

Техническое задание для проекта Blog Platform

Введение

Проект Blog Platform предназначен для управления авторами и статьями. В рамках этого проекта реализовать REST API для создания, удаления и изменения авторов и статей, а также для получения статей определенного автора.

Цели

1. Обеспечить возможность создания, удаления и изменения авторов.
2. Обеспечить возможность создания, удаления и изменения статей.
3. Обеспечить возможность получения всех статей и статей определенного автора.

Технологии

- Java Development Kit (JDK) 17 or later
- Apache Maven 3.9.6
- Spring Framework (Spring Boot)
- REST API

Функциональные требования

1. Управление авторами

Написать следующие CRUD операции над сущностью автор.

1.1 Создание автора

- **Метод:** POST
- **URL:** /authors/createAuthor
- **Параметры:**
 - nickname (String): никнейм автора
- **Описание:** Создает нового автора.
- **Ответ:** Возвращает объект с информацией о созданном авторе.

1.2 Удаление автора

- **Метод:** DELETE
- **URL:** /authors/deleteAuthor
- **Параметры:**
 - nickname (String): никнейм автора
- **Описание:** Удаляет автора по никнейму.
- **Ответ:** Возвращает объект с информацией о результате удаления.

1.3 Получение статей автора

- **Метод:** GET
- **URL:** /authors/getAuthorArticles
- **Параметры:**
 - nickname (String): никнейм автора

- **Описание:** Возвращает список статей определенного автора.
- **Ответ:** Возвращает список статей.

1.4 Изменение никнейма автора

- **Метод:** PATCH
- **URL:** /authors/changeNickname
- **Параметры:**
 - `oldNickname (String)`: старый никнейм автора
 - `newNickname (String)`: новый никнейм автора
- **Описание:** Изменяет никнейм автора.
- **Ответ:** Возвращает объект с информацией о результате изменения.

2. Управление статьями

Написать следующие CRUD операции над сущностью статья.

2.1 Создание статьи

- **Метод:** POST
- **URL:** /authors/createArticle
- **Параметры:**
 - `nickname (String)`: никнейм автора
 - `title (String)`: заголовок статьи
- **Описание:** Создает новую статью для автора.
- **Ответ:** Возвращает объект с информацией о созданной статье.

2.2 Удаление статьи

- **Метод:** DELETE
- **URL:** /authors/deleteArticle
- **Параметры:**
 - `nickname (String)`: никнейм автора
 - `title (String)`: заголовок статьи
- **Описание:** Удаляет статью автора по заголовку.
- **Ответ:** Возвращает объект с информацией о результате удаления.

2.3 Изменение заголовка статьи

- **Метод:** PATCH
- **URL:** /authors/changeArticle
- **Параметры:**
 - `nickname (String)`: никнейм автора
 - `oldTitle (String)`: старый заголовок статьи
 - `newTitle (String)`: новый заголовок статьи
- **Описание:** Изменяет заголовок статьи автора.
- **Ответ:** Возвращает объект с информацией о результате изменения.

3. Получение всех статей

3.1 Получение всех статей

- **Метод:** GET
- **URL:** /article/getAllArticles
- **Описание:** Возвращает список всех статей.
- **Ответ:** Возвращает список статей.

Обработчик ошибок

- **Общие требования**

1. Реализовать централизованный обработчик ошибок для перехвата всех исключений.
2. Обработчик должен возвращать понятные и информативные сообщения об ошибках клиенту.
3. Сообщения об ошибках должны соответствовать стандарту HTTP-ответов и содержать статус ошибки, описание и дополнительную информацию.

- **Компоненты обработки ошибок**

- **ExceptionHandler:** Класс, отвечающий за перехват и обработку исключений.
- **CustomException:** Пользовательские исключения для специфических ошибок.
- **ResponseError:** Структура ответа с информацией об ошибке.

Тестирование

- Юнит-тесты должны охватывать 100% методов всех сервисов проекта.
- Для написания тестов использовать JUnit и Mockito
- Тесты должны автоматически запускаться при сборке проекта.

Использование Sonar Cloud

- **Общие требования**

- Регулярный анализ кода с использованием SonarCloud.
- Достижение нулевого уровня критических и блокирующих проблем.
- Сведение к минимуму кодовых запахов (code smells).

Заключение

Данное техническое задание охватывает основные аспекты управления авторами и статьями в системе Blog Platform. Включает требования к функционалу, нефункциональные требования и дополнительные аспекты, которые необходимо учесть при реализации проекта.