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 “Математической кибернетики и информационных технологий”

Лабораторная работа № 2

Выполнил: студент группы БВТ2306

Соколова Полина Кирилловна

Проверял: Лапин Виктор Андреевич

Москва

2023

Оглавление

1)Цель работы

2)Работа

3)Вывод

Цель работы

Цель данной работы является работа с FastAPI

Работа

1. Открываем консоль в Python и устанавливаем

pip install fastapi

pip install uvicorn // интерфейс прикладного программирования Python

pip install pyjokes // библиотека Python для однострочных шуток

1. Создаем файл main.py. Импортируем fastapi и pyjokes в main.py

from fastapi import FastAPI

import pyjokes

1. Создаем объект FastAPI, куда будут подключаться роуты.

app = FastAPI()

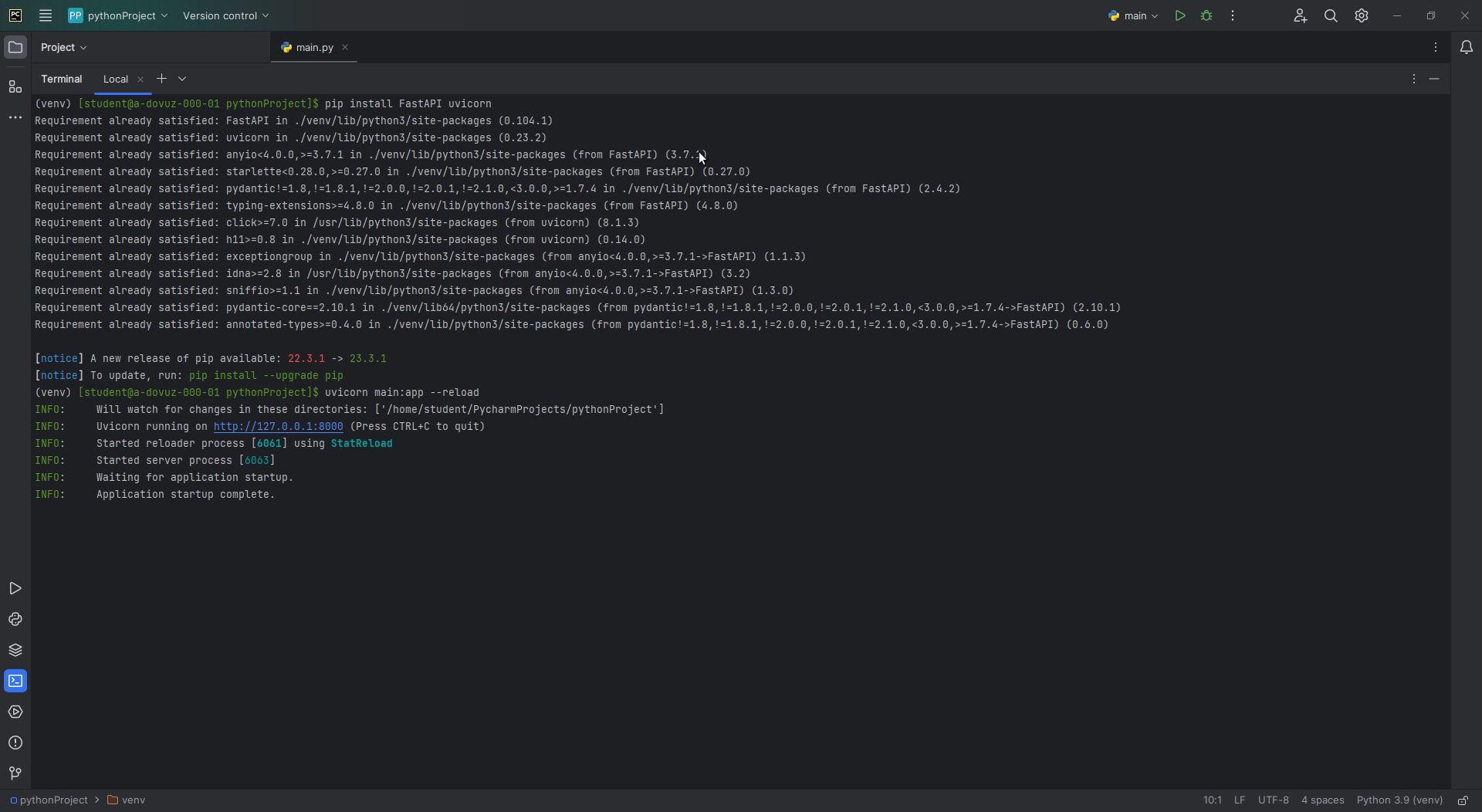
1. Создаем простой роут.

@app.get('/')

def joke():

    return pyjokes.get\_joke()

1. Для начала работы необходимо запустить unicorn (веб-сервер). В консоли пишем main:app –reload
2. Консоль выдает



После этого переходим по ссылке, и в адресной строке вбиваем /docs

1. Добавляем еще один роут ,где будет параметры пути, чтобы мы могли представить любую шутку от конкретного человека.

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

def friend\_joke(friend:str):

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

1. Добавляем еще один роут, где можно выбрать количество шуток.

@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

1. Есть другой способ передачи инфы в роут. Нужна библиотека pydantic класс BaseModel.

from pydantic import BaseModel

1. Создаем на его основе схему получение шутки, которую и передадим функции на входе.

class Joke(BaseModel):

    friend: str

    joke: str

class Jokelnput(BaseModel):

    friend: str

@app.post("/")

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

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

1. Схема позволяют и задавать образец ожидаемого ответа.

@app.post("/")

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

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

1. Добавим комментарий с описанием роута и поставим а параметрах respone\_model, чтобы добавить валидацию ответа функции.

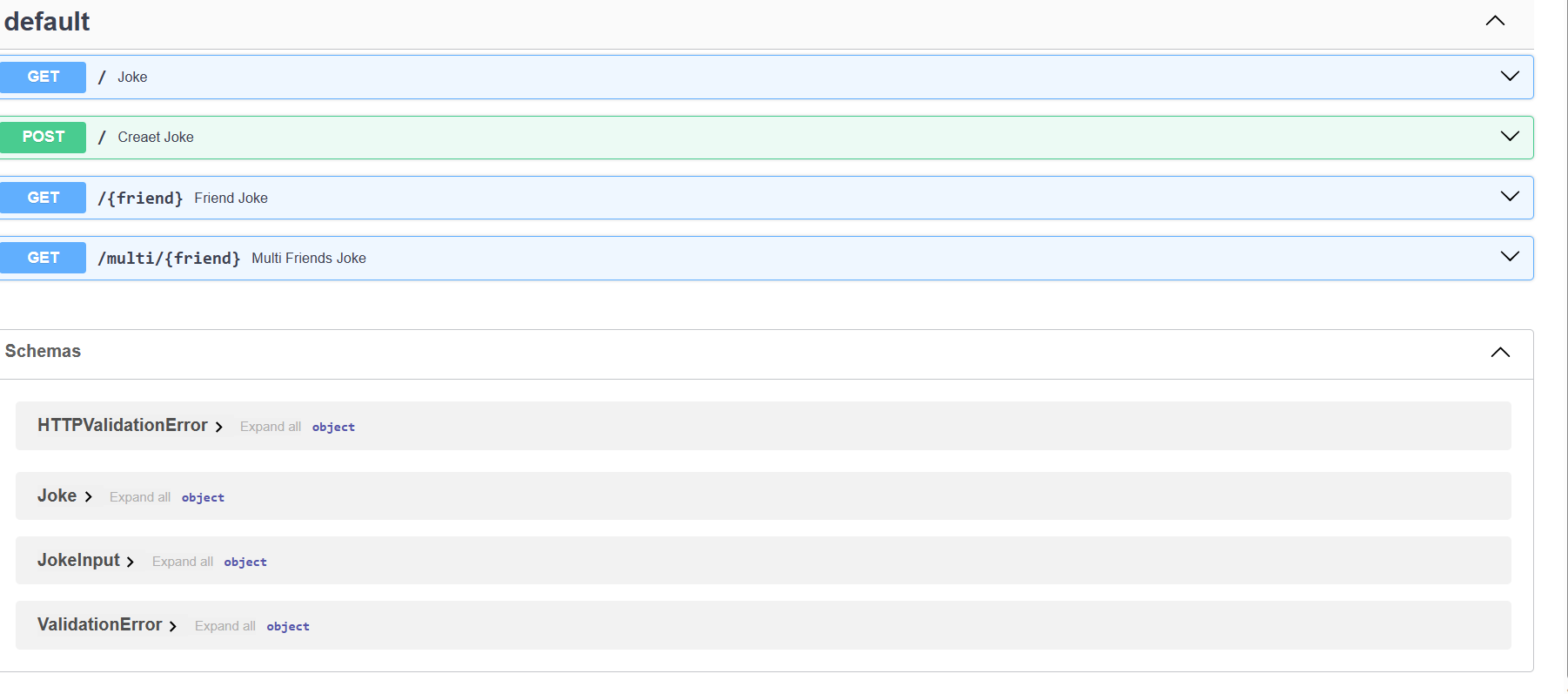
@app.post("/",response\_model=Joke)

def creaet\_joke(joke\_input:Jokelnput):

    """Создание шутки"""

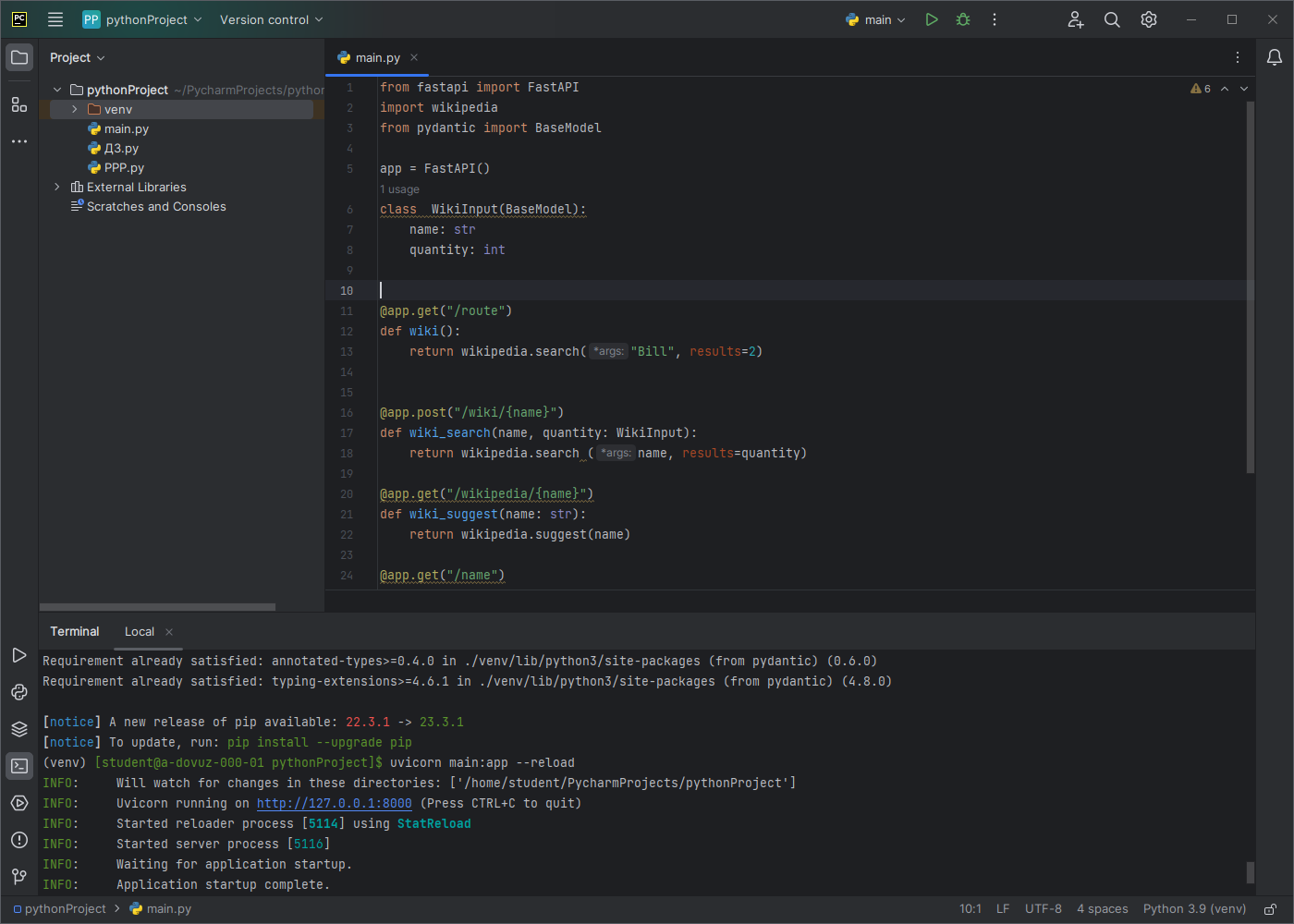
    return Joke(friend=joke\_input.friend,joke=pyjokes.get\_joke())

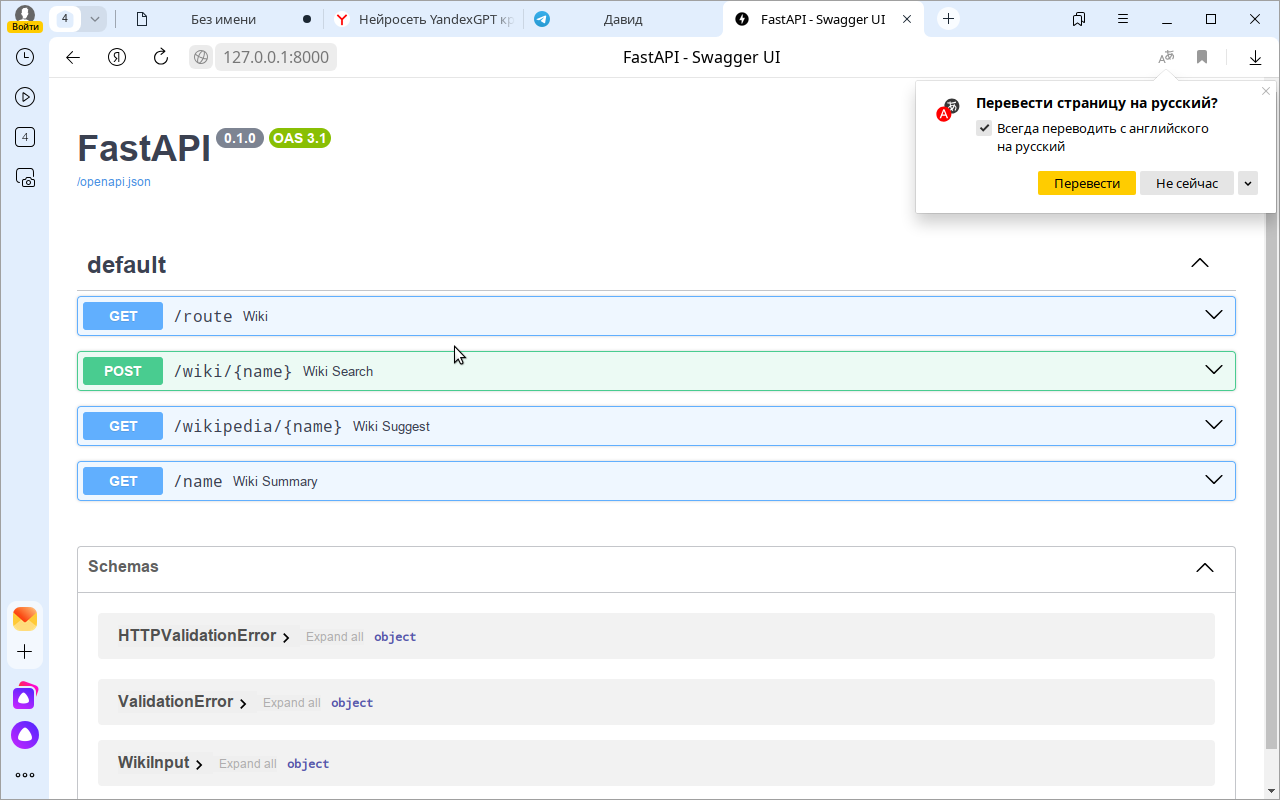
1. Что получилось в конце работы.



д/з

Создаем простой REST сервис, использующий библиотеку wikipedia.





Вывод

В данной лабораторной работе я научилась работать с FastAPI ,а именно с ее библиотеками и локальными серверами.