

Система управления сотрудниками RESTfull сервис.

Архитектура - 3 уровневая с разделением по слоям - Controller, Service, Repository. Стараться избегать дублирования кода.

Технологии:

Java 21, Spring Boot, Swagger, Spring Data, PostgreSQL, Flyway, Maven, Mapstruct, Lombok.

Авторизация любая

Скорость выполнения задания (без ущерба качеству кода и требованиям задачи), качество кода - отсутствие дублирования кода, общие вещи выносить

в отдельные компоненты или базовые классы, чистый читаемый код, работоспособность приложения и возможность легко запустить/проверить, разделение кода на небольшие части, отсутствие спагетти-кода, применение паттернов, следование SOLID принципам.

Необходимо реализовать пример бэкенд-приложения для системы управления

сотрудниками (employee management system).

Описание:

В системе ведется учет сотрудников. У каждого сотрудника есть фотография, статус и менеджер (это минимальный набор, можно произвольных атрибутов добавить на свое усмотрение). Каждый сотрудник состоит в нескольких отделах(группах). Должна быть возможность создать/отредактировать/удалить

информацию о сотруднике (включая то, в каких отделах состоит сотрудник и возможность смены фото сотрудника). Должна быть возможность найти сотрудника по имени сотрудника, статусу сотрудника, имени менеджера и

названиям отделов.

Должна быть 2 роли:

администратор и простой пользователь. Простой пользователь имеет права только на просмотр. Админ имеет неограниченные права.

- + реализовать spring-boot приложение (только бэк, REST API, без фроненда)

- + в корне репозитория создать docker-compose файл для запуска БД - приложение при запуске должно подключаться к данной бд

- (+ если для запуска нужны доп сервисы - добавить их тоже в файл)

- + написать краткую инструкцию по запуску приложения в файл readme в корне репозитория

Общие требования к API:

- + Необходимо наличие валидации вводимых данных. Ответ API должен содержать информацию почему запрос не валиден.

- + Удаление объектов сделать через soft delete (поставить флаг) и в эндпоинтах отдавать все не удаленные.

- + Исходные/тестовые данные необходимо задать в виде миграций Flyway.

- + В эндпоинте получения списка сотрудников необходимо наличие:

- + фильтрации по нескольким критериям одновременно (поиск по имени, статусу сотрудника, возможность найти сотрудников нескольких конкретных отделов одновременно) - реализовать любым из способов (например criteriaAPI/specifications/queryDSL/Jooq/...)

Учесть возможность получения пользователей как без

фильтрации, так и с частично/полностью переданными фильтрами в эндпоинт

- + сортировки (по произвольному полю)
- + постраничного просмотра (Pagination)