Quick MongoDB with Java Spring

29.04.2022
Pawel Sosnowski

Prerequisites

- MongoDB running, exposed at port 27017.
- Database containing required collections.
- Collections containing required documents.
- Lombok
- (some JSON parser)

POM

```
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongodb-driver-sync</artifactId>
    <version>4.1.2</version>
</dependency>
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.12.8</version>
</dependency>
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>bson</artifactId>
    <version>4.1.2</version>
</dependency>
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-mongodb</artifactId>
    <version>3.1.6</version>
</dependency>
```

* Of course use newest versions

Creating entities

```
@Getter
@Setter
@ToString
@Document(collection = "places")
@AllArgsConstructor
public class Warehouse {
   @Id
   @Field("_id")
   private String id;

   @Field("name")
   private String name;

   @Field("location")
   private Coords coords;
}
```

Creating repositories

```
public interface WarehouseRepository extends MongoRepository<Warehouse, String> {
    Optional<Warehouse> findById(String id);
}
```

Introducing MongoDB to a Controller

```
@Controller
public class WarehouseController {

    @Autowired
    WarehouseRepository warehouseRepository;
    //...
```

Inserting JSON into MongoDB

```
@GetMapping("/insert_db")
@ResponseBody
@SuppressWarnings("unchecked")
public String GetWarehouseDataInsertedToDB(@RequestBody String body) {
    JSONParser jsonParser = new JSONParser();
    JSONArray jsonArray = new JSONArray();

    try {
        jsonArray.add(jsonParser.parse(body));
    } catch (ParseException e) {
        log.log(Level.SEVERE, "Failed to insert to database.");
        return "Failed";
    }

    try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {
        MongoDatabase theDB = mongoClient.getDatabase("car_sales");

        MongoCollection<Document> placesCollection = theDB.getCollection("places");

        for (Object jsonObject : (JSONArray) jsonArray.get(0)) {
            Document place = new Document();

            for (String key : Arrays.asList("_id", "name", "location")) {
                place.append(key, doced.get(key));
            }
            try {
                placesCollection.insertOne(place);
            } catch (MongoWriteException e) {
                log.log(Level.WARNING, "Duplicate record found: place, not inserting.");
            }
        }
    }
    return "Success";
}
```

Getting data from MongoDB

```
@GetMapping("/warehouse")
public ResponseEntity<Object> getWarehouse(
        @RequestParam() String id
) {
    try {
        Optional<Warehouse> warehouse = warehouseRepository.findById(id);
        if(warehouse.isEmpty()) {
            log.log(Level.WARNING, "Did not find warehouse with id: "+id);
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        }
        return new ResponseEntity<>(warehouse, HttpStatus.OK);
    } catch (Exception e) {
        e.printStackTrace();
        return new ResponseEntity<>(null, HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```