

## Техническое задание: Social Media API

Цель проекта: Разработать RESTful API для социальной медиа платформы, позволяющей пользователям регистрироваться, входить в систему, создавать посты, переписываться, подписываться на других пользователей и получать свою ленту активности.

### Требования:

#### 1. Аутентификация и авторизация:

- Пользователи могут зарегистрироваться, указав имя пользователя, электронную почту и пароль.
- Пользователи могут войти в систему, предоставив правильные учетные данные.
- API должен обеспечивать защиту конфиденциальности пользовательских данных, включая хэширование паролей и использование JWT.

#### 1. Управление постами:

- Пользователи могут создавать новые посты, указывая текст, заголовок и прикрепляя изображения.
- Пользователи могут просматривать посты других пользователей.
- Пользователи могут обновлять и удалять свои собственные посты.

#### 2. Взаимодействие пользователей:

- Пользователи могут отправлять заявки в друзья другим пользователям. С этого момента, пользователь, отправивший заявку, остается подписчиком до тех пор, пока сам не откажется от подписки. Если пользователь, получивший заявку, принимает ее, оба пользователя становятся друзьями. Если отклонит, то пользователь, отправивший заявку, как и указано ранее, все равно остается подписчиком.
- Пользователи, являющиеся друзьями, также являются подписчиками друг на друга.

- Если один из друзей удаляет другого из друзей, то он также отписывается.

Второй пользователь при этом должен остаться подписчиком.

- Друзья могут писать друг другу сообщения (реализация чата не нужна, пользователи могут запросить переписку с помощью запроса а)

### 3. Подписки и лента активности:

- Лента активности пользователя должна отображать последние посты от пользователей, на которых он подписан.

- Лента активности должна поддерживать пагинацию и сортировку по времени создания постов.

### 4. Обработка ошибок:

- API должно обрабатывать и возвращать понятные сообщения об ошибках при неправильном запросе или внутренних проблемах сервера.

- API должно осуществлять валидацию введенных данных и возвращать информативные сообщения при неправильном формате

### 5. Документация API:

- API должно быть хорошо задокументировано с использованием инструментов, таких как Swagger или OpenAPI.

- Документация должна содержать описания доступных эндпоинтов, форматы запросов и ответов, а также требования к аутентификации.

### Технологии и инструменты:

- Язык программирования: Java

- Фреймворк: Spring (рекомендуется использовать Spring Boot)

- База данных: Рекомендуется использовать PostgreSQL или MySQL

- Аутентификация и авторизация: Spring Security

- Документация API: Swagger или OpenAPI

### Ожидаемые результаты:

- Разработанное RESTful API, способное выполнять указанные требования.

- Код проекта, хорошо структурированный и документированный.

- Тесты, покрывающие основные функциональные возможности API.
- Документация API, описывающая доступные эндпоинты и их использование.

Некоторые факторы, влияющие на оценку:

- Полнота соблюдения требований;
- Масштабируемость кода и соблюдение важных принципов разработки;
- Читаемость кода.