Тебе необходимо создать бэкенд для работы со складом рулонов металла.

Это задание нужно выполнить на Python.

DTO рулона:

* id;
* длина;
* вес;
* дата добавления;
* дата удаления.

Обязательные пункты:

1. Routes (base /api):

1.1. POST /coil.

Добавление нового рулона на склад. Длина и вес обязательно должны быть

заданы.

В случае успеха возвращает назначенный id рулона.

1.2. DELETE /coil.

Удаление рулона с указанным id со склада.

1.3. GET /coil.

Получение списка рулонов со склада по указанному диапазону id / веса / длины /

даты добавления / даты удаления со склада.

1.4. GET /coil/stats.

Получение статистики по рулонам за определённый период:

* количество добавленных рулонов;
* количество удалённых рулонов;
* средняя длина, вес рулонов, находившихся на складе в этот период;
* максимальная и минимальная длина и вес рулонов, находившихся на складе в этот период;
* суммарный вес рулонов на складе за период;
* максимальный и минимальный промежуток между добавлением и удалением рулона.

2. Данные по рулонам должны храниться в базе данных (желательно

PostgreSQL/SQLite).

3. Также должны быть обработаны стандартные кейсы ошибок (например, недоступна

БД, не существует рулон при какой-то работе с ним).

4. Используемый стек: FastAPI, SQLAlchemy, pydantic (версии и до 2.0, и после 2.0

подойдут).

Бонусные баллы:

1. GET /coil берёт на вход комбинацию диапазонов.

2. GET /coil/stats дополнительно возвращает:

2.1. День, когда на складе находилось минимальное и максимальное количество

рулонов за указанный период.

2.2. День, когда суммарный вес рулонов на складе был минимальным и

максимальным в указанный период.

3. Проект должен быть обёрнут в Docker.

4. Конфигурации к подключению к БД должны быть настраиваемыми через файл или

ENV.

5. Проект должен быть покрыт тестами.

6. Проект должен проходить mypy, flake8 и прочее.

7. Отсутствие глобальных переменных.

Идеи для улучшения:

1. Использование абстракций/интерфейсов для возможной замены транспорта.

Например, с PostgreSQL на какое-нибудь хранилище (InMemory / Redis / File).

2. Тесты не должны зависеть от реальной базы данных (т. е. использовать класс,

фальшивку или моки).

Нам очень интересно посмотреть, как ты программируешь. Если что-то не удалось

реализовать — не отчаивайся, мы это тоже посмотрим.