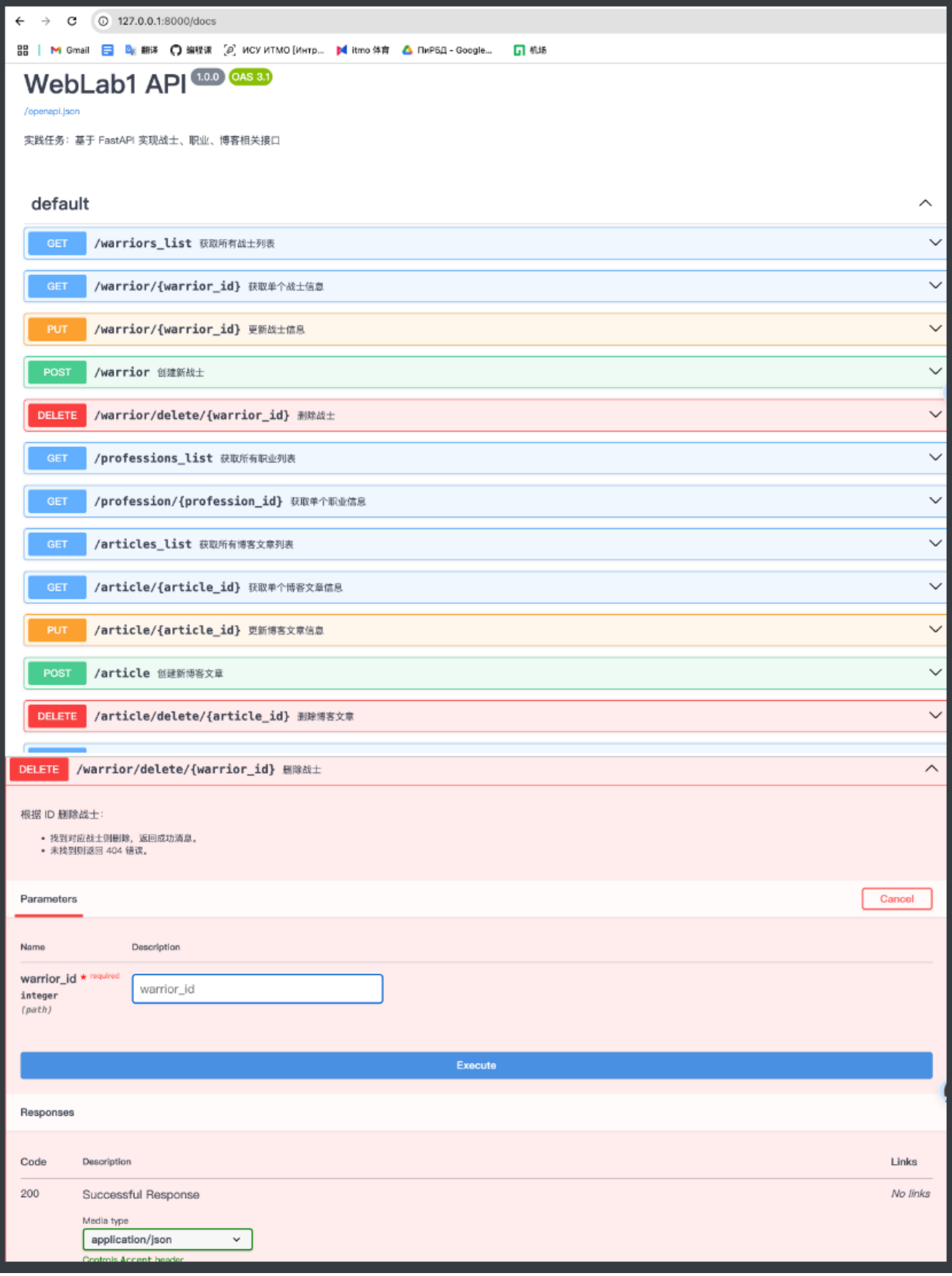
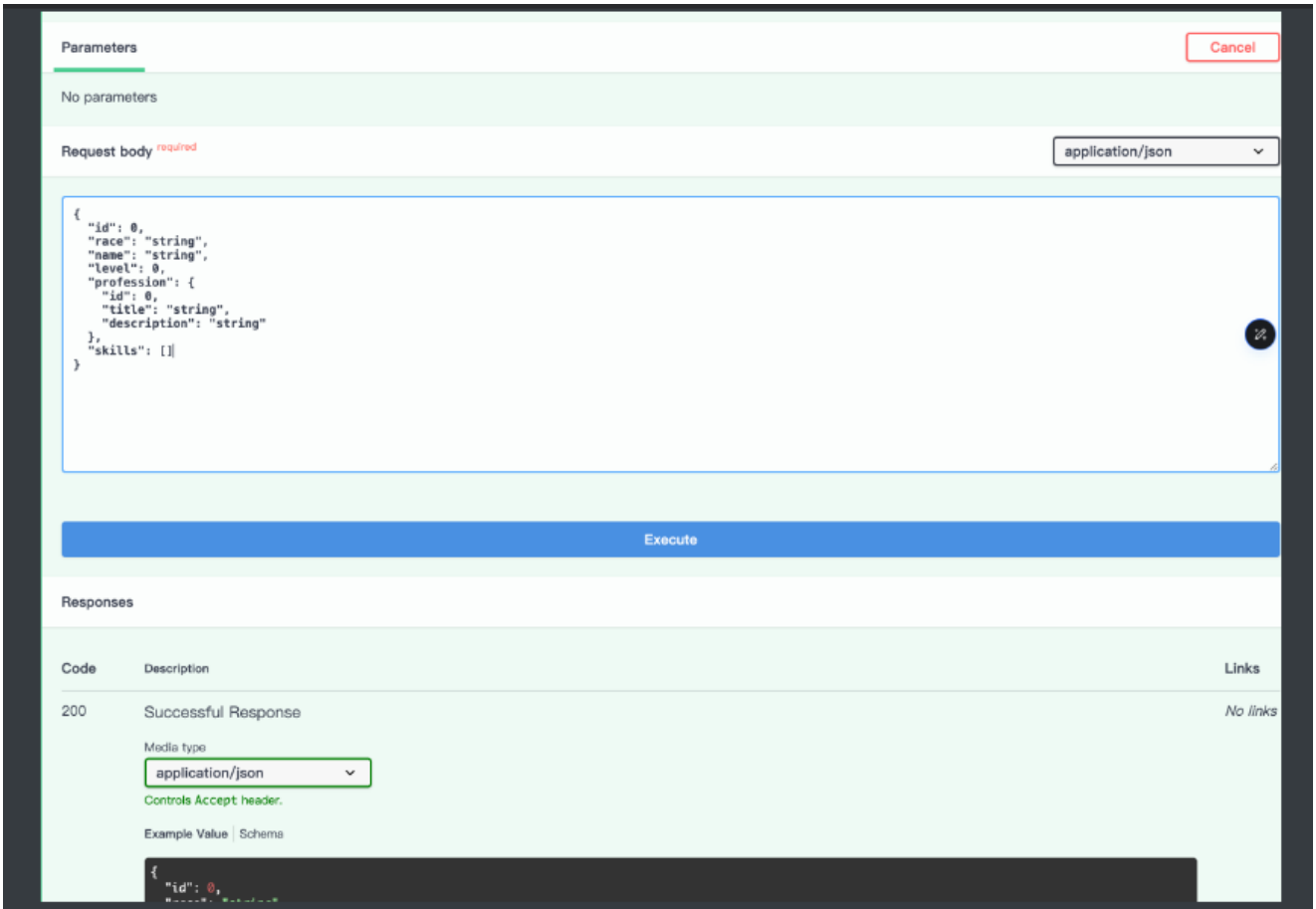
Лабораторная работа 1. Реализация серверного приложения FastAPI.

Необходимо реализовать полноценное серверное приложение с помощью фреймворка FastAPI с применением дополнительных средств и библиотек.

pra1

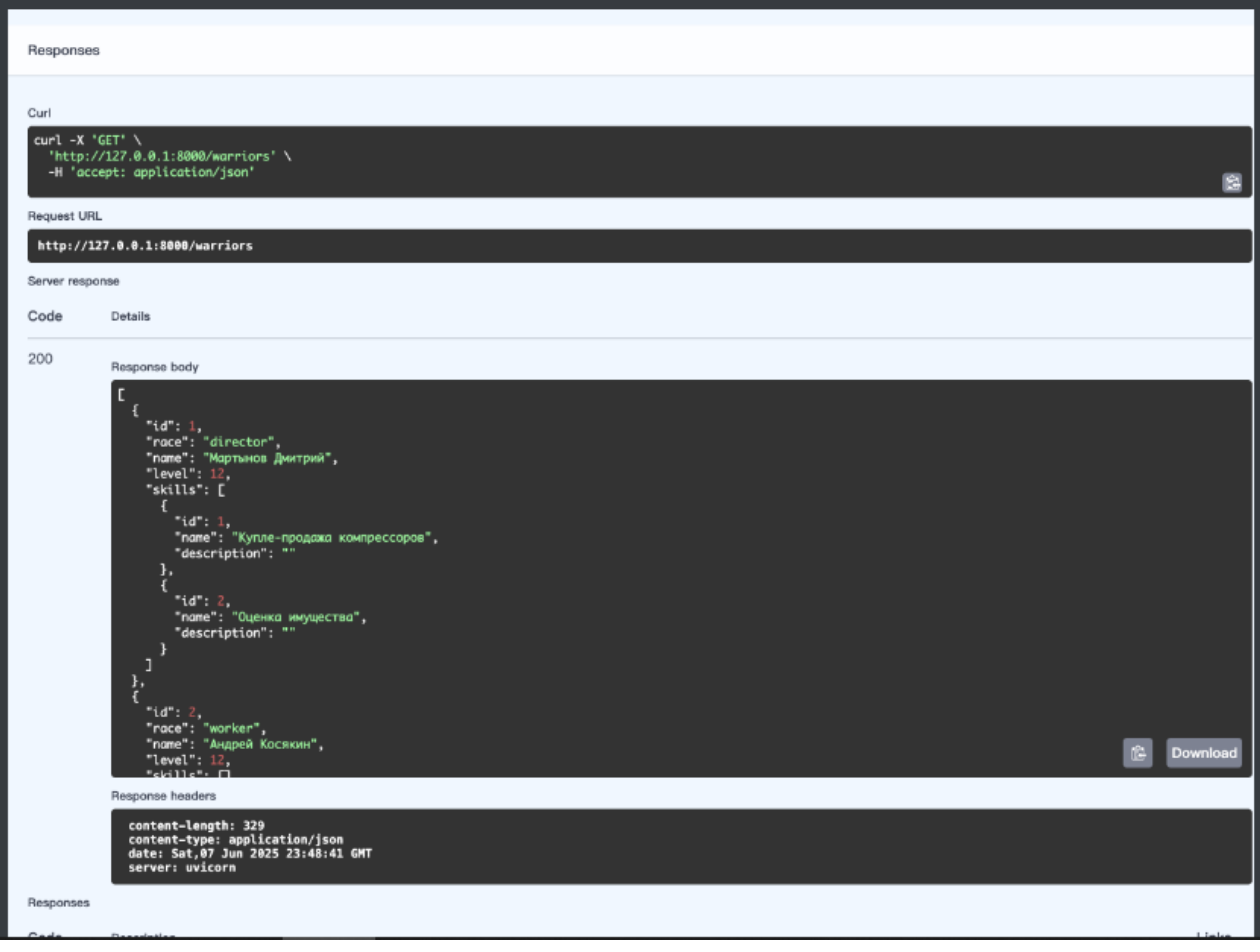
Создание конечных точек api и виртуальных баз данных

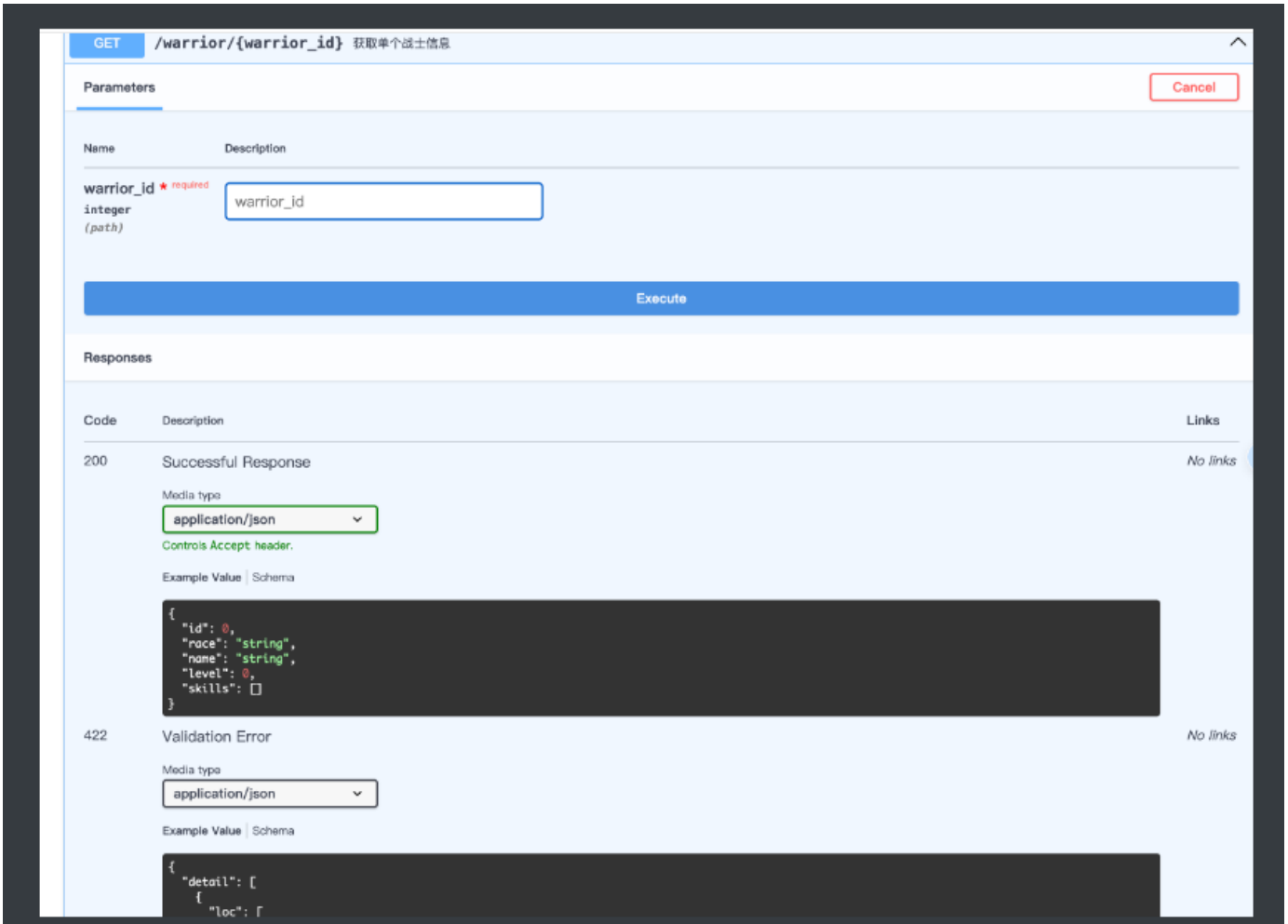


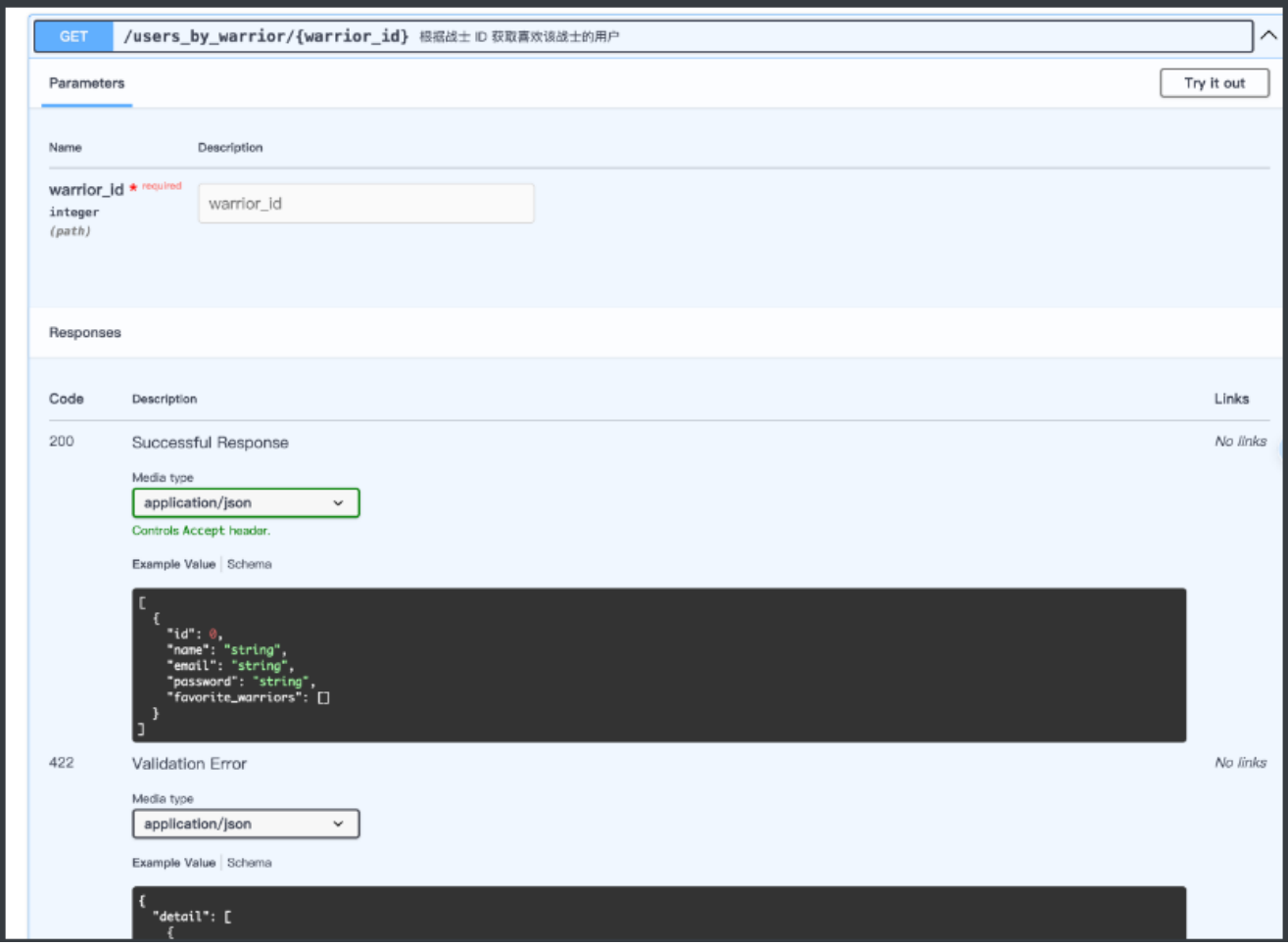


pra2

Настройка БД, SQLModel и миграции через Alembic

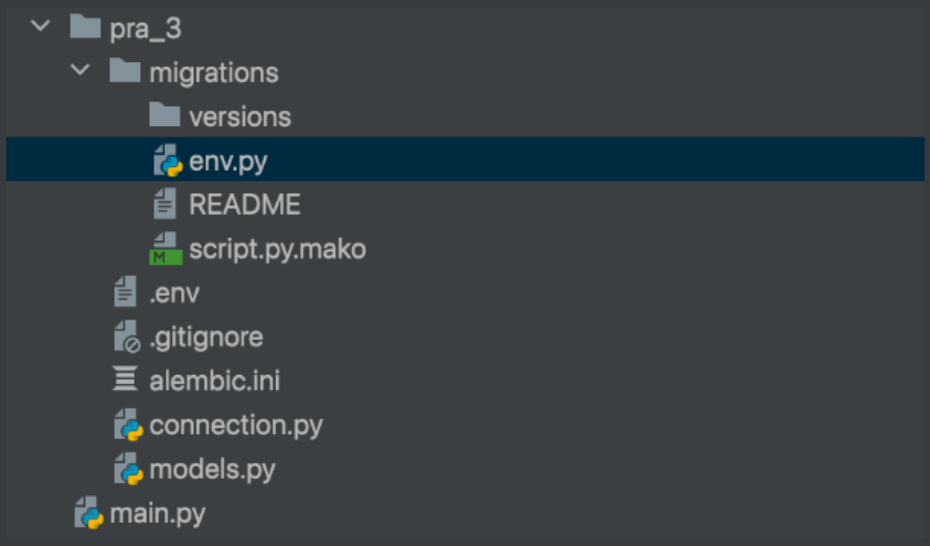



GET
/warrior/{warrior_id} 获取单个战士信息
Parameters
Cancel
Name
Description
warrior_id * required
integer
(path)
warrior_id
Execute
Responses
Code
Description
Links
200
Successful Response
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
{
"id": 0,
"race": "string",
"name": "string",
"level": 0,
"skills": []
}
422
Validation Error
No links
Media type
application/json
Example Value | Schema
{
"detail": [
{
"loc": [

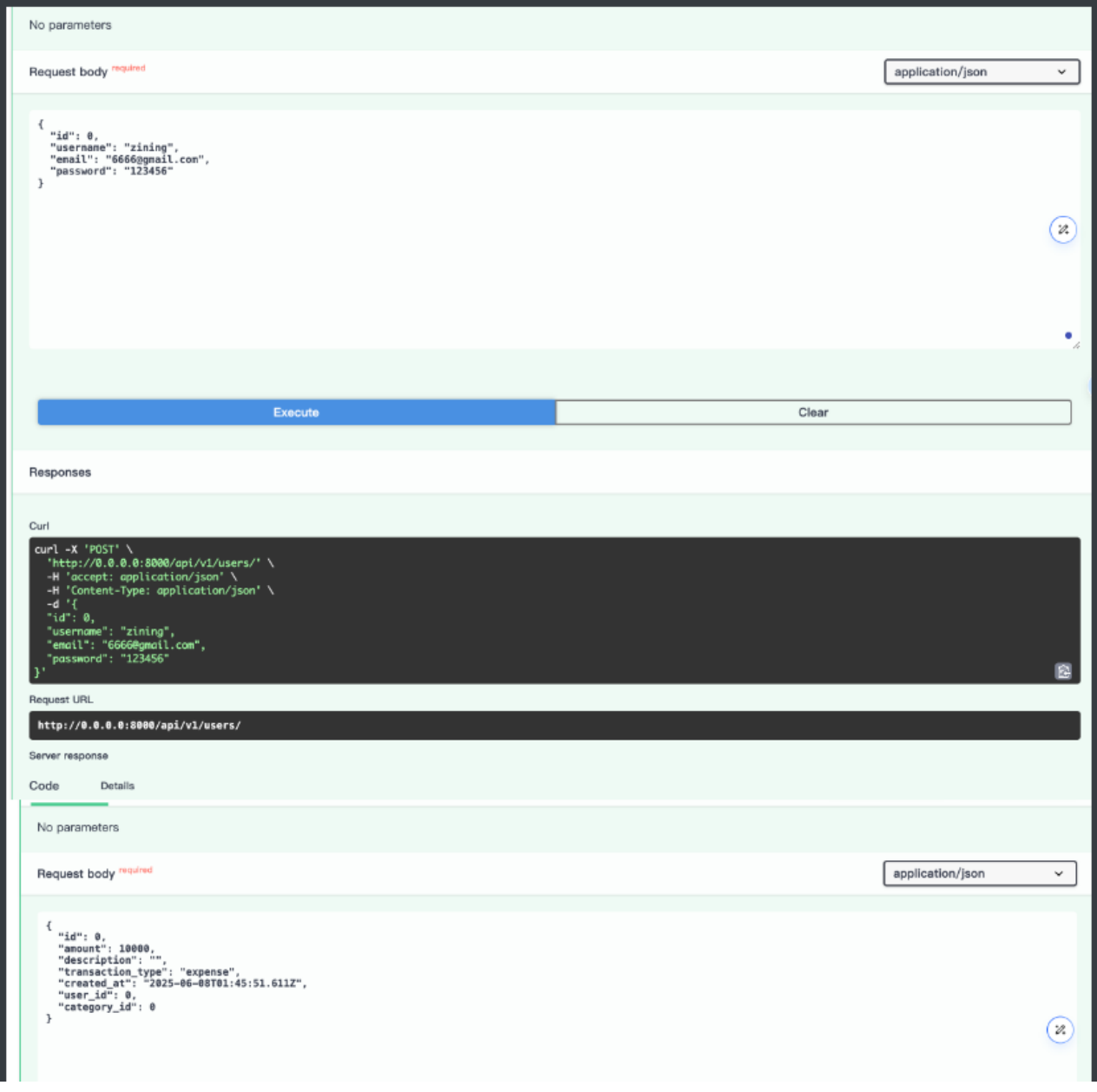

GET
/users_by_warrior/{warrior_id} 根据战士 ID 获取喜欢该战士的用户
Parameters
Try it out
Name
Description
warrior_id * required
integer
(path)
warrior_id
Responses
Code
Description
Links
200
Successful Response
No links
Media type
application/json
Controls Accept header.
Example Value | Schema
[
{
"id": 0,
"name": "string",
"email": "string",
"password": "string",
"favorite_warriors": []
}
]
422
Validation Error
No links
Media type
application/json
Example Value | Schema
{
"detail": [
{

pra3

Миграции, ENV, GitIgnore и структура проекта



Лабораторная работа 1



Curl

```
curl -X 'POST' \
  'http://0.0.0.0:8000/api/v1/transactions/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 0,
  "amount": 10000,
  "description": "",
  "transaction_type": "expense",
  "created_at": "2025-06-08T01:45:51.611Z",
  "user_id": 0,
  "category_id": 0
}'
```

Request URL

```
http://0.0.0.0:8000/api/v1/transactions/
```

Request body required

application/json

```
{
  "id": 0,
  "category": "string",
  "amount": 100000,
  "start_date": "2025-06-08T01:46:45.468Z",
  "end_date": "2025-06-08T01:46:45.468Z",
  "user_id": 0
}
```

Execute | Clear

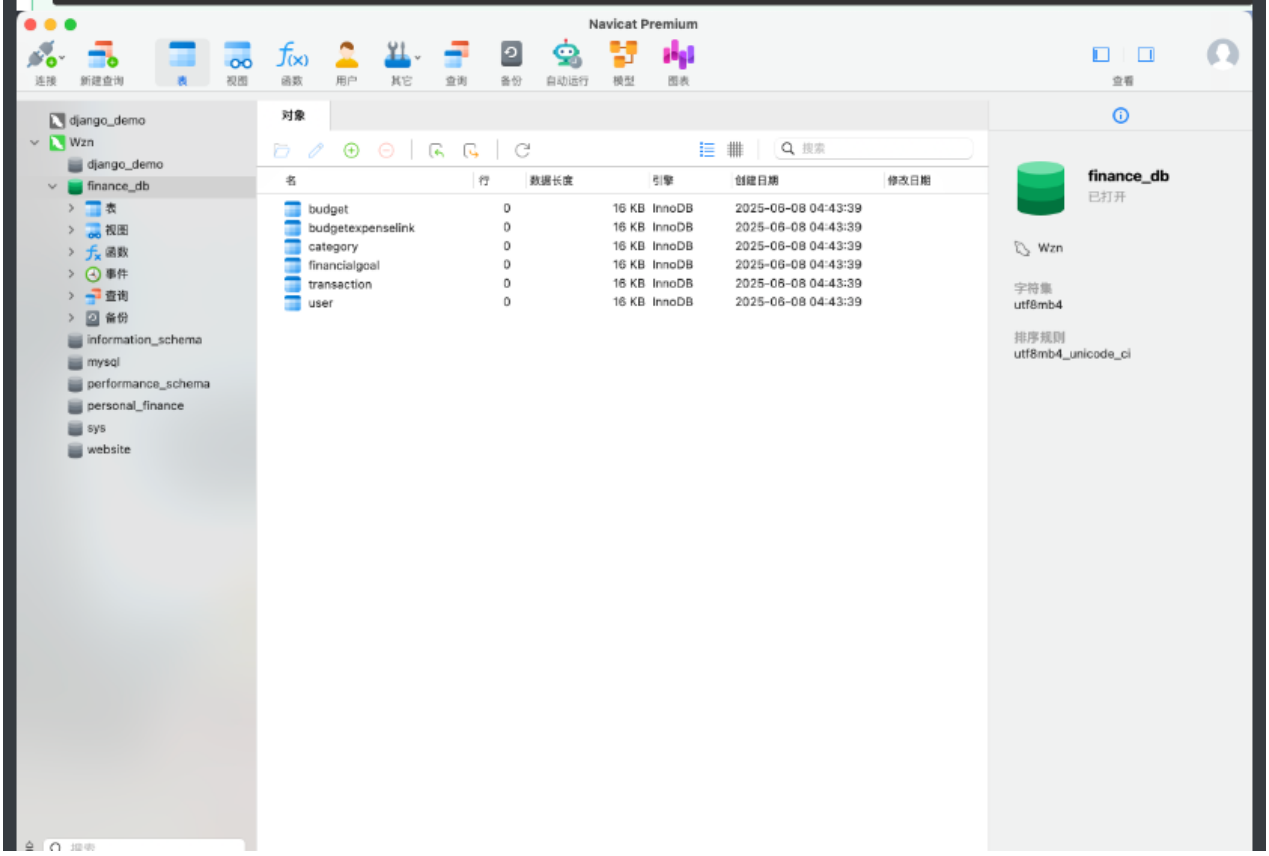
Responses

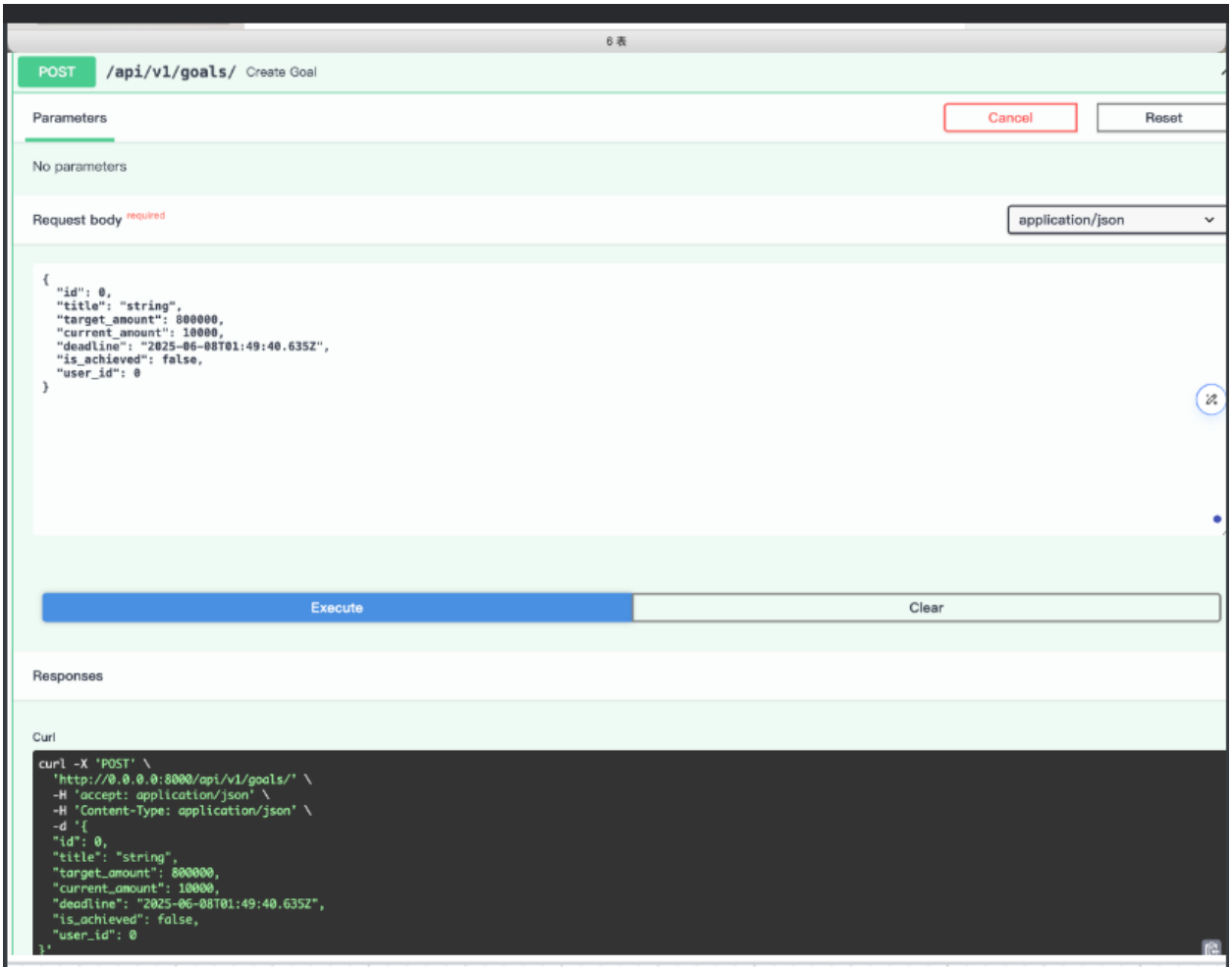
Curl

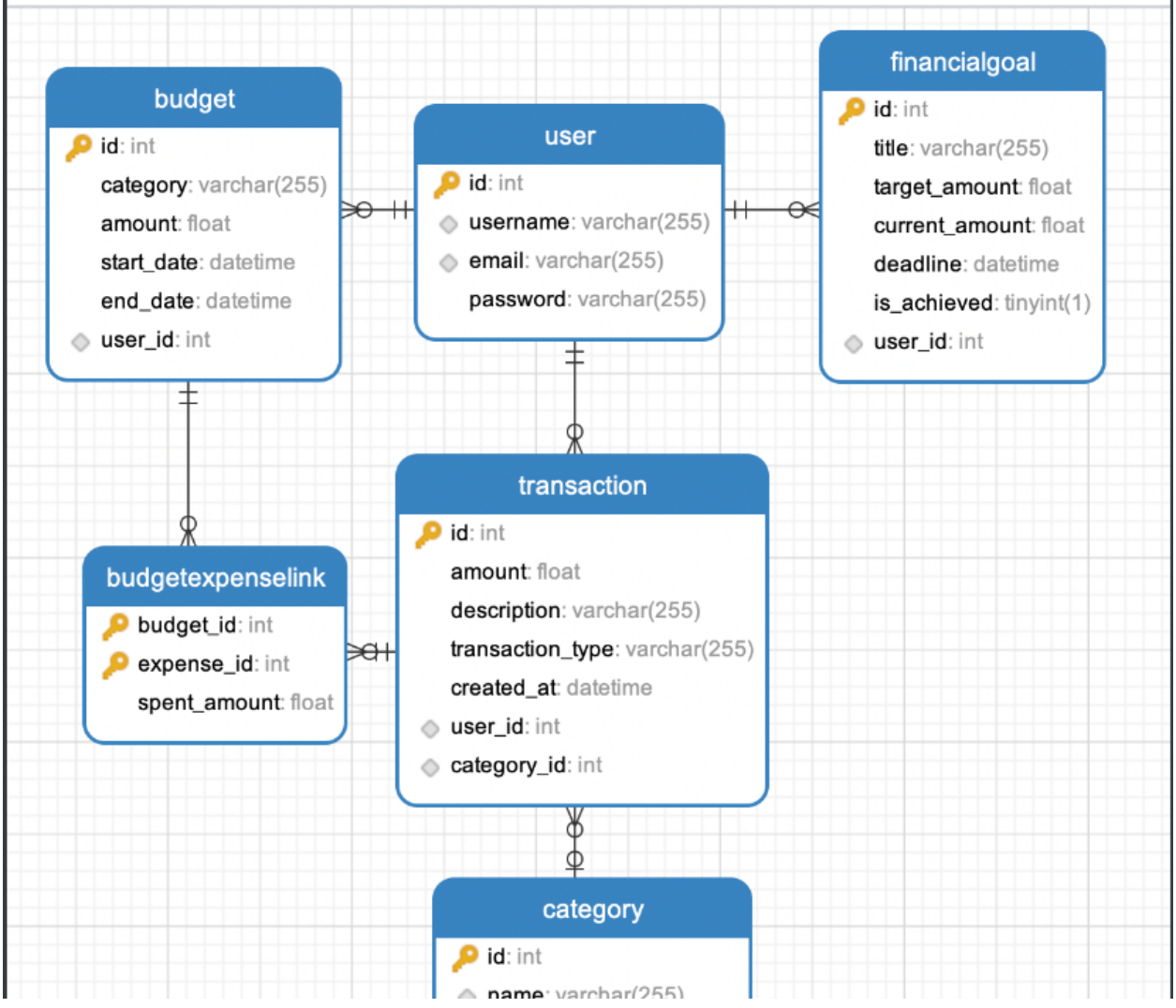
```
curl -X 'POST' \
  'http://0.0.0.0:8000/api/v1/budgets/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 0,
  "category": "string",
  "amount": 100000,
  "start_date": "2025-06-08T01:46:45.468Z",
  "end_date": "2025-06-08T01:46:45.468Z",
  "user_id": 0
}'
```

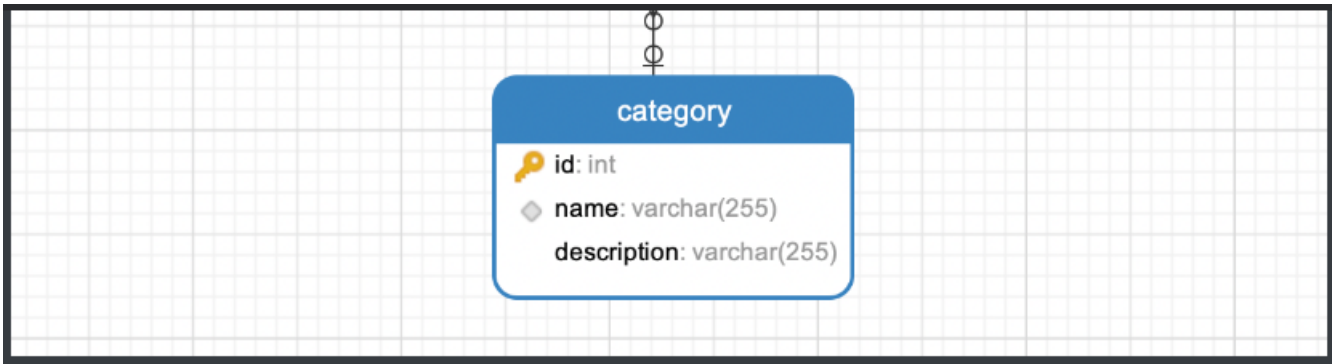
Request URL

```
http://0.0.0.0:8000/api/v1/budgets/
```




6 表
POST
/api/v1/goals/
Create Goal
Parameters
Cancel
Reset
No parameters
Request body
required
application/json
{
"id": 0,
"title": "string",
"target_amount": 800000,
"current_amount": 10000,
"deadline": "2025-06-08T01:49:40.635Z",
"is_achieved": false,
"user_id": 0
}
Execute
Clear
Responses
Curl
curl -X 'POST' \
'http://0.0.0.0:8000/api/v1/goals/' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"id": 0,
"title": "string",
"target_amount": 800000,
"current_amount": 10000,
"deadline": "2025-06-08T01:49:40.635Z",
"is_achieved": false,
"user_id": 0
}'


budget
id: int
category: varchar(255)
amount: float
start_date: datetime
end_date: datetime
user_id: int
user
id: int
username: varchar(255)
email: varchar(255)
password: varchar(255)
financialgoal
id: int
title: varchar(255)
target_amount: float
current_amount: float
deadline: datetime
is_achieved: tinyint(1)
user_id: int
budgetexpenselink
budget_id: int
expense_id: int
spent_amount: float
transaction
id: int
amount: float
description: varchar(255)
transaction_type: varchar(255)
created_at: datetime
user_id: int
category_id: int
category
id: int
name: varchar(255)



Лабораторная работа 2. Потоки. Процессы. Асинхронность.

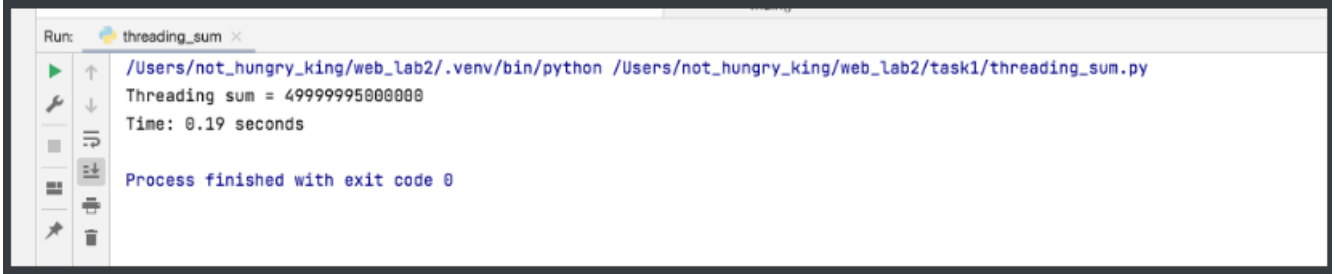
Цель работы: понять отличия потоками и процессами и понять, что такое ассинхронность в Python.

Работа о потоках, процессах и асинхронности поможет студентам развить навыки создания эффективных и быстродействующих программ, что важно для работы с большими объемами данных и выполнения вычислений.

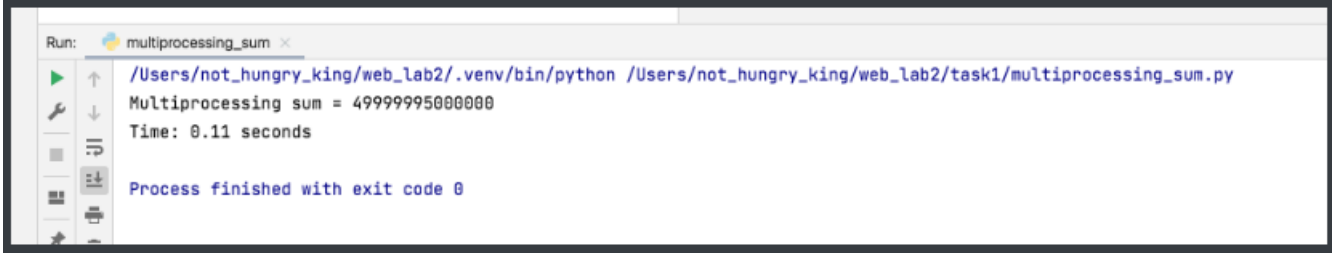
Задача 1. Различия между threading, multiprocessing и async в Python

Задача: Напишите три различных программы на Python, использующие каждый из подходов: threading, multiprocessing и async. Каждая программа должна решать считать сумму всех чисел от 1 до 10000000000000. Разделите вычисления на несколько параллельных задач для ускорения выполнения.

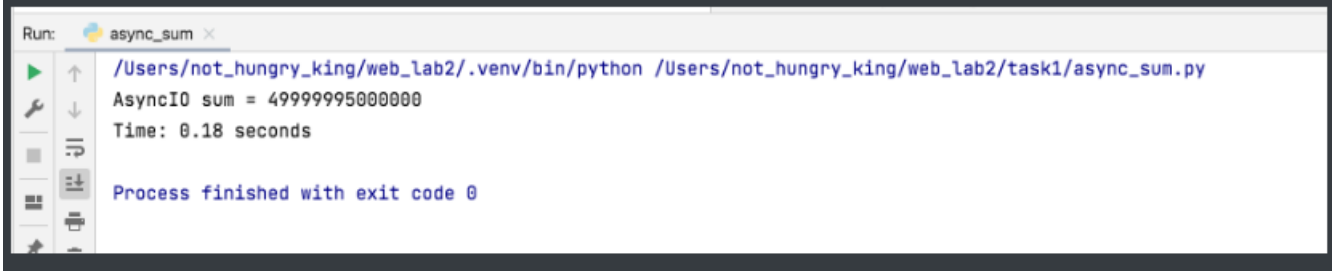
threading.py:



Multiprocessing — Процессы, multiprocessing.py:



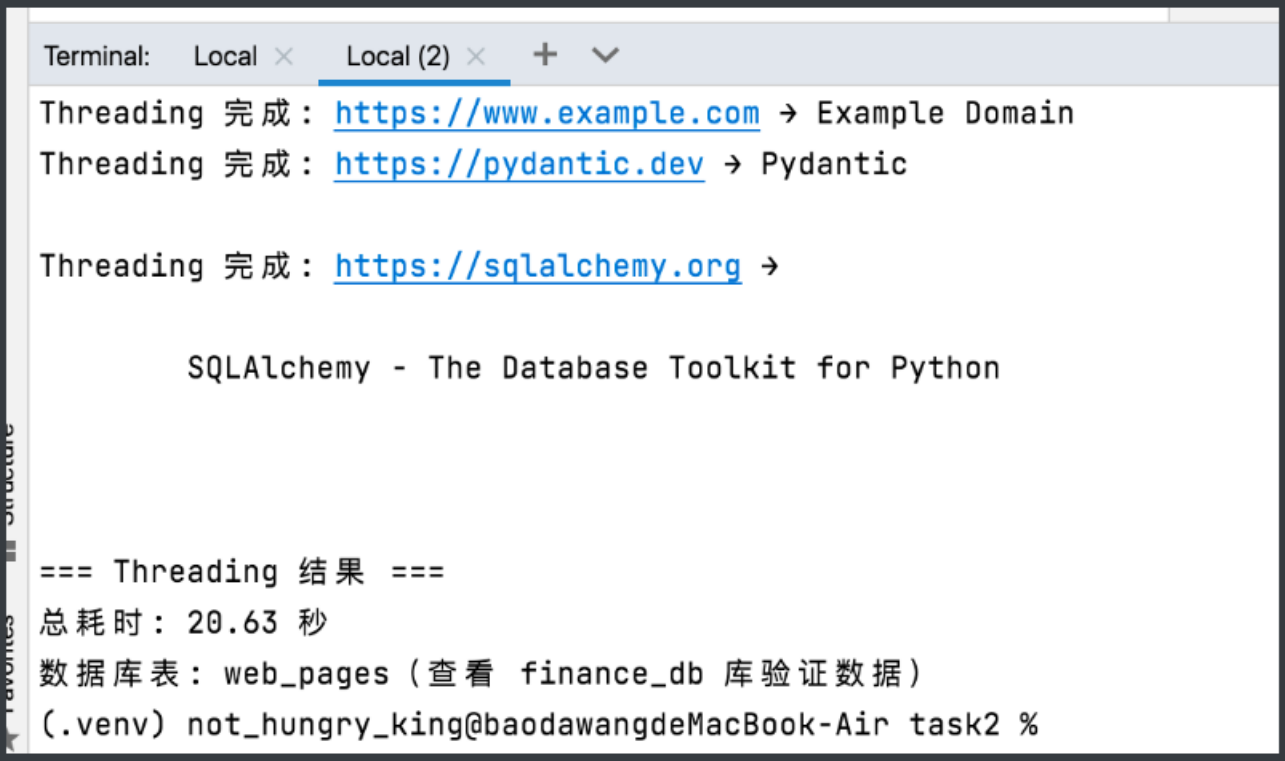
Async - Асинхронность, async.py:



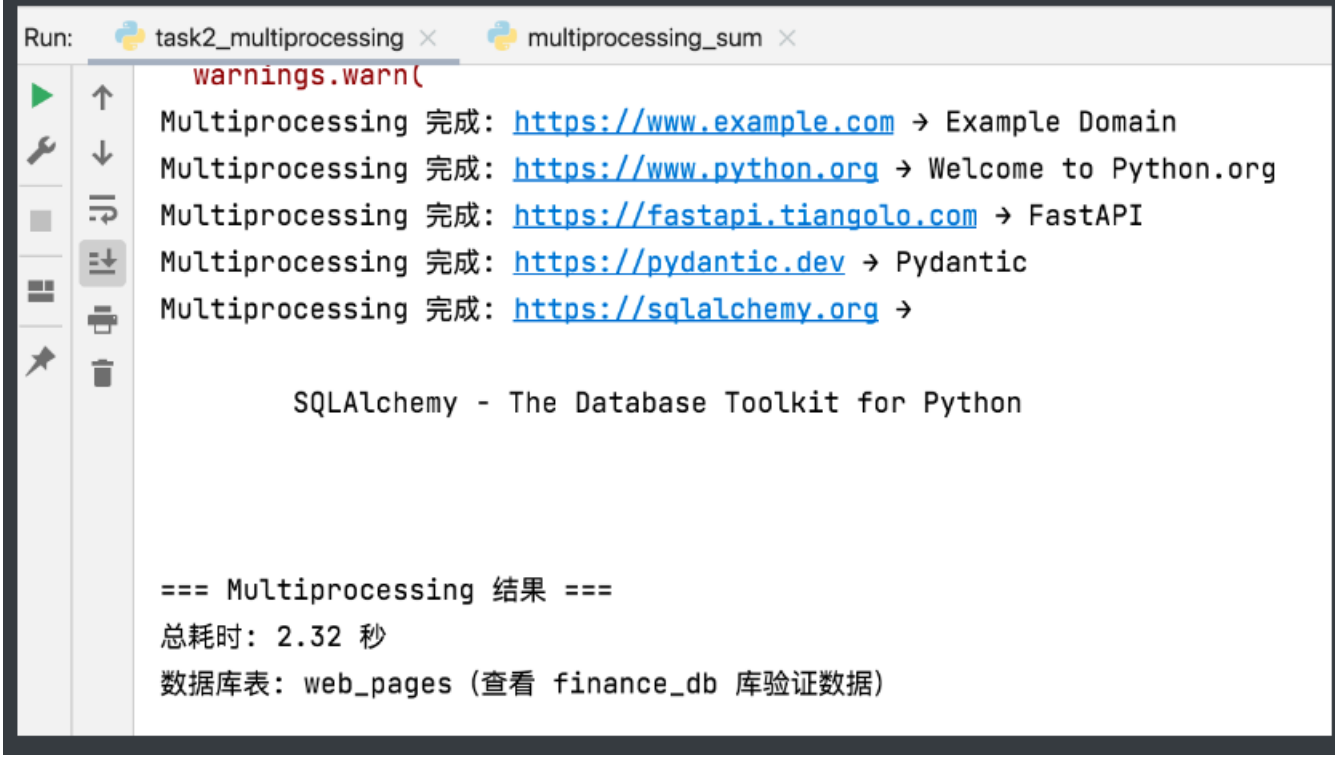
Задача 2. Параллельный парсинг веб-страниц с сохранением в базу данных

Задача: Напишите программу на Python для параллельного парсинга нескольких веб-страниц с сохранением данных в базу данных с использованием подходов threading, multiprocessing и async. Каждая программа должна парсить информацию с нескольких веб-сайтов, сохранять их в базу данных.

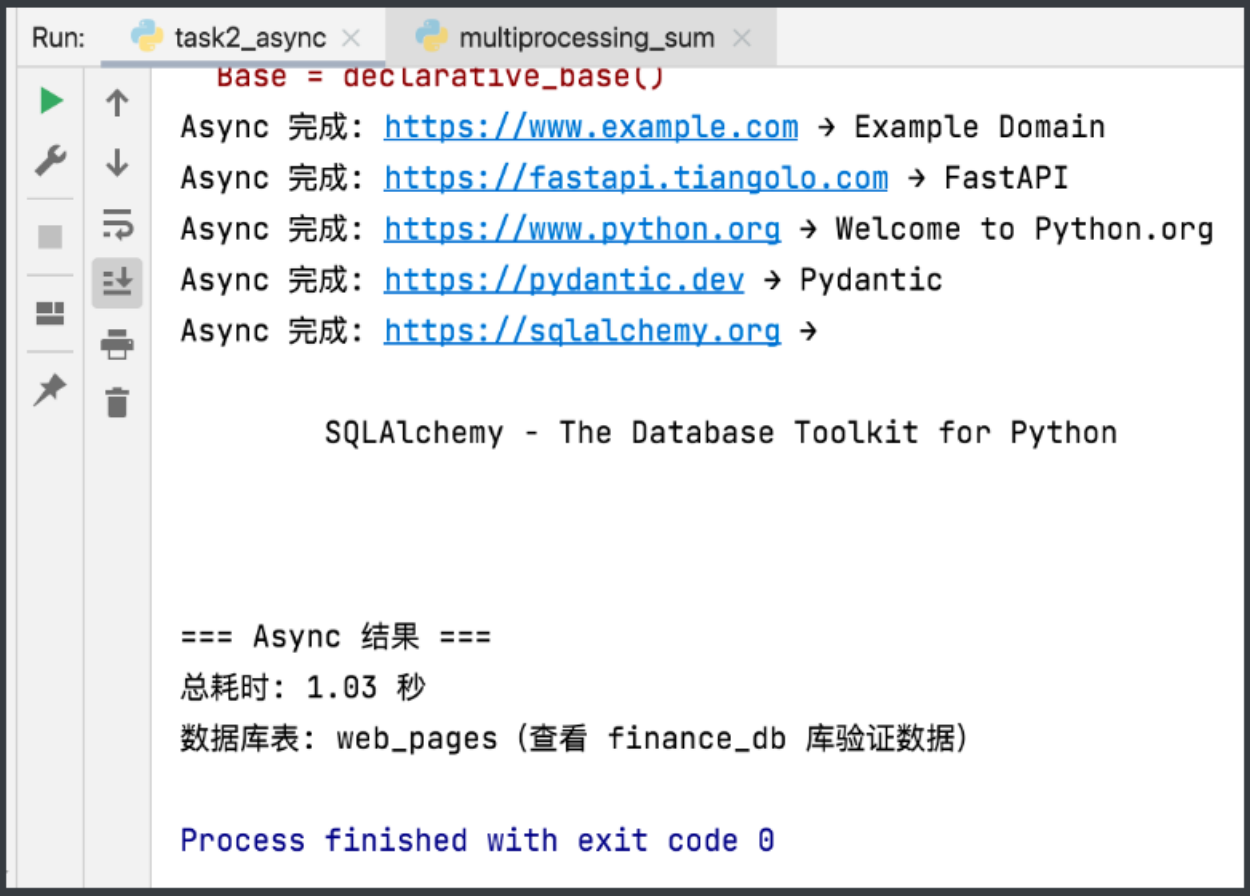
threading.py:



Multiprocessing — Процессы, multiprocessing.py:

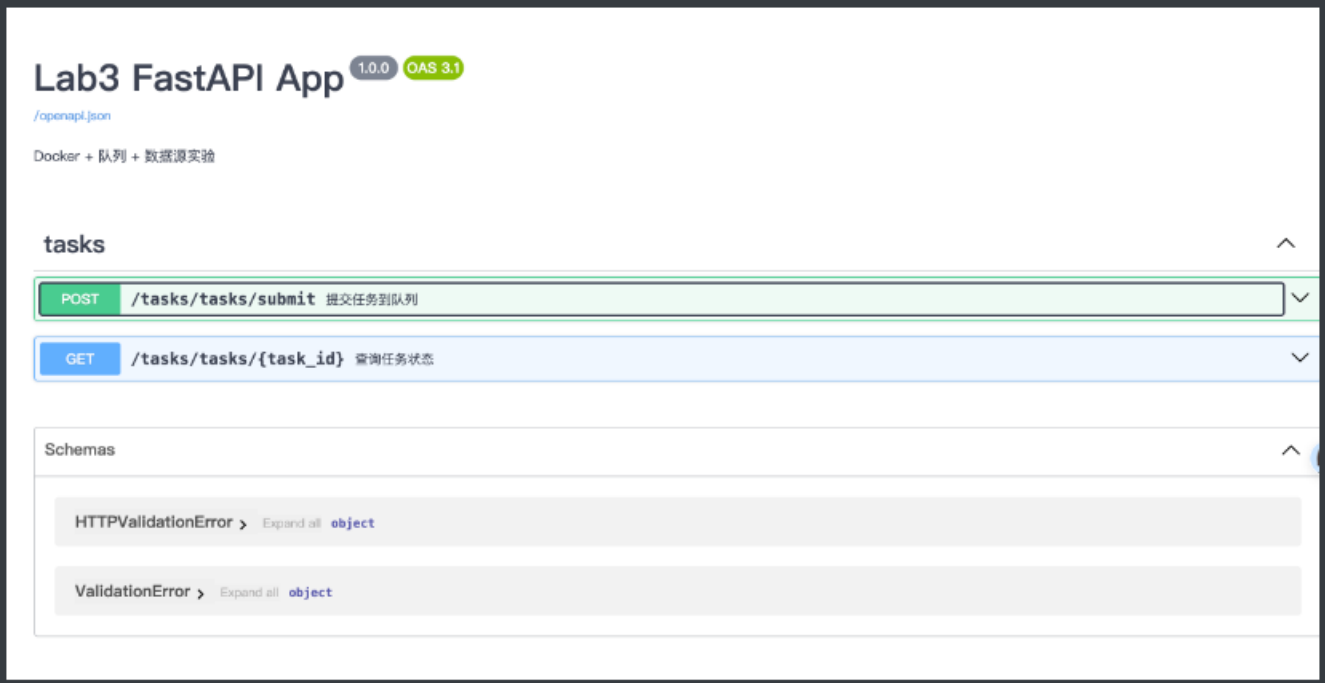


Async - Асинхронность, async.py:



Лабораторная работа 3: Упаковка FastAPI приложения в Docker, Работа с источниками данных и Очереди

Цель: Научиться упаковывать FastAPI приложение в Docker, интегрировать парсер данных с базой данных и вызывать парсер через API и очередь.



код из файла Dockerfile:

```
FROM python:3.9-slim
WORKDIR /app
COPY Requirements.txt .
RUN pip install --no-cache-dir -r Requirements.txt
COPY . .
EXPOSE 8000
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

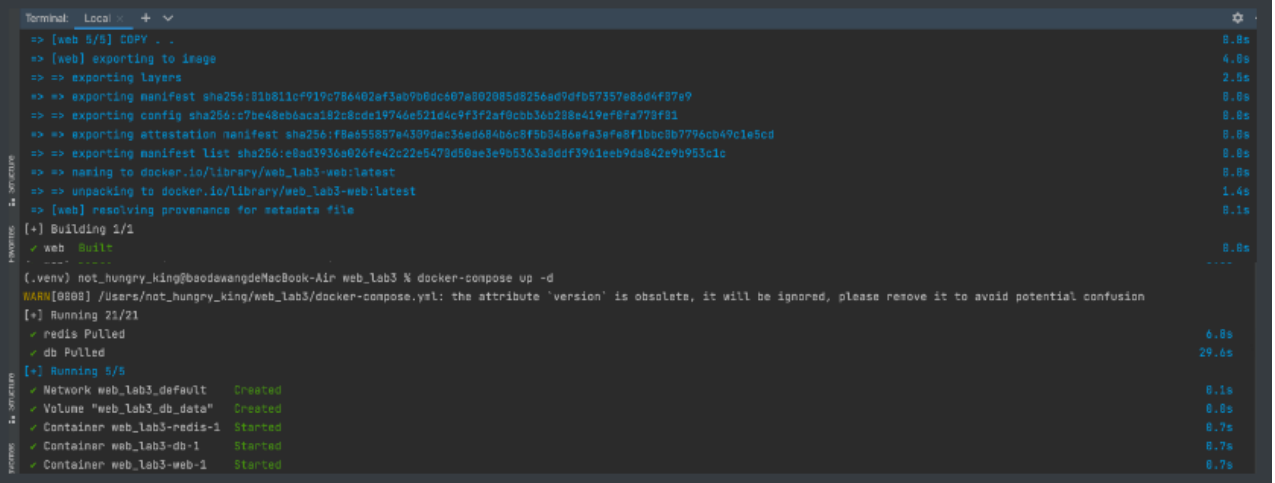
код из файла Docker-compose.yml

```
version: "3.8"

services:
  web:
    build:
      context: .
      dockerfile: Docker/Dockerfile
    ports:
      - "8000:8000"
    environment:
      - DATABASE_URL=mysql+pymysql://root:wzn001021@db/finance_db
      - REDIS_URL=redis://redis:6379/0
    depends_on:
      - db
      - redis

  worker:
    build:
      context: .
      dockerfile: Docker/Dockerfile
    command: celery -A app.celery_config.celery_app worker --loglevel=info
    environment:
      - DATABASE_URL=mysql+pymysql://root:wzn001021@db/finance_db
      - REDIS_URL=redis://redis:6379/0
    depends_on:
      - db
      - redis

  db:
    image: mysql:8.0
    environment:
      - MYSQL_ROOT_PASSWORD=wzn001021
      - MYSQL_DATABASE=finance_db
    volumes:
      - mysql_data:/var/lib/mysql

  redis:
    image: redis:7.0-alpine
    ports:
      - "6379:6379"

volumes:
  mysql_data:
```

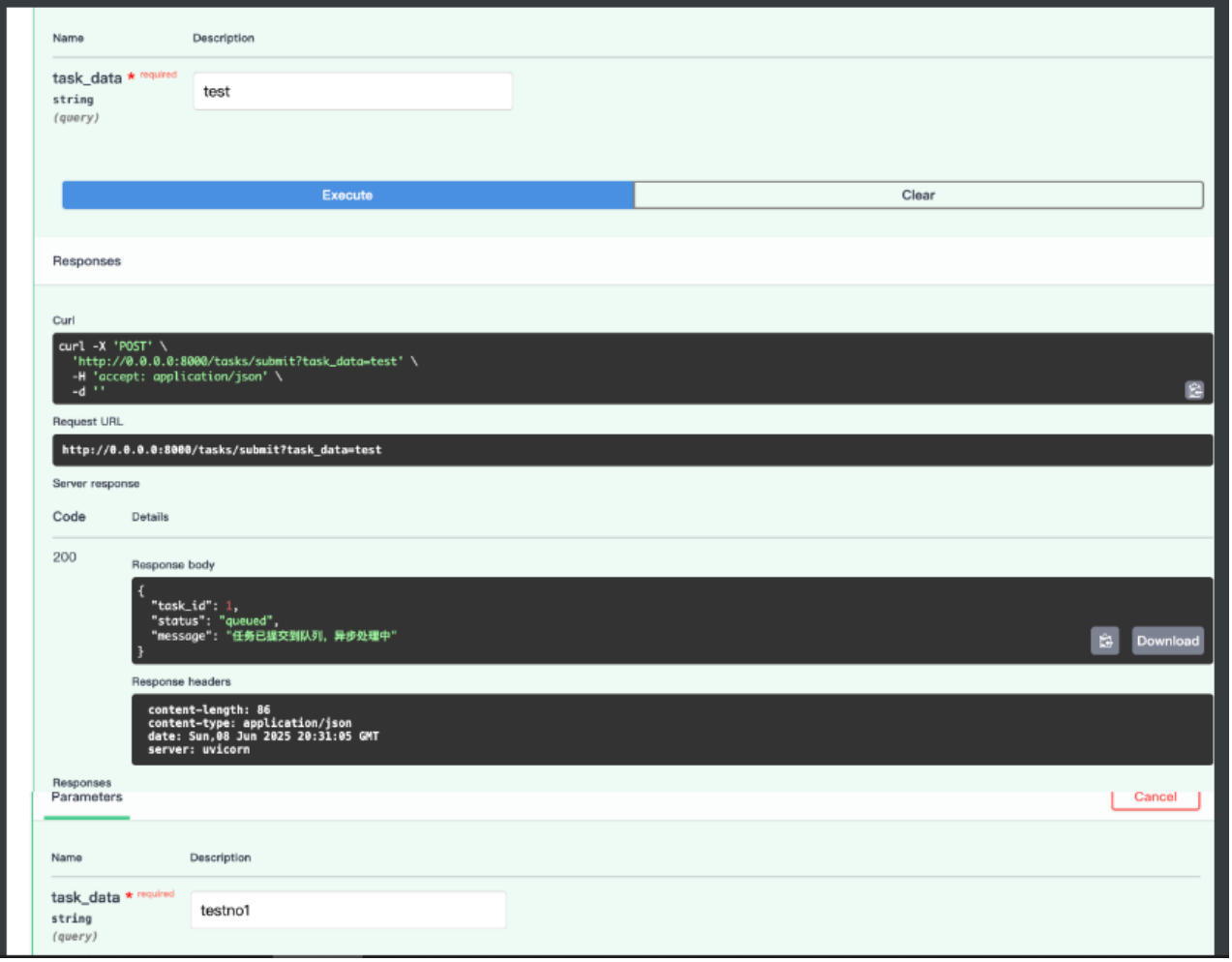
код из файла celery_config.py:

```python
from celery import Celery
from os import getenv

celery_app = Celery(
    "lab3_worker",
    broker=getenv("REDIS_URL", "redis://localhost:6379/0"),
    backend=getenv("REDIS_URL", "redis://localhost:6379/0"),
    include=["app.routers.tasks"]
)

celery_app.conf.timezone = "UTC"
celery_app.conf.task_serializer = "json"
```

Запуск бэкенда celery и клиента FASTApi:



Выполните функциональное тестирование:



```
}
```

Response headers

```
content-length: 86
content-type: application/json
date: Sun,08 Jun 2025 20:31:05 GMT
server: uvicorn
```

Responses

Parameters Cancel

Name	Description
task_data * required string (query)	testno1

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://0.0.0.0:8000/tasks/submit?task_data=testno1' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://0.0.0.0:8000/tasks/submit?task_data=testno1
```

Server response

Code	Details
200	Response body

```
{
  "task_id": 1,
  "status": "queued",
  "message": "任务已提交到队列，异步处理中"
}
```

Download

Response headers

```
content-length: 86
content-type: application/json
date: Sun,08 Jun 2025 20:41:49 GMT
server: uvicorn
```

Responses

Code	Description	Links

Gmail 翻译 编程课 ИСУ ИТМО [интр... itmo 体育 ПИРЫД - Google... 机场

Name	Description
task_id * required integer (path)	1

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://0.0.0.0:8000/tasks/1' \
  -H 'accept: application/json'
```

Request URL

```
http://0.0.0.0:8000/tasks/1
```

Server response

Code	Details
200	Response body

```
{
  "task_id": 1,
  "status": "completed",
  "data": "testno1",
  "created_at": "2025-06-08T23:41:50",
  "completed_at": "2025-06-08T23:41:56"
}
```

Download

Response headers

```
content-length: 123
content-type: application/json
date: Sun,08 Jun 2025 20:42:25 GMT
server: uvicorn
```

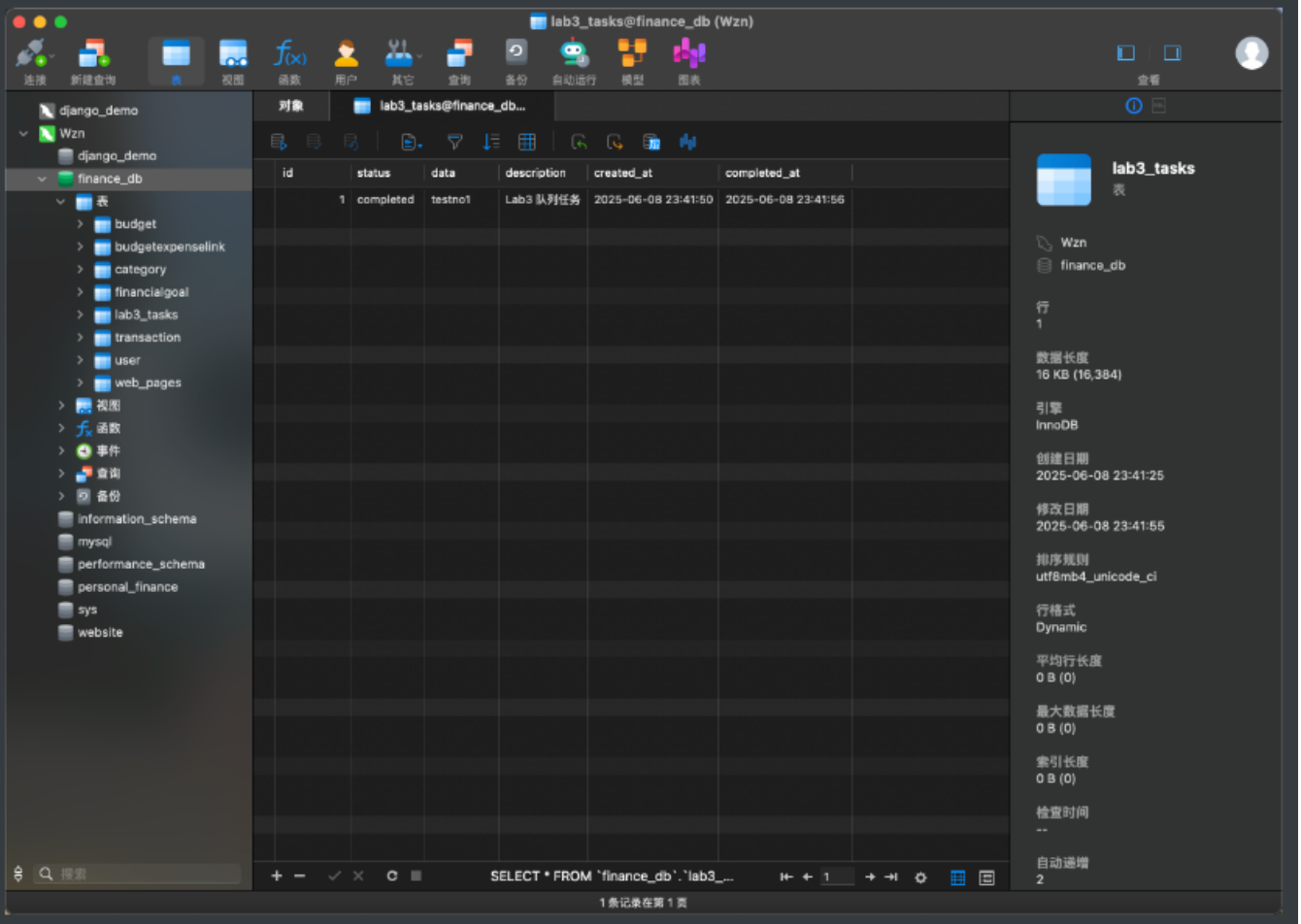
Responses

Code	Description	Links

Responses

Code	Description	Links

База данных бэк-офиса:



Терминал выполнения и сельдерейный дисплей: