MongoDB desde Spring

DigitalHouse>

Certified Tech Developer
The Ultimate Degree

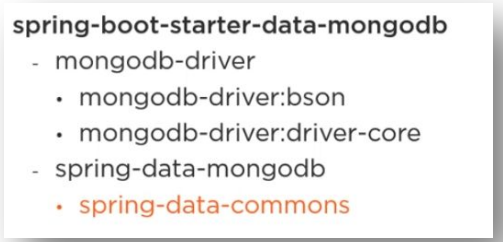
Dependencias

Con SpringBoot Initializer desde IntelliJ IDEA podemos crear un proyecto Spring con las siguientes dependencias en el pom.xml:

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>
```

spring-boot-starter-data-mongodb tiene todo lo necesario para crear un proyecto Springboot con Mongo.



Propiedades de conexión

En resources creamos un file con el nombre application.properties donde agregaremos la configuración.

```
#mongodb
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=mongoexample
```

Clases y estructuras del proyecto

La clase **Book** que va a mapear a la Collection books.

```
@Document(collection = “books”)
public class Book{

@Id
private String id;
private String author;
@Field(name = “book”)
private String bookTitle;
}
```

Clases y estructuras del proyecto

El **BookRepository** que extiende de MongoRepository.

```
@repository
public interface BookRepository extends
MongoRepository<Book,String> {
}
```

Clases y estructuras del proyecto

El **BookService** con el método findAllBooks.

```
@Service
public class BookService{

  private final BookRepository bookRepository;

  public BookService(BookRepository  bookRepository){
    this.bookRepository = bookRepository;
  }

  public List<Book> findAllBooks();
    return bookRepository.findAll();
}
```

Clases y estructuras del proyecto

El **Controller.**

```
@RestController
@RequestMapping(value = “/mongoexample”)
public class BookController{

  private final BookService bookService;

  public BookController(BookService bookService){
    this.bookService = bookService;
  }

  @GetMapping(value = “/books”)
  public List<Book> getAllBooks();
    return bookService.findAllBooks();
}
```

Queries especiales

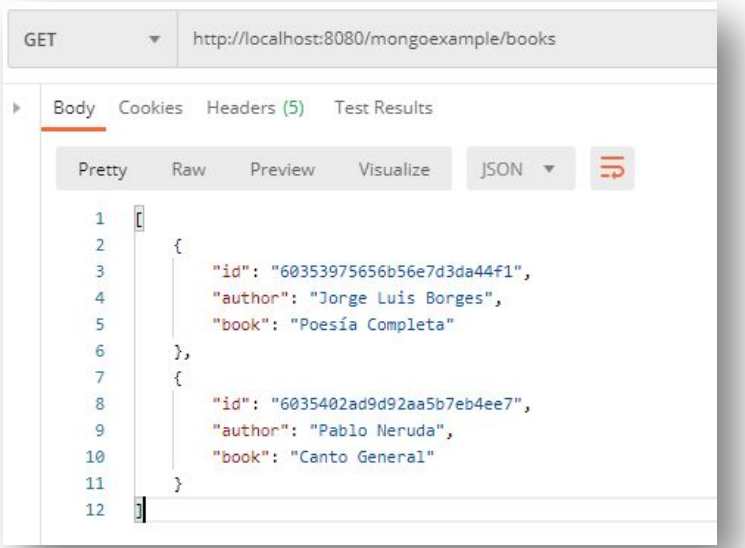
En el caso de querer realizar una query que tome un parámetro y filtre por el mismo, como por ejemplo findBooksByAuthor(), sencillamente se debe pasar el parámetro con el mismo nombre que posee el field en la BD, en este caso “author”, de la siguiente manera:

```
@Repository
public interface BookRepository extends MongoRepository
<Book,String>{
  List<Book> findBooksByAuthor(String author);
}
```

Probando la aplicación

Correr la aplicación con mvn spring-boot:run. Luego, desde el navegador o desde Postman realizar un GET a la url: http://localhost:8080/mongoexample/books

Realizará una request que producirá una query a nuestra BD de Mongo local y nos devolverá una response con todos los documents de la collection Books.



DigitalHouse>