Домашнее задание:

Цель практической работы

Закрепить на практике навыки работы со Spring Framework при создании веб-приложения с API для управления новостной лентой.

Что входит в практическую работу

1. Разработать веб-приложение с использованием Spring Framework.
2. Создать API для управления новостной лентой путём выполнения REST-запросов методами GET, POST, PUT и DELETE, при которых будет происходить:
    - получение одной новости по её идентификатору (getById);
    - получение списка всех новостей (getAll);
    - создание новой новости (create);
    - обновление существующей новости (update);
    - удаление новости по её идентификатору (deleteById).

Что нужно сделать

1. Создайте веб-приложение с использованием Spring Boot. Настройте проект и добавьте зависимость <spring-boot-starter-web>.
2. Создайте сущность News для отображения новостей в вашем приложении. Эта сущность должна содержать следующие поля:
    - id (класс Long) — уникальный идентификатор новости.
    - title (класс String) — заголовок новости.
    - text (класс String) — текст новости.
    - date (класс Instant) — дата и время публикации новости.
3. Создайте контроллер API для управления новостной лентой. Этот контроллер должен предоставлять методы для выполнения операций над новостями. API должно соответствовать следующим эндпоинтам:
    - **Получение одной новости по её идентификатору:**
      URL: /api/news/{id}
      HTTP-метод: GET
      Описание: Получение информации о новости по её уникальному идентификатору.
      Параметры пути:

id (Long): Уникальный идентификатор новости.
Ответ:
Код HTTP-ответа 200 (OK) и объект News в формате JSON с информацией о новости в следующем формате:

```
{
  "id": 1,
  "title": "Заголовок новости",
  "text": "Текст новости",
  "date": "2023-10-20T12:00:00Z"
}
```

Код HTTP-ответа 404 (Not Found), если новость с указанным ID не найдена.

```
{
 "message": "Новость с ID 1 не найдена."
}
```

- **Получение списка всех новостей:**
  URL: /api/news
  HTTP-метод: GET
  Описание: Получение списка всех новостей в формате JSON.
  Ответ:
  Код HTTP-ответа 200 (OK) и массив объектов News с информацией о новостях в следующем формате:

```
[
 {
  "id": 1,
  "title": "Заголовок новости 1",
  "text": "Текст новости 1",
  "date": "2023-10-20T12:00:00Z"
 },
 {
  "id": 2,
  "title": "Заголовок новости 2",
  "text": "Текст новости 2",
  "date": "2023-10-21T10:30:00Z"
 }
]
```

- **Создание новой новости:**
  URL: /api/news
  HTTP-метод: POST
  Описание: Создание новой новости.
  Запрос:Объект News в формате JSON с заполненными полями title, text и date:

{
"title": "Новая новость",
"text": "Текст новости",
}

Ответ:Код HTTP-ответа 201 (Created) и объект News в формате JSON с информацией о созданной новости:

{
"id": 3,
"title": "Новая новость",
"text": "Текст новости",
"date": "2023-10-22T14:45:00Z"
}

- **Обновление существующей новости:**
  URL: /api/news
  HTTP-метод: PUT
  Описание: Обновление существующей новости.
  Запрос:Объект News в формате JSON с заполненными полями id, title, text и date.

{
"id": 3,
"title": "Обновлённая новость",
"text": "Обновлённые новости",
}

Ответ:Код HTTP-ответа 200 (OK) и объект News в формате JSON с информацией об обновлённой новости:

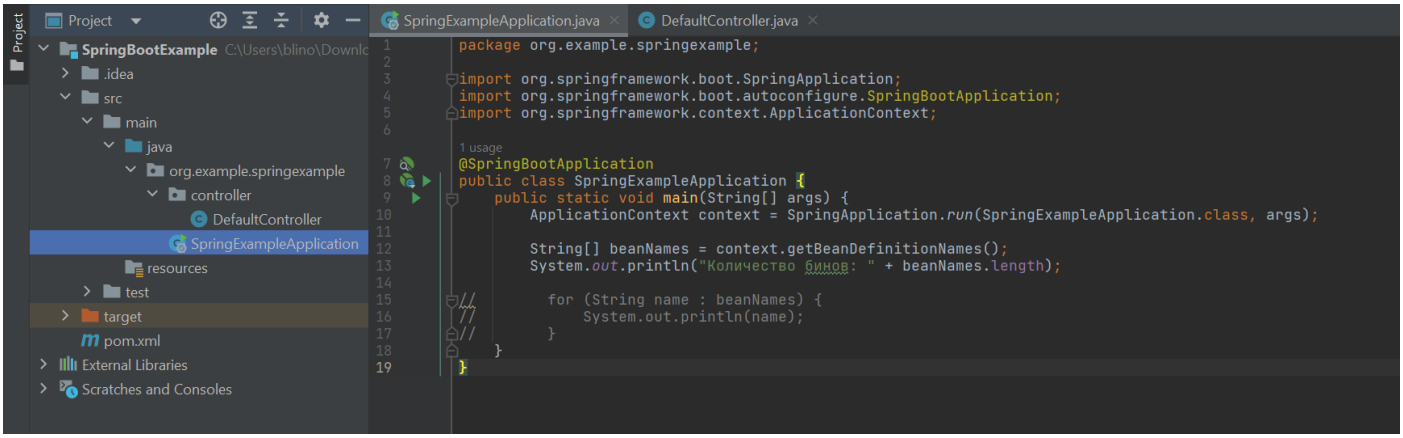
{
"id": 3,
"title": "Обновлённая новость",
"text": "Обновлённые новости",
"date": "2023-10-22T14:55:00Z"
}

- **Удаление новости по её идентификатору:** URL: /api/news/{id}
  HTTP-метод: DELETE
  Описание: Удаление новости по её уникальному идентификатору.
  Параметры пути: id (Long): Уникальный идентификатор новости.
  Ответ:
  Код HTTP-ответа 204 (No Content) в случае успешного удаления новости.
  Код HTTP-ответа 404 (Not Found), если новость с указанным ID не найдена, и ответ в формате:

{
"message": "Новость с ID 3 не найдена."
}

4. Создайте сервис для управления новостной лентой. В этом сервисе используйте ConcurrentHashMap для хранения данных о новостях. Сервис должен предоставлять следующие методы:
    - getById — получение новости по её идентификатору.
    - getAll — получение списка всех новостей.
    - create — создание новой новости.
    - update — обновление существующей новости.
    - deleteById — удаление новости по её идентификатору.
5. Для проверки работы приложения можете использовать приложение Postman и создать запросы к вашему приложению. Используйте приведённые в примерах JSON для тела ваших запросов.

Проект с занятия:



DefaultController:

```java
package org.example.springexample.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.TreeMap;

@RestController // = @ResponseBody + @Controller
public class DefaultController {

    private TreeMap<Integer, String> todo;

    public DefaultController() {
        todo = new TreeMap<>();
        todo.put(1, "Купить молоко!");
        todo.put(2, "Покормить кота!");
    }

    @GetMapping(path =  "/todo")
    public TreeMap<Integer, String> get() {
        return todo;
    }

    @PostMapping(path = "/todo")
    public ResponseEntity post(@RequestParam String item) {
        if (item == null || item.isEmpty()) {
            return new ResponseEntity(HttpStatus.BAD_REQUEST);
        }

        todo.put(todo.lastKey() + 1, item);
        return new ResponseEntity(HttpStatus.CREATED);
    }

    @PutMapping(path = "/todo/{id}")
    public ResponseEntity put(@PathVariable int id, @RequestParam String item) {
        todo.put(id, item);
        return new ResponseEntity(HttpStatus.OK);
    }

    @DeleteMapping(path = "/todo/{id}")
    public ResponseEntity put(@PathVariable int id) {
        todo.remove(id);
        return new ResponseEntity(HttpStatus.OK);
    }
}
```

POM файл:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example.springexample</groupId>
    <artifactId>SpringBootExample</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.3</version>
    </parent>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

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

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <version>3.1.3</version>
            </plugin>
        </plugins>
    </build>

</project>
```