|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра практической и прикладной информатики (ППИ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Интерфейсы прикладного программирования»

**Практическое занятие № 5**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-01-19, Кузин Д.О.* | (подпись) | |
| Преподаватель | *Коновалова С.В.* | (подпись) | |
| Отчет представлен | « » 2021г. | |  | |

Москва 2021 г.

**Цель занятия:**

Целью данной работы является первичное знакомство студентов с интерфейсами прикладного программирования и их применением на практике.

**Результат работы:**

Листинг файла swagger.yaml

openapi: "3.0.0"

info:

version: 1.0.0

title: Shop imitation

license:

name: MIT

servers:

- url: http://localhost:8081

paths:

/products:

get:

description: Get list of all products.

operationId: getProductList

parameters:

- name: tags

in: query

description: tags to filter by

required: false

style: form

schema:

type: array

items:

type: string

- name: limit

in: query

description: list size limit

required: false

schema:

type: integer

format: int32

responses:

'200':

description: Product was succesfully found.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/Product'

default:

description: unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

post:

description: Creates a new product in repository

operationId: addProduct

requestBody:

description: Product to add to the repository

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/NewProduct'

responses:

'201':

description: Product was succesfully created.

content:

application/json:

schema:

$ref: '#/components/schemas/Product'

default:

description: unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

/products/{id}:

get:

description: Returns Product with a specified ID if it exists

operationId: findProductByID

parameters:

- name: id

in: path

description: ID of Product

required: true

schema:

type: integer

format: int64

responses:

'200':

description: Product with id was succesfully founded.

content:

application/json:

schema:

$ref: '#/components/schemas/Product'

default:

description: unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

delete:

description: deletes a single Product with specified ID from repository

operationId: deleteProduct

parameters:

- name: id

in: path

description: ID of Product to delete

required: true

schema:

type: integer

format: int64

responses:

'204':

description: Product was deleted from repository

default:

description: unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

put:

operationId: updateProduct

description: update Product with a specified ID

parameters:

- name: id

in: path

description: ID of Product to update

required: true

schema:

type: integer

format: int64

requestBody:

description: Product to update in the repository

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/NewProduct'

responses:

'201':

description: New Product was created

content:

application/json:

schema:

$ref: '#/components/schemas/Product'

default:

description: unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

components:

schemas:

Product:

allOf:

- $ref: '#/components/schemas/NewProduct'

- type: object

required:

- id

properties:

id:

type: integer

format: int64

NewProduct:

type: object

required:

- name

- count

properties:

name:

type: string

count:

type: integer

format: int32

Error:

type: object

required:

- code

- message

properties:

code:

type: integer

format: int32

message:

type: string

1. Демонстрация html страницы спецификации

redoc-cli bundle -o index.html swagger.yaml

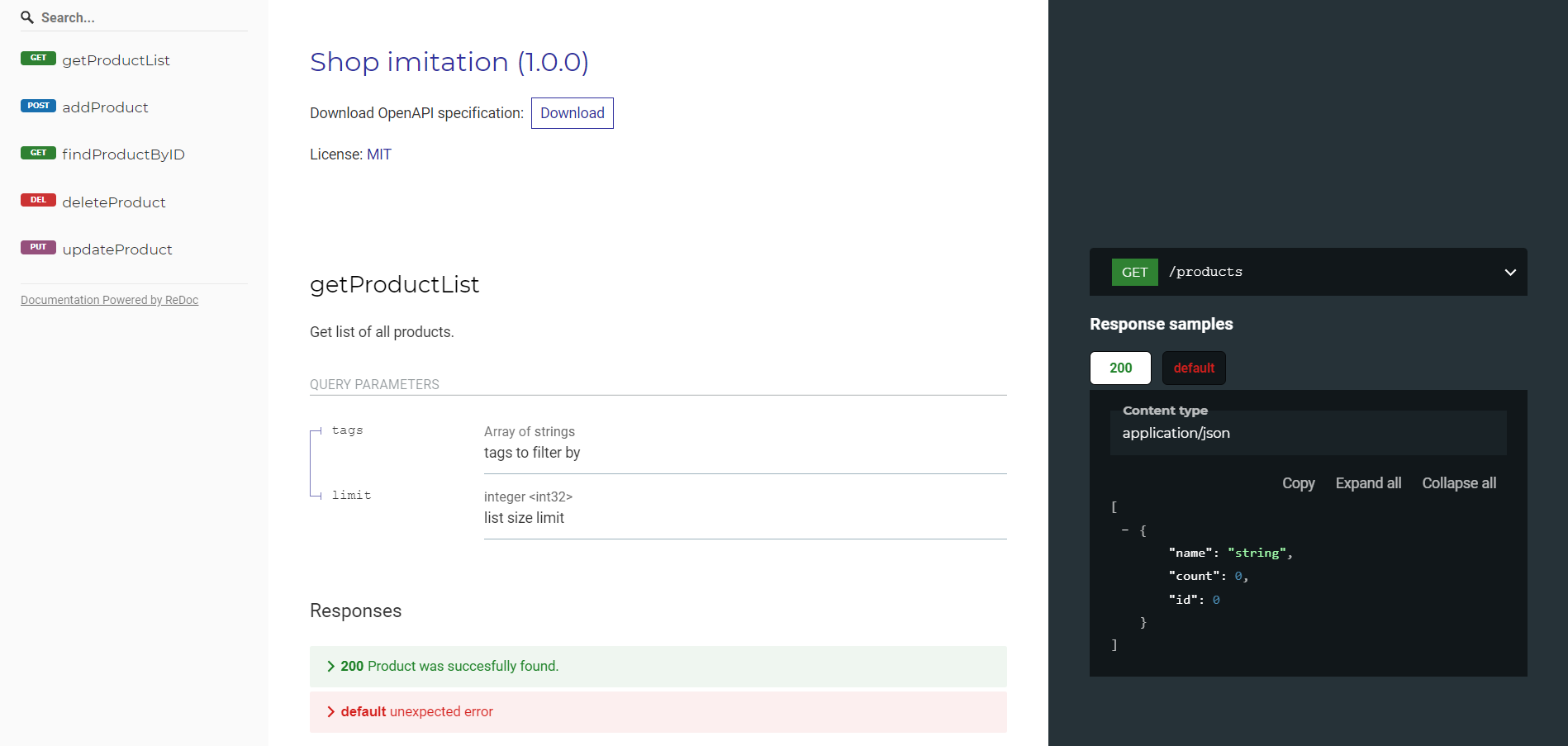


Рис. 1 – Демонстрация сгенерированного index.html

1. Запуск сервера с спецификацией

redoc-cli serve swagger.yaml

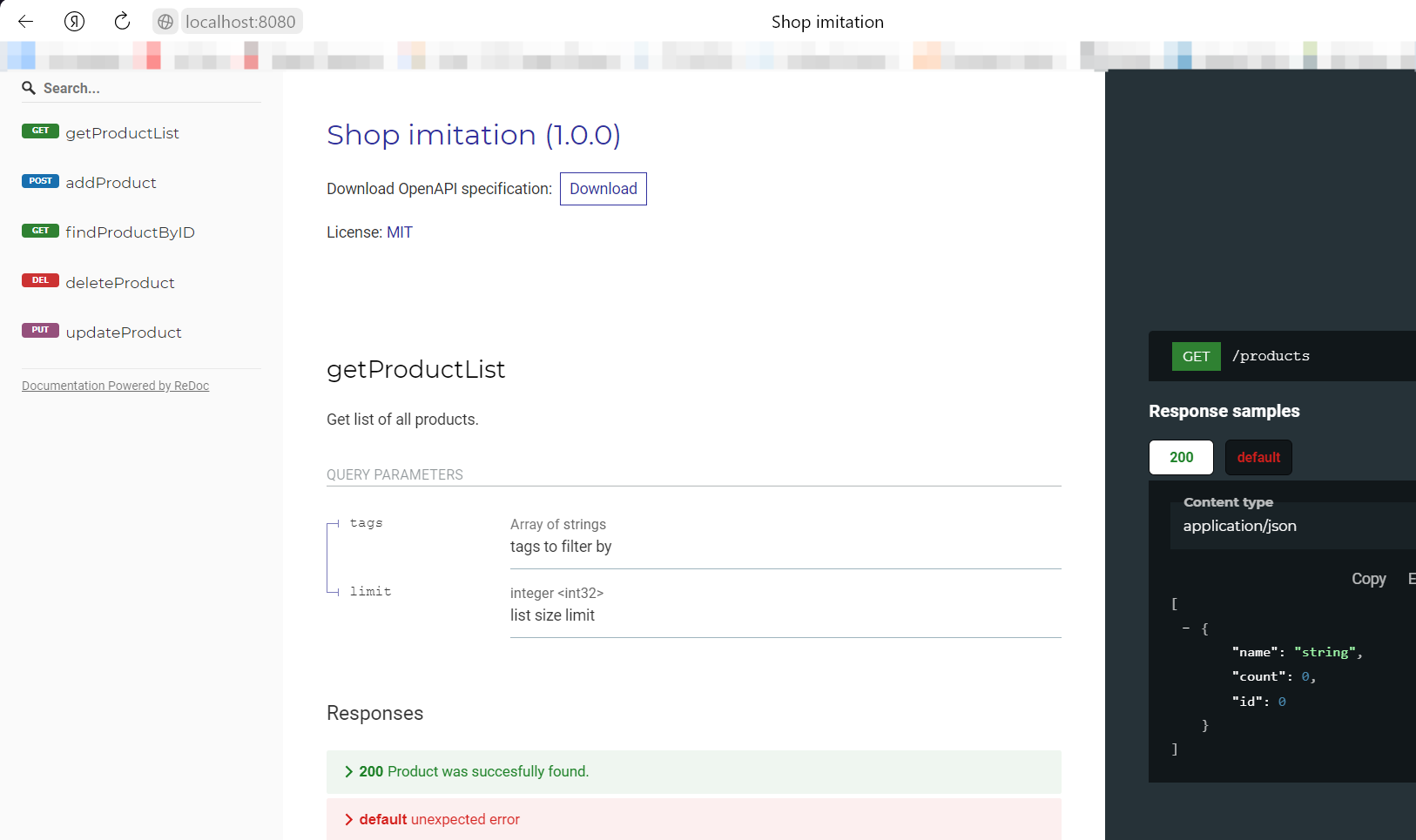


Рис. 2 – Демонстрация работы сервера с спецификацией

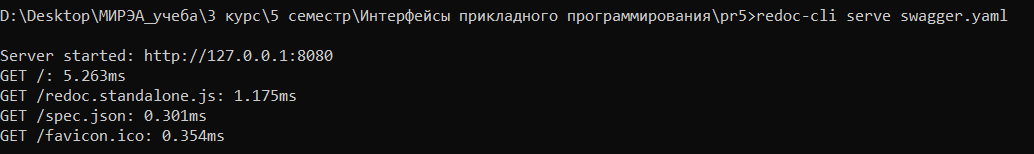


Рис. 3 – Демонстрация cmd

1. Запуск имитационного API

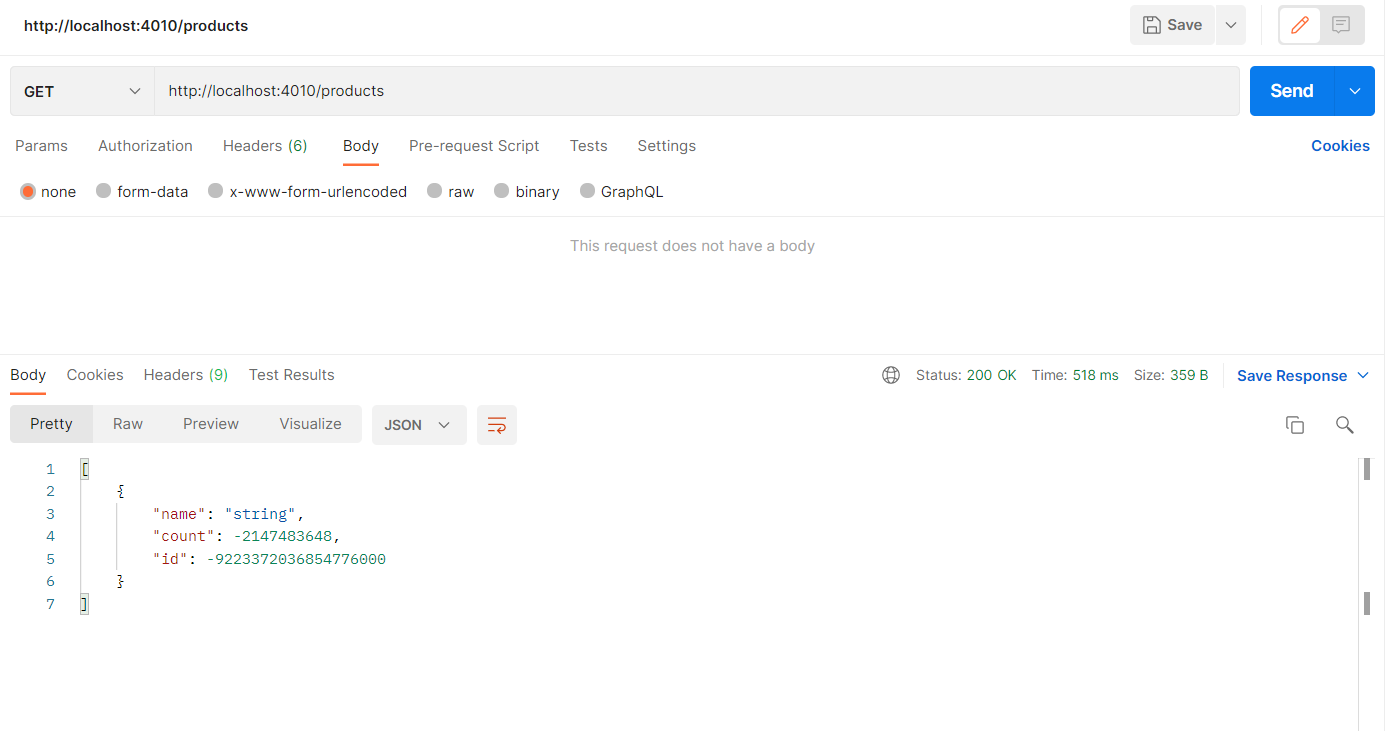


Рис. 4 – Демонстрация работы GET запроса

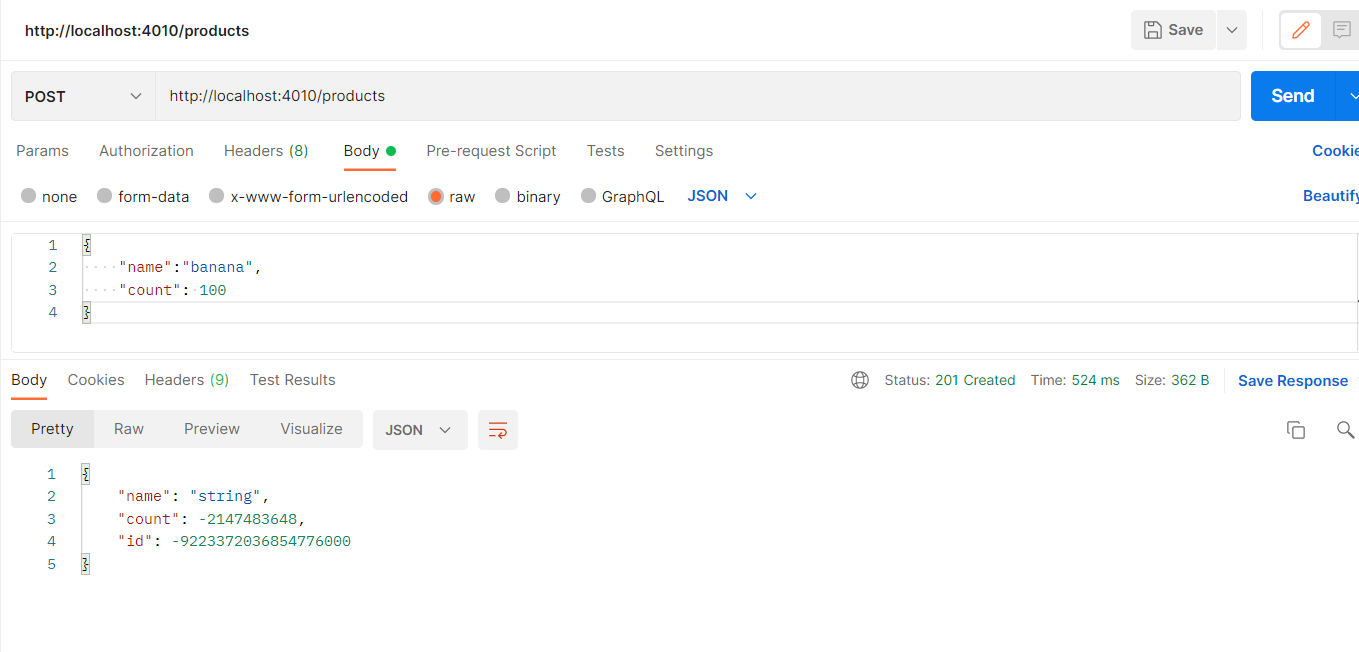


Рис. 5 – Демонстрация работы POST запроса

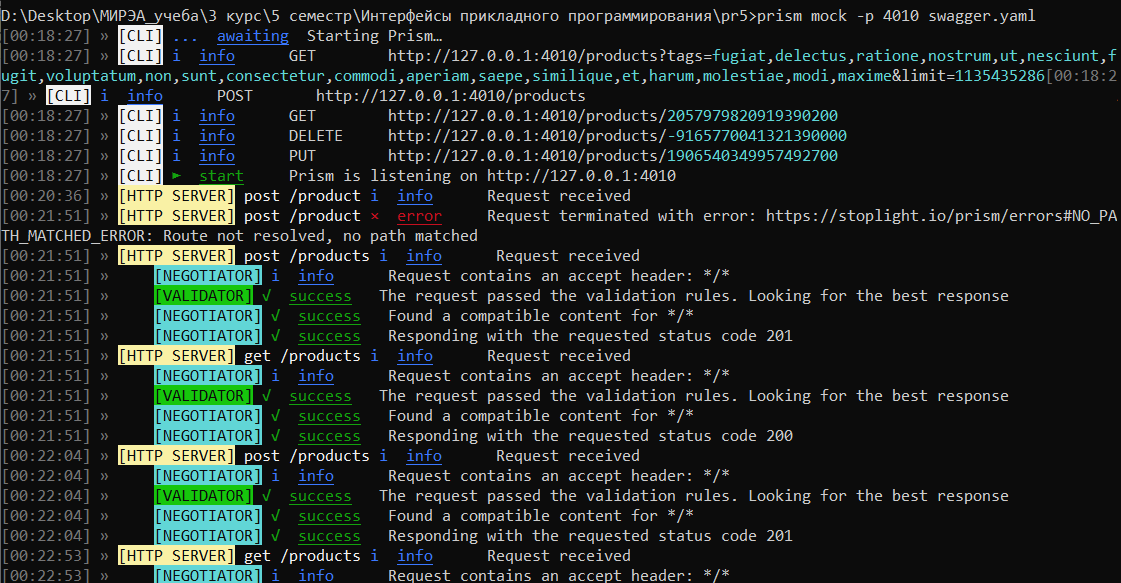


Рис. 6 – Демонстрация cmd

**Вывод:**

В ходе выполнения практической работы №5 были освоены практические навыки работы по разработке Rest API, а также по взаимодействию с этим API через Postman.