СПРИНТ 6

Разработка API на Python

Задание

Пусть имеются следующие сущности (модели).

Резервуар (таблица tanks)

- id (int)
- name (str)
- max_capacity (float)
- current_capacity (float)
- product_id (int) – id продукта
- created_at (datetime)
- created_by (int) – id пользователя, который добавил объект
- modifed_at (datetime)
- modifed_by (int) – id пользователя, который обновил информацию об объекте в БД

Операция (таблица operations)

- id (int)
- mass (float) – масса операции в тоннах
- date_start (datetime) – время начала операции
- date_end (datetime) – время окончания операции
- tank_id (int) – id резервуара
- product_id (int) – id продукта
- created_at (datetime)
- created_by (int) – id пользователя, который добавил объект
- modifed_at (datetime)
- modifed_by (int) – id пользователя, который обновил информацию об объекте в БД

Продукт (таблица products)

- id (int)

- name (str)
- created_at (datetime)
- created_by (int) – id пользователя, который добавил объект
- modifed_at (datetime)
- modifed_by (int) – id пользователя, который обновил информацию об объекте в БД

Пользователь (таблица users)

- id (int)
- username (str)
- password_hashed (str)
- role (str, одно из значений «admin», «viewer») – роль пользователя
- created_at (datetime)
- created_by (int) – id пользователя, который добавил объект
- modifed_at (datetime)
- modifed_by (int) – id пользователя, который обновил информацию об объекте в БД

Задание (25 баллов).

Требуется разработать приложение API, позволяющее управлять (получать, добавлять, обновлять, удалять) описанными сущностями. При этом необходимо выполнить следующие условия:

- Схема базы данных должна быть создана в результате **создания и выполнения миграций** (количество созданных миграций в данном задании не имеет значения).
- Для каждой сущности необходимо создать **отдельный роутер**, в котором обязательно должны быть **5 основных методов** (получить все объекты, получить один объект по ключу, добавить объект, обновить объект, удалить объект); если запрашиваемый объект не найден, возвращать статус 404; при успешном добавлении объекта – статус 201; при успешном удалении объекта – статус 204.

- Значения полей created_at, created_by должны **устанавливаться автоматически** при создании каждого объекта. Аналогично значения полей modified_at и modified_by устанавливаются при модификации объектов.
- Сделать так, чтобы при запуске приложения в таблицу users **добавлялся пользователь с ролью администратора** (если его нет в таблице). Данные для входа администратора (имя и пароль) брать из файла env.
- Создавать, обновлять, удалять и просматривать других пользователей **может только пользователь с ролью администратора**. При попытке выполнения такого действия рядовым пользователем необходимо возвращать статус 403 (Forbidden).
- Выполнение всех действий в системе **доступно только авторизованным пользователям**. При попытке выполнения запросов без авторизации возвращать статус 401.
- В роутер для операций добавить дополнительный эндпоинт (GET), который принимает 4 параметра (tank_id, product_id, date_start и date_end). Необходимо реализовать запрос, который **формирует (и отдает на скачивание) отчет в формате csv**. При формировании отчета выбирать операции по совпадению tank_id и product_id. При этом диапазон [date_start, date_end] (из передаваемых в метод параметров) должен находиться строго внутри диапазона [date_start, date_end] у операции.
- В методе, используемом для получения всех операций, необходимо не только выводить значения tank_id и product_id, но и соответствующие объекты (резервуар и продукт). **В результате должны получиться вложенные объекты.**
- В роутер для резервуаров добавить метод GET **для изменения значения current_capacity у резервуара** (в метод передается id резервуара, новое значение current_capacity). В случае успеха возвращать объект резервуара с обновленным значением current_capacity. Если резервуар не найден, возвращать статус 404.

- В роутер для операций добавить метод GET **для получения списка всех операций по данному резервуару**. Проверять наличие в системе резервуара, в случае его отсутствия возвращать статус 404.
- Конфигурирование проекта должно выполняться **только через файл .env в комбинации с объектом настроек** (settings на основе BaseSettings из pydantic).
- Сервисы должны внедряться в роутеры посредством **механизма внедрения зависимостей**. Сессия должна внедряться в сервисы также посредством механизма внедрения зависимостей.
- **Все методы должны быть задокументированы** (должно быть задано свойство name в декораторе, большее – по вашему желанию). У всех тегов (разделов) Swagger UI должно присутствовать описание.

Задание не предполагает субъективной оценки кода. Выполнил – получил баллы. Не выполнил – не получил.

В случае наличия грубых ошибок, на них будет обращено внимание.