

Лабораторная работа №5.

Spring Boot, Spring Data JPA, страницы Thymeleaf, REST API

Часть I

1. Создать Spring Boot проект.

Для этого можно использовать Spring Initializr (<https://start.spring.io/>)

Добавить в проект зависимости:

Spring Web
Spring Data JPA
MySQL driver
Thymeleaf

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

2. Использовать в проекте классы-сущности из лабораторной №2

Создать интерфейсы репозитории - наследники интерфейса CrudRepository.

В файле application.properties задать конфигурацию сервера и используемой БД:

```
server.servlet.context-path=/DemoSpring
server.port=8082
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://${localhost}:3306/db_example
spring.datasource.username=springuser
spring.datasource.password=ThePassword
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.generate-ddl=true
#spring.jpa.show-sql: true
```

3. Создать классы – Spring контроллеры (@Controller) в соответствии со своей предметной областью. В качестве пользовательского интерфейса использовать страницы Thymeleaf. Spring Boot обеспечивает автоматическую настройку Thymeleaf. Для этого достаточно добавить зависимость spring-boot-starter-thymeleaf.

Запустить проект можно из корня приложения maven командой
mvn spring-boot:run

Часть2 (дополнительно задание)

1. Создать класс – REST контроллер (@RestController) для сущности из лабораторной работы №1 (Angular). Реализовать базовые методы получения, добавления, удаления данных.

2. Добавить в Spring REST контроллер аннотацию @CrossOrigin(origins = "http://localhost:4200") для того, чтобы разрешить междоменные запросы.

В Angular приложении лабораторной работы №1 внести изменения в класс service.ts:

Импортировать модули:

```
import { HttpClient, HttpHeaders } from '@angular/common/http';
```

Получать/изменять данные для приложения с помощью http запросов. Пример метода, возвращающего список книг по HTTP GET запросу:

```
bookUrl ='http://localhost:8080/books';
```

```
getBooks(): Observable<Book[]> {  
    return this.httpClient.get<Book[]>(this.bookUrl); }
```

	Срок выдачи задания:	Срок сдачи задания:
12 группа	11.12.23	21.12.23
13 группа	11.12.23	18.12.23

Номер варианта согласовывается с преподавателем

Максимальное количество баллов за работу– 20 (часть1) и 20 (часть2).

Номер окончательной версии:

1 версия – коэффициент 1, 2 версия – коэффициент 0,6, 3 версия – коэффициент 0,4.

Дата сдачи

До указанной даты – коэффициент 1, опоздание до 1 недели – коэффициент 0,6, позже – коэффициент 0,4.