Application Frameworks: Java | Lab Session 2

Spring Boot Application Setup

1. Go to https://start.spring.io/ and select the following for generating a project.
   - Project: Maven
   - Language: Java
   - Spring Boot: 2.1.4
   - Project Metadata: (Group: com.example, Artifact: demo)
   - Dependencies: Web, MongoDB, Actuator

Start MongoDB to start using that within the Spring Boot application. Use the following code to provide a different dbpath than the default if default path is not accessible.
`mongod --dbpath ../MongoDB_Path/`

2. Execute the "`mvn clean install`" command on the extracted directory.

3. Create a domain class named "`Message`" inside a package named "`com.example.demo.domain`" which has fields of following types and generate getter and setter methods for all those fields.

```
@Id
private String id;

@NotEmpty
private String message;

@JsonIgnore
private boolean deleted;

@CreatedDate
private Date createdAt;

@LastModifiedDate
private Date updatedAt;
```

Use the `@Document("message")` annotation to name the collection explicitly.

4. Add a controller class named "`MessageController`" inside a package named "`com.example.demo.controller`".

5. Create 5 methods in the controller for create a message, update a message, delete a message, retrieve a single message and to retrieve all messages. (Use the following annotations)

`@RestController`
`@RequestMapping()` [with "value" and "method" parameters]
`@PathVariable`
`@RequestBody`

ex:

```
@RestController
@RequestMapping(value = "/messages")
public class MessageController {
```

6. Add the following springfox swagger dependencies to the POM file and check whether the API documentation is generated.

```
<dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-swagger2</artifactId>
      <version>2.9.2</version>
</dependency>
<dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-swagger-ui</artifactId>
      <version>2.9.2</version>
</dependency>
```

Add the annotation `@EnableSwagger2` to the `DemoApplication` class containing the main method.

Use the command *`java -jar target/demo-0.0.1-SNAPSHOT.jar`* to run the application.

7. Add a repository interface named "`MessageRepository`" inside the package "`com.example.demo.repository`" which extends "`MongoRepository`" interface.

```
extends MongoRepository<Message, String>
```

8. Add a service class named "`MessageService`" inside the package "`com.example.demo.service`" which uses the following annotations.

`@Autowired` to inject an instance of MongoRepository.
`@Service` to mark the class as a service class.

Use the following methods of the repository instance.
`save, findAll, findById (this will be returning an entity of type Optional<Message>)`

ex: create method

```
public Message createMessage(Message message) {

      message.setId(UUID.randomUUID().toString());
      message.setDeleted(false);
      message.setCreatedAt(Calendar.getInstance().getTime());

      return messageRepository.save(message);
}
```

** Delete method should be a soft delete where the entry is not deleted from the DB. The "`deleted`" boolean flag can be used to implement this.

9. Inject an instance of the created service to the `MessageController` class and save the entries to the database using the injected service.

10. Modify the single entry retrieval method to comply with the following structure.

```
@RequestMapping(value = "/{id}", method = RequestMethod.GET)
public HttpEntity<Message> getMessage(@PathVariable("id") String id) {

	Optional<Message> messageOpt = messageService.getMessage(id);

	if (messageOpt.isPresent()) {
		return new ResponseEntity<Message>(messageOpt.get(), HttpStatus.OK);
	} else {
		return new ResponseEntity<Message>(HttpStatus.NOT_FOUND);
	}
}
```

11. Set a name for the database by adding the following entry to the application.properties so that it won't take the default db name.

```
spring.data.mongodb.database=messagedb
```

12. Set a different port for the application to run by using the following configuration entry.

```
server.port=8081
```