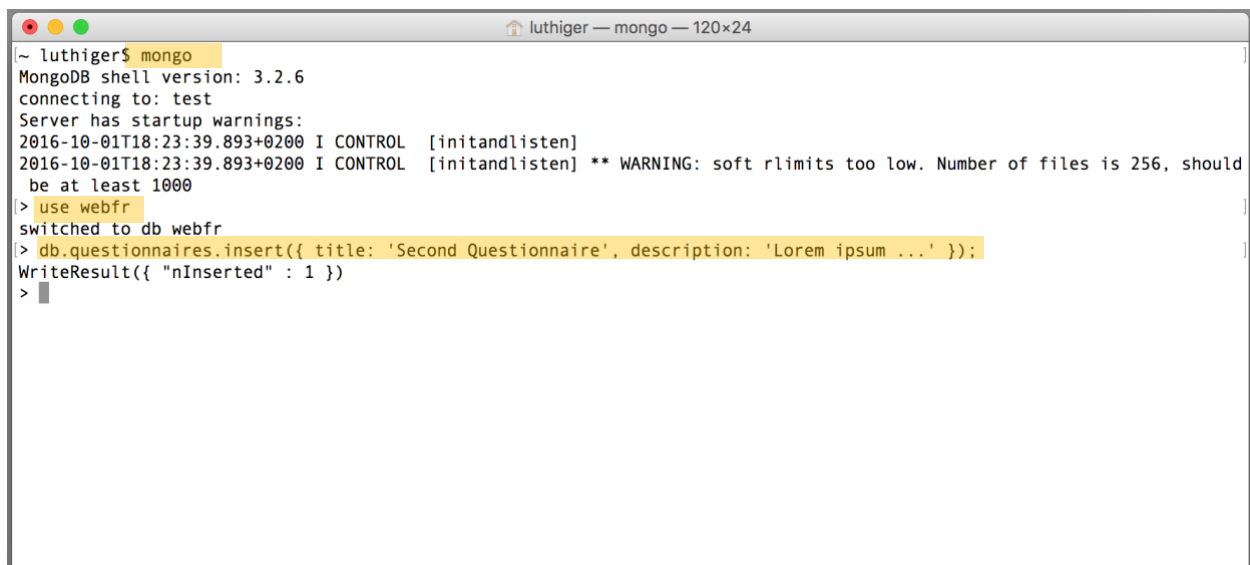
#### Hinweise:

#1: Das "QuestionnaireRepository" ist nun ein MongoRepository. Dabei muss die Entität, hier "Questionnaire", und der Typ der ID, hier "String", angegeben werden.

#2: Das MongoRepository stellt alle elementaren CRUD-Operationen automatisch bereit. Hier wird eine neue Methode definiert, um Questionnaire-Entitäten nach dem Titel suchen zu können.

#### Aufgabe 3: Mongo Entität erstellen

Für Testzwecke können sie über die mongo-Shell verschiedene Entitäten erstellen. Orientieren sich am Ablauf in der Abbildung 1, um eine solche Entität zu erzeugen.



```
luthiger$ mongo  
MongoDB shell version: 3.2.6  
connecting to: test  
Server has startup warnings:  
2016-10-01T18:23:39.893+0200 I CONTROL [initandlisten]  
2016-10-01T18:23:39.893+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000  
> use webfr  
switched to db webfr  
> db.questionnaires.insert({ title: 'Second Questionnaire', description: 'Lorem ipsum ...' });  
WriteResult({ 'nInserted' : 1 })  
>
```

Abbildung 1: Befehl in der mongo-Shell, um eine Questionnaire Entität zu erstellen

#### Hinweise:

- Weitere Befehle in der mongo-Shell  
Alle Questionnaire Entitäten auflisten  
> db.questionnaires.find();

Search Criteria einsetzen – über ID

> db.questionnaires.find( { \_id: ObjectId('57eff90493198b457f894f1c') } )

Search Criteria einsetzen – über Title

> db.questionnaires.find( { title:"Hallo" } )

#### Aufgabe 4: Webapplikation testen

Testen sie die Antwort auf folgende URL Requests:

**<http://localhost:8080/flashcard-mvc/questionnaires>**

Vergessen sie nicht die mongoDB zu starten!