**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Лабораторная работа №15

# Основы работы с FastAPI.

Выполнил: Студент группы

БПИ2401

Костиль Антон

Москва

2025

Цель работы: Создать простой REST сервис, использующий библиотеку wikipedia. Создать 1 роут с параметром path, 1 роут с параметром query, 1 роут с передачей параметров в теле запроса. Все запросы должны возвращаться и валидирироваться по схемам.

Ход работы:

Шаг 1:  
Импорт нужных библиотек.

from fastapi import FastAPI

from pydantic import BaseModel

import wikipedia

Шаг 2:  
Создание объекта FastAPI

app = FastAPI()

Шаг 3:  
Создание Pydantic-схем для валидации и возвращения запроса

class ArticleRequest(BaseModel):

    title: str

class ArticleResponse(BaseModel):

    title: str

    summary: str

Шаг 4:  
Создание роута с PATH параметром

@app.get("/article/{title}", response\_model=ArticleResponse)

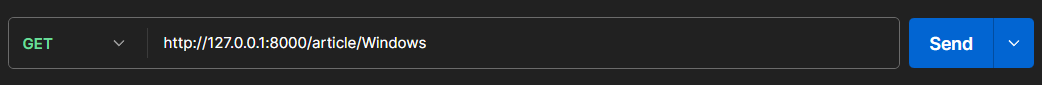
def get\_article\_by\_title(title: str):

    summary = wikipedia.summary(title)

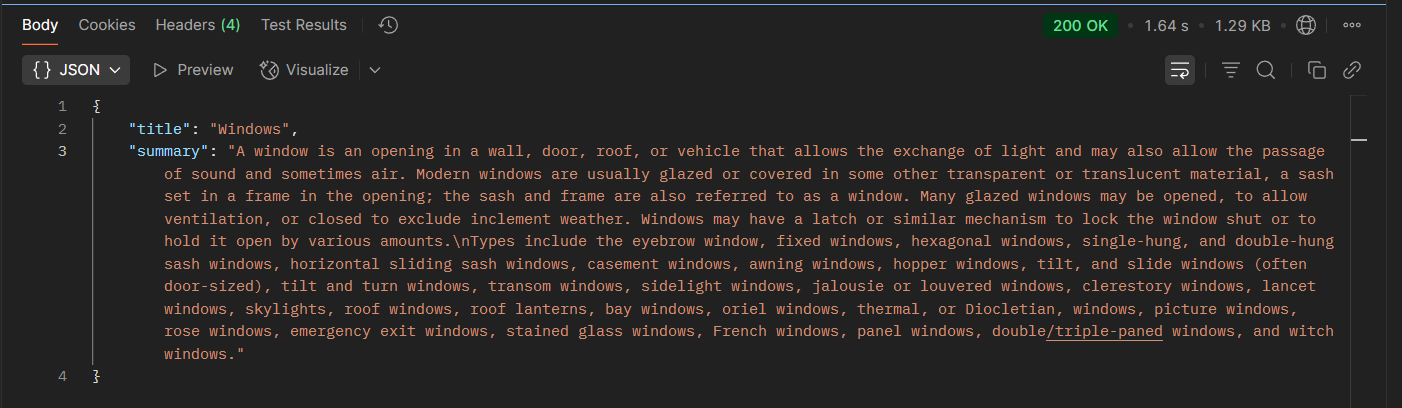
    return {"title": title, "summary": summary}

Тестирование через Postman:

Запрос:



Ответ:



Шаг 5:  
Создание роута с query-параметром

@app.get("/search", response\_model=ArticleResponse)

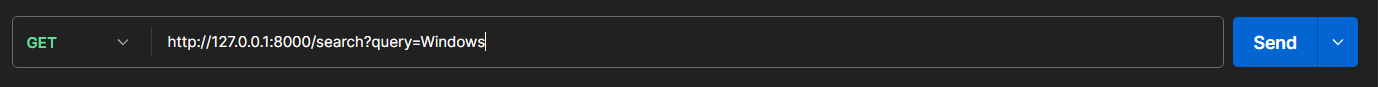
def search\_article(query: str):

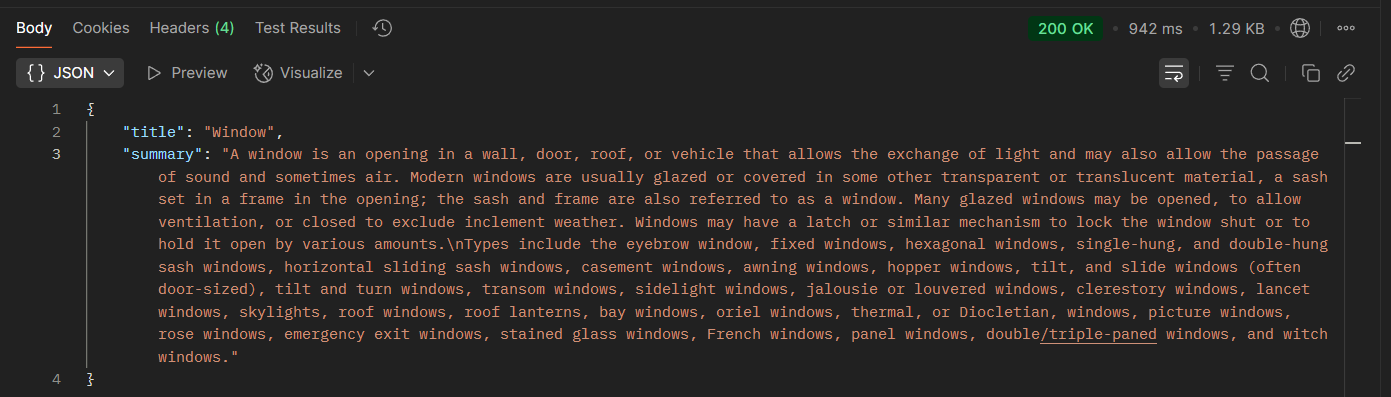
    page = wikipedia.page(query)

    return {"title": page.title, "summary": page.summary}

Тестирование через Postman:

Запрос:

  
Ответ:



Шаг 6:

Создание роута с передачей параметров в теле запроса

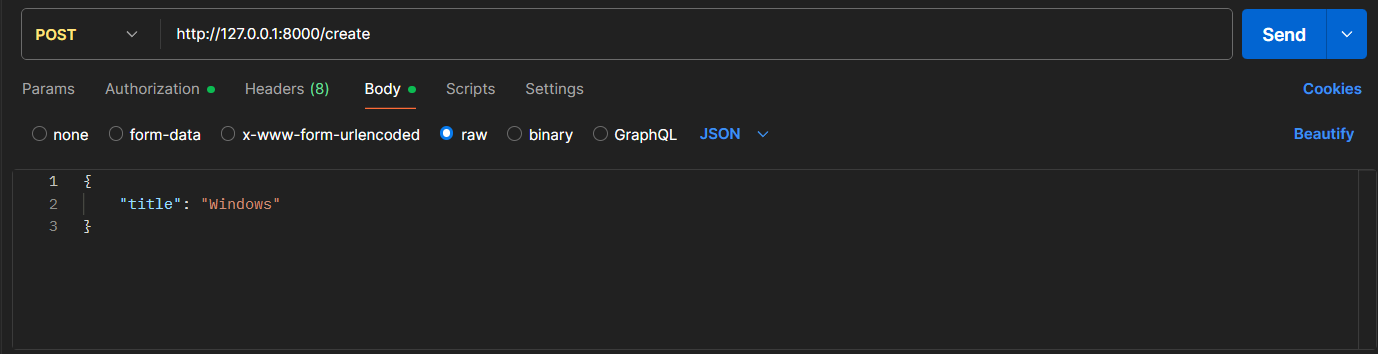
@app.post("/create", response\_model=ArticleResponse)

def create\_article(request: ArticleRequest):

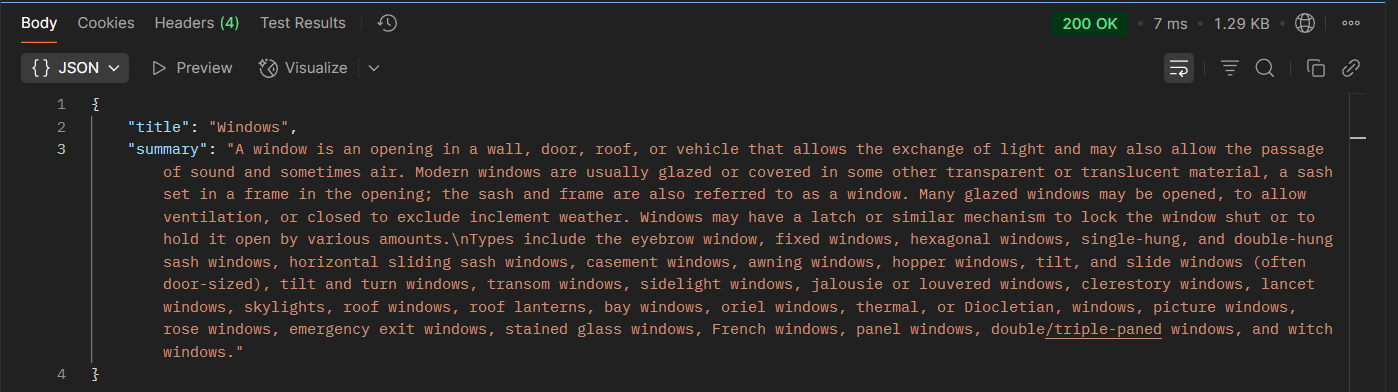
    summary = wikipedia.summary(request.title)

    return {"title": request.title, "summary": summary}

Тестирование через Postman:  
Запрос:



Ответ:



Вывод: Мной были написаны роуты с path, query параметрами, а также с передачей данных в теле запроса, и были успешно протестированы.