

Работа над ошибками

Друзья, я признаю, что курс не идеален и может содержать ошибки или небольшие неточности/противоречия. Тем не менее, я не боюсь признавать допущенные ошибки и хочу дать вам максимально полезный материал.

Файловая структура

В предыдущих уроках мной было решено объединить работу с отелями и номерами в одну папку `hotels/`. Я считаю это ошибкой и предлагаю перестроить папки и файлы следующим образом: мы не будем выделять папку `rooms/` на один уровень с `users/`, `hotels/` и `bookings/`. Вместо этого мы поместим ее внутрь папки `hotels/`. Таким образом, мы покажем, что номера являются неким подмножеством отелей. Внутри этой папки будут созданы привычные нам файлы `models`, `schemas`, `dao` и `router`.

Роутер(ы) для отелей и номеров

В задании в следующем шаге вы увидите, что для получения номеров в отеле необходим эндпоинт `/hotels/1/rooms`, который имеет префикс `/hotels` и указание на номер отеля. На данном этапе, учитывая разбиение отелей и номеров по разным папкам, есть 2 варианта развития событий: создать внутри папки `rooms/` отдельный роутер, который также будет иметь префикс `/hotels`, или импортировать роутер отелей и добавлять эндпоинты к нему, не создавая новый. Я остановился на втором варианте, но на данном этапе вы можете не согласиться и сделать по-своему.

1 вариант

```
# ===== hotels.py =====
router = APIRouter(prefix="/hotels")

@router.get("")
def get_hotels():
    ...

# ===== rooms.py =====
router = APIRouter(prefix="/hotels")

@router.get("/{hotel_id}/rooms")
def get_rooms():
    ...
```

2 вариант

```
# ===== hotels.py =====
router = APIRouter(prefix="/hotels")

@router.get("")
def get_hotels():
    ...

# ===== rooms.py =====
from app.hotels.router import router

@router.get("/{hotel_id}/rooms")
def get_rooms():
    ...
```

Обратите внимание, что во втором случае мы не создаем отдельный роутер, а импортируем уже существующий и добавляем эндпоинты к нему.