

# Разработка Системы Управления Задачами

## Описание задачи:

Вам необходимо разработать простую систему управления задачами (Task Management System) с использованием Java. Система должна обеспечивать создание, редактирование, удаление и просмотр задач. Каждая задача должна содержать заголовок, описание, статус (например, "в ожидании", "в процессе", "завершено") и приоритет (например, "высокий", "средний", "низкий"), а также автора задачи и исполнителя. Реализовать необходимо только API.

## Требования:

1. Сервис должен поддерживать аутентификацию и авторизацию пользователей по email и паролю.
2. Доступ к API должен быть аутентифицирован с помощью JWT токена.
3. Пользователи могут управлять своими задачами: создавать новые, редактировать существующие, просматривать и удалять, менять статус и назначать исполнителей задачи.
4. Пользователи могут просматривать задачи других пользователей, а исполнители задачи могут менять статус своих задач.
5. К задачам можно оставлять комментарии.
6. API должно позволять получать задачи конкретного автора или исполнителя, а также все комментарии к ним. Необходимо обеспечить фильтрацию и пагинацию вывода.
7. Сервис должен корректно обрабатывать ошибки и возвращать понятные сообщения, а также валидировать входящие данные.
8. Сервис должен быть хорошо задокументирован. API должен быть описан с помощью Open API и Swagger. В сервисе должен быть настроен Swagger UI. Необходимо написать README с инструкциями для локального запуска проекта. Дев среду нужно поднимать с помощью docker compose.
9. Напишите несколько базовых тестов для проверки основных функций вашей системы.
10. Используйте для реализации системы язык Java 17+, Spring, Spring Boot. В качестве БД можно использовать PostgreSQL или MySQL. Для реализации аутентификации и авторизации нужно использовать Spring Security. Можно использовать дополнительные инструменты, если в этом есть необходимость (например кэш).

**Оценка:** Оцениваться будут следующие аспекты:

1. Соответствие требованиям.
2. Качество и чистота кода.
3. Проектирование системы и использование ООП.

4. Наличие тестов и их покрытие.
5. Обработка ошибок.