ПРЕОБРАЗОВАНИЕ SQLALCHEMY В PYDENTIC И РАБОТА С ДАННЫМИ

ВВЕДЕНИЕ

Задачи урока

01 Научиться преобразовывать ответ от SqlAlchemy в Pydantic для того чтобы с ним можно было работать и отдавать в fastAPI в качестве ответа на запрос

Pydantic
это библиотека для проверки и сериализации данных в Python. Она предоставляет простой и элегантный способ определения моделей данных с помощью аннотаций типов Python. Pydantic обеспечивает валидацию данных и генерацию схемы данных автоматически, основываясь на определении моделей

Одной из главных особенностей **Pydantic** является возможность автоматической генерации валидаторов для моделей данных на основе типов данных. Валидаторы автоматически проверяют значения полей на соответствие определенным типам и предоставляют дружественные сообщения об ошибках. Это позволяет обеспечить типобезопасность данных и защиту от ошибок ввода

Кроме того, **Pydantic** предлагает интуитивно понятный интерфейс для работы с данными, включая поддержку различных методов сериализации, включая **JSON** и форматы данных, такие как **YAML**, **TOML** и **others**. С помощью **Pydantic** вы можете легко преобразовывать данные в различные форматы и валидировать их перед обработкой

Пример использования Pydantic для определения моделей данных:

```
from pydantic import BaseModel

class Files(BaseModel):
    Name: str
    Link: str
```

В библиотеке **Pydantic** модуль **model_validate** предоставляет функциональность для валидации моделей данных, определенных с помощью **Pydantic**. Он предлагает простой способ проверить, соответствуют ли данные заданным правилам валидации и возвращает объект ошибки в случае нарушения этих правил

Функция **model_validate** позволяет вам вручную выполнить валидацию экземпляра модели **Pydantic**, указав его в качестве аргумента. Если данные не соответствуют определенным правилам, будет сгенерировано исключение **ValueError** со списком сообщений об ошибках

Пример использования model_validate:

```
...
UsersWithFileDTO.model_validate(row, from_attributes=True)
```

ИТОГИ

- Для Валидации нам вначале необходимо реализовать **ODT** объект - это класс со всеми необходимыми полями
- После чего мы можем передать строку, которую нам вернул SqlAlchemy и используя метод **model_validate** преобразовать ответ в **ODT**