**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

**Лабораторная работа № 14. Создание простого REST API с использованием FastAPI**

Выполнил: Студент группы

БПИ2401

Исламов Эмин

**Цель работы: ознакомиться с основами FastAPI и научиться создавать простые REST-сервисы с различными типами параметров.**

**Ход работы:**

**Шаг 1. Установка библиотек**

pip install fastapi

pip install uvicorn

pip install pyjokes

**Шаг 2. Main.py**

from fastapi import FastAPI

import pyjokes

app = FastAPI()

@app.get("/")

def joke():

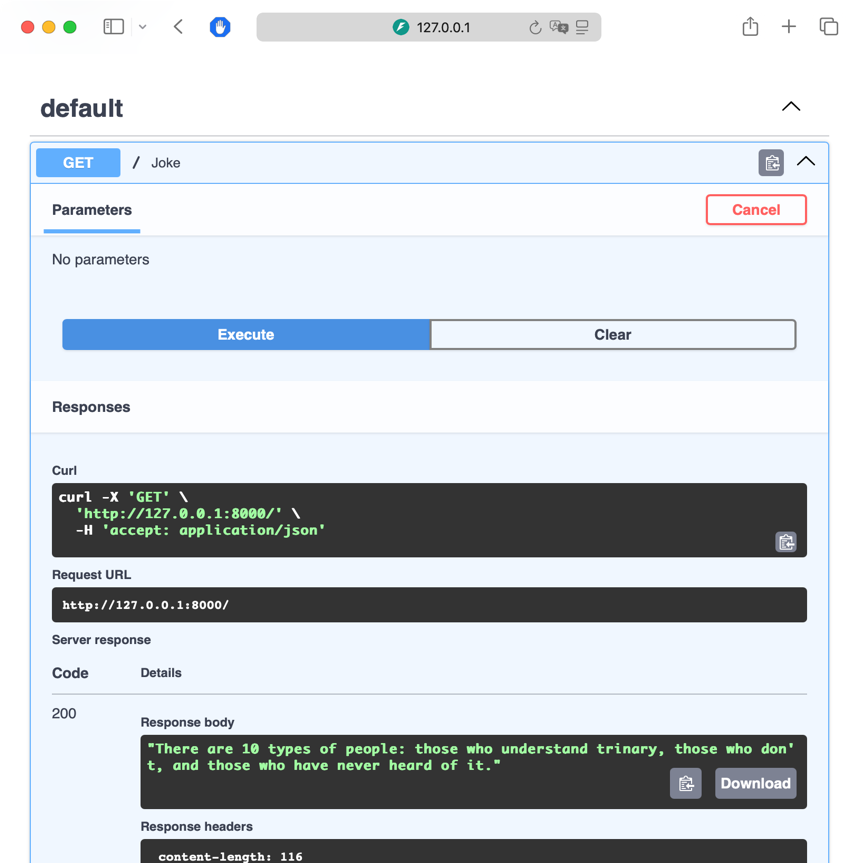
return pyjokes.get\_joke()

**Шаг 3. Запуск uvicorn**

uvicorn main:app –reload

Шаг 4. Swagger-документация

<http://127.0.0.1:8000/docs>



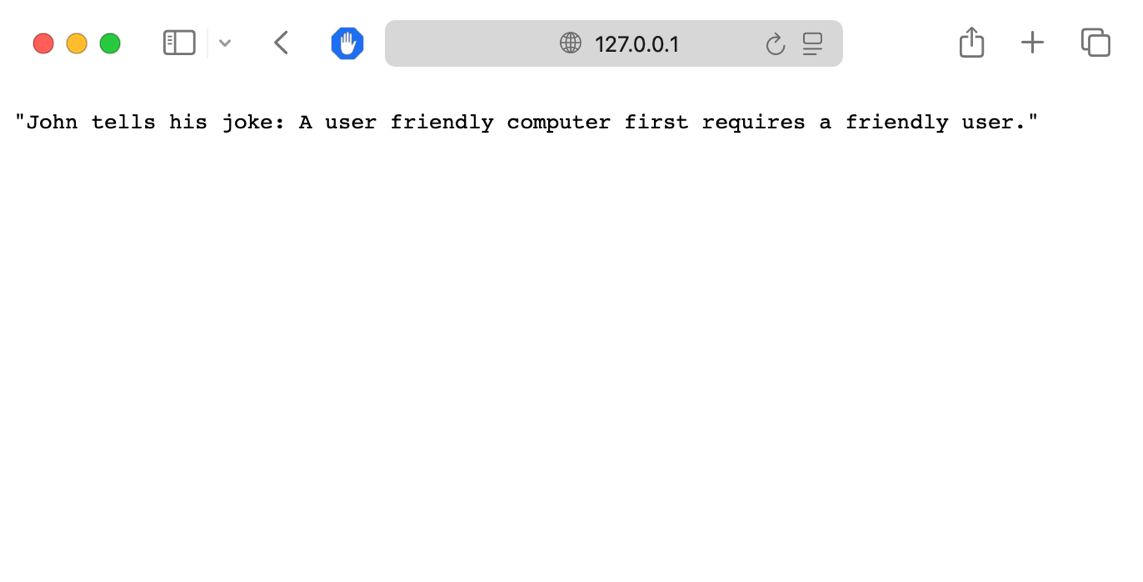
**Шаг 5. Роут с параметром в пути**

@app.get("/{friend}")

def friends\_joke(friend: str):

    return friend + " tells his joke: " + pyjokes.get\_joke()

Проверяем по адресу <http://127.0.0.1:8000/John>



**Шаг 6. Роут с query-параметром**

@app.get("/multi/{friend}")

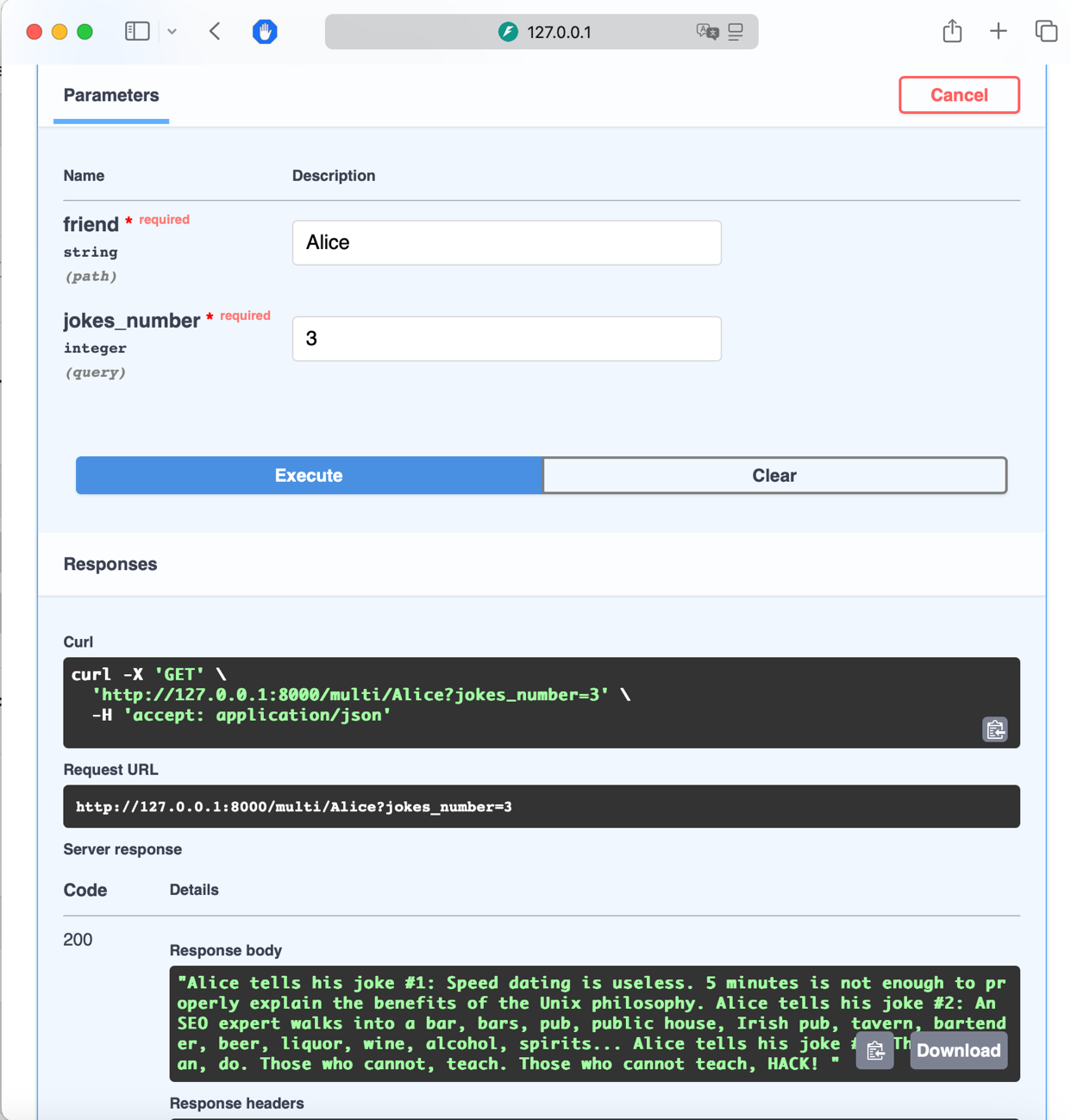
def multi\_friends\_joke(friend: str, jokes\_number: int):

    result = ""

    for i in range(jokes\_number):

        result += friend + f" tells his joke #{i + 1}: " + pyjokes.get\_joke() + " "

    return result



**Шаг 7. POST-запрос с Body**

from pydantic import BaseModel

class Joke(BaseModel):

    friend: str

    joke: str

class JokeInput(BaseModel):

    friend: str

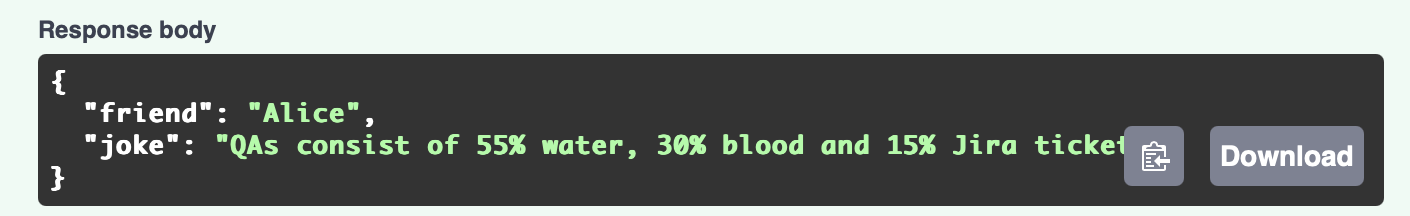
@app.post("/", response\_model=Joke)

def create\_joke(joke\_input: JokeInput):

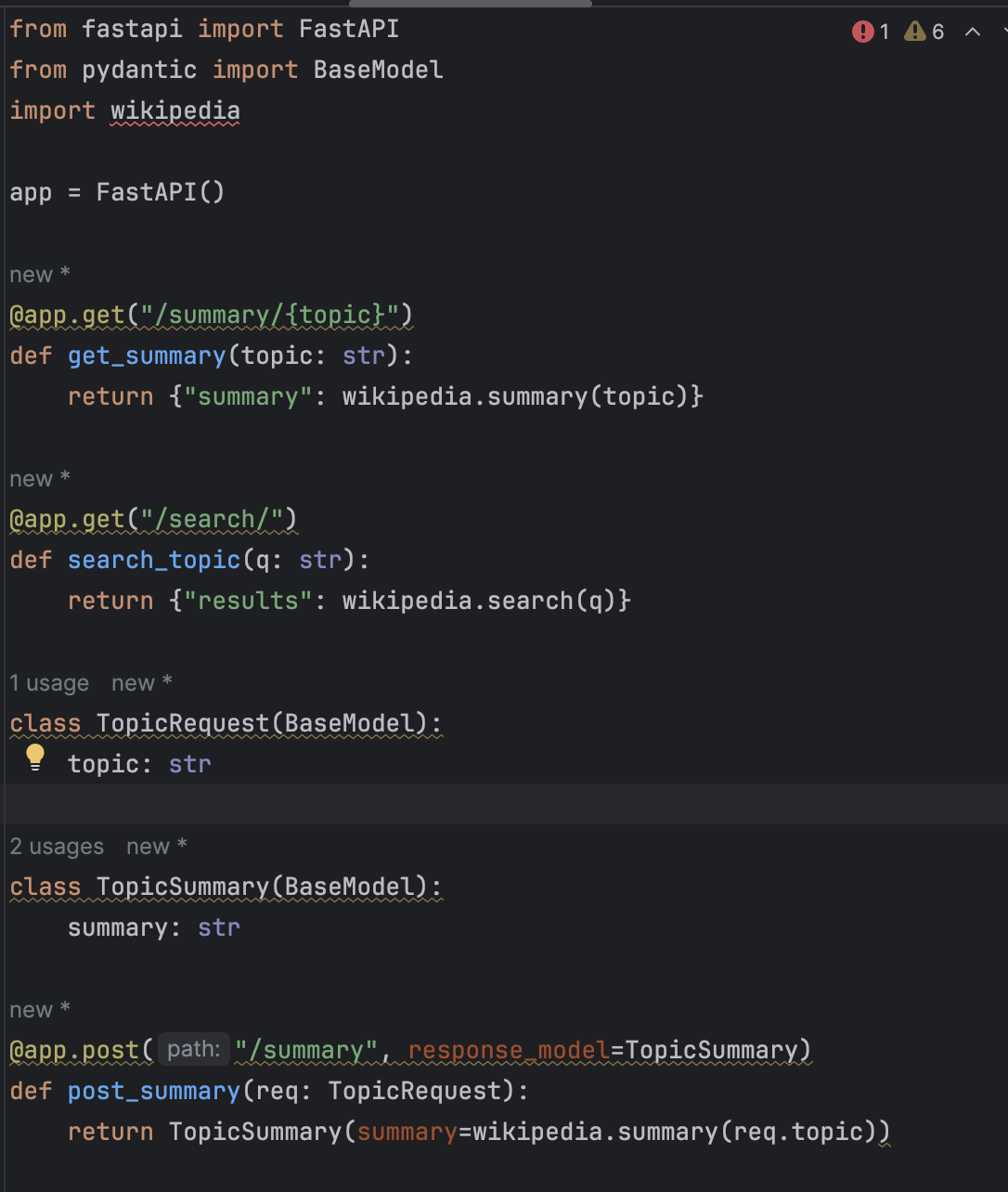
    return Joke(friend=joke\_input.friend, joke=pyjokes.get\_joke())

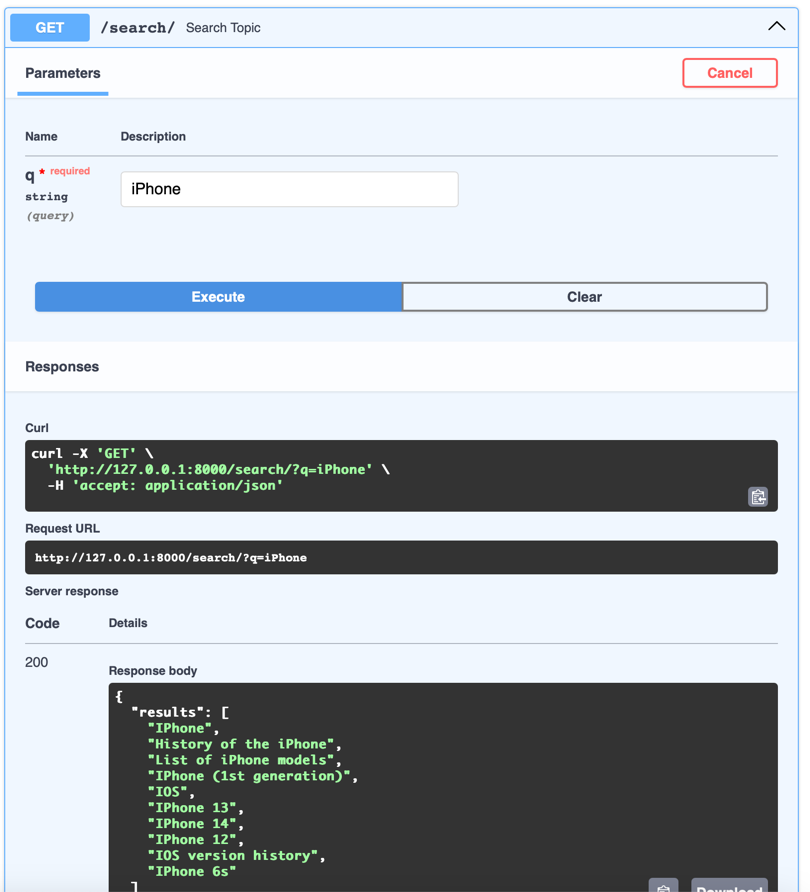
Тело запроса

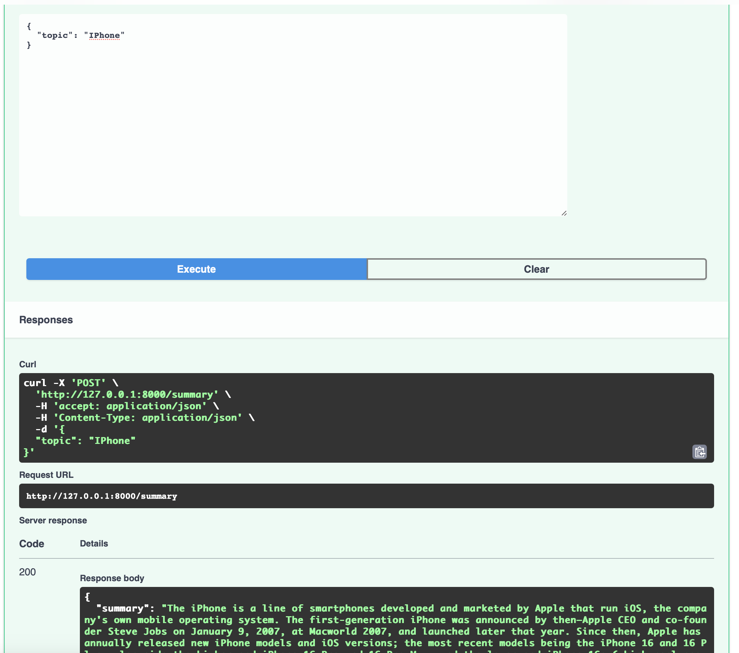
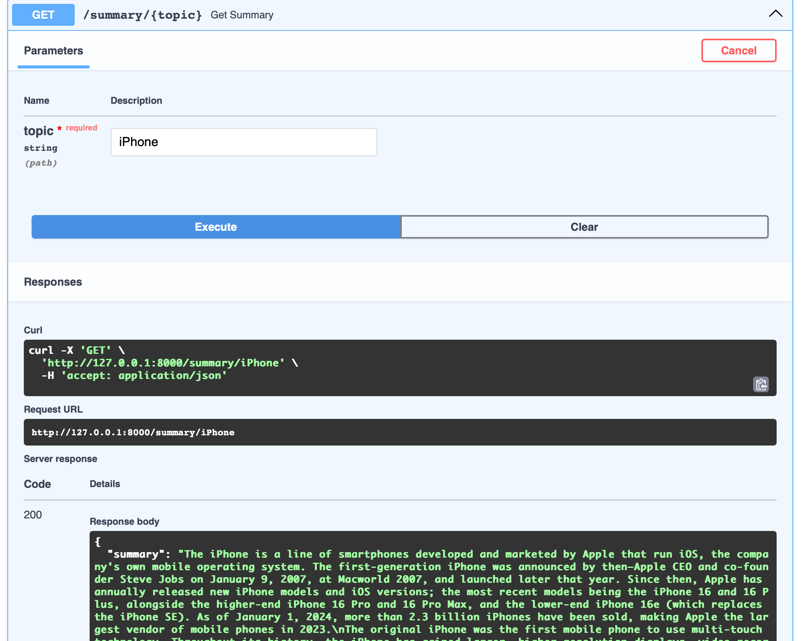
{  "friend": "Alice"}



**Шаг 8. Самостоятельное задание**







**Вывод:** В ходе лабораторной работы были освоены основы FastAPI, реализованы REST-эндпоинты с параметрами пути, запроса и тела, а также протестированы с помощью Swagger.