#### 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:
   1. получение одной новости по её идентификатору (getById);
   2. получение списка всех новостей (getAll);
   3. создание новой новости (create);
   4. обновление существующей новости (update);
   5. удаление новости по её идентификатору (deleteById).

#### Что нужно сделать

Выполните задание в репозитории java\_basics в проекте SpringBootApplication:

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:
   1. id (класс Long) — уникальный идентификатор новости.
   2. title (класс String) — заголовок новости.
   3. text (класс String) — текст новости.
   4. 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:
   1. 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."   
}

* 1. 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"   
  }  
]

* 1. 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",   
}

* 1. ****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"  
}

* 1. 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",   
}

* 1. ****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"  
}

* 1. 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."   
}

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

#### Что оценивается

* Проект Spring Boot создан и настроен согласно вводным данным задания.
* Создан контроллер API с методами для выполнения операций над новостями (GET, POST, PUT, DELETE).
* Создан сервис, использующий ConcurrentHashMap для хранения данных о новостях и предоставляющий методы getById, getAll, create, update и deleteById.
* Весь перечисленный функционал работает через API: новости создаются, выводятся по ID и общим списком, изменяются и удаляются.

#### Критерии оценки

****Принято:**** методы API должны корректно выполнять операции и возвращать соответствующие ответы и статусы HTTP. Выполнены все пункты задания.

****На доработку:**** задание не выполнено или выполнено неточно.