

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

**Отчет**

**Практическая работа №4-5**

**(Реализация через Swagger)**

**Выполнил:**

**Таначев Егор**

**Группа К33412**

**Проверил:**

**Добряков Д. И.**

**Санкт-Петербург**

**2023 г.**

## Задача

Документировать и протестировать API из Лабораторной работы №2 средствами Postman.

## Ход работы

Установим swagger в наш проект с помощью команд: *npm install swagger-ui-express* и *npm install @types/swagger-ui-express --save-dev*. После этого подключим пакет во входной точке приложения Core/index.ts при инициализации express, как показано на Рисунке 1.

```
33     private createApp(): express.Application {
34         const app = express()
35         app.use(cors())
36         app.use(bodyParser.json())
37         app.use(passport.initialize())
38         app.use('/v1', routes)
39
40         const swaggerDoc = require('../swagger.json')
41         app.use('/docs', swaggerUi.serve)
42         app.use('/docs', swaggerUi.setup(swaggerDoc))
43
44         return app
45     }
```

Рисунок 1 – Подключение пакет

Наша документация будет отображаться по адресу *http://localhost:8000/docs/*. После этого создадим файл *swagger.json* в папке *src*, здесь будет отображена структура нашей документации, как показано на Рисунке 2.

```
1  {
2      "swagger": "2.0",
3      "info": {
4          "version": "1.0.0",
5          "title": "REST API",
6          "description": ""
7      },
8      "host": "localhost:8000",
9      "basePath": "/v1",
```

```

10     "schemes": [
11         "http"
12     ],
13     "consumes": [
14         "application/json"
15     ],
16     "produces": [
17         "application/json"
18     ],
19     "paths": {
20 >         "/users/": { ...
56         },
57 >         "/users/id/:id": { ...
86         },
87 >         "/users/email/:email": { ...
116        },
117 >         "/users/all": { ...
146        },
147 >         "/users/login": { ...
184        },
185 >         "/users/profile": { ...
213        },
214 >         "/users/refresh": { ...
242        },
243 >         "/products": { ...
294        },
295 >         "/products/:id": { ...
349        },
350 >         "/supplies/supply": { ...
401        },
402 >         "/supplies/supply/:id": { ...
456        },
457 >         "/supplies/amount?product-id=:id": { ...
492        },

```

Рисунок 2 – Файл swagger.json

Запустим приложение и посмотрим документацию, как показано на Рисунке 3-7.

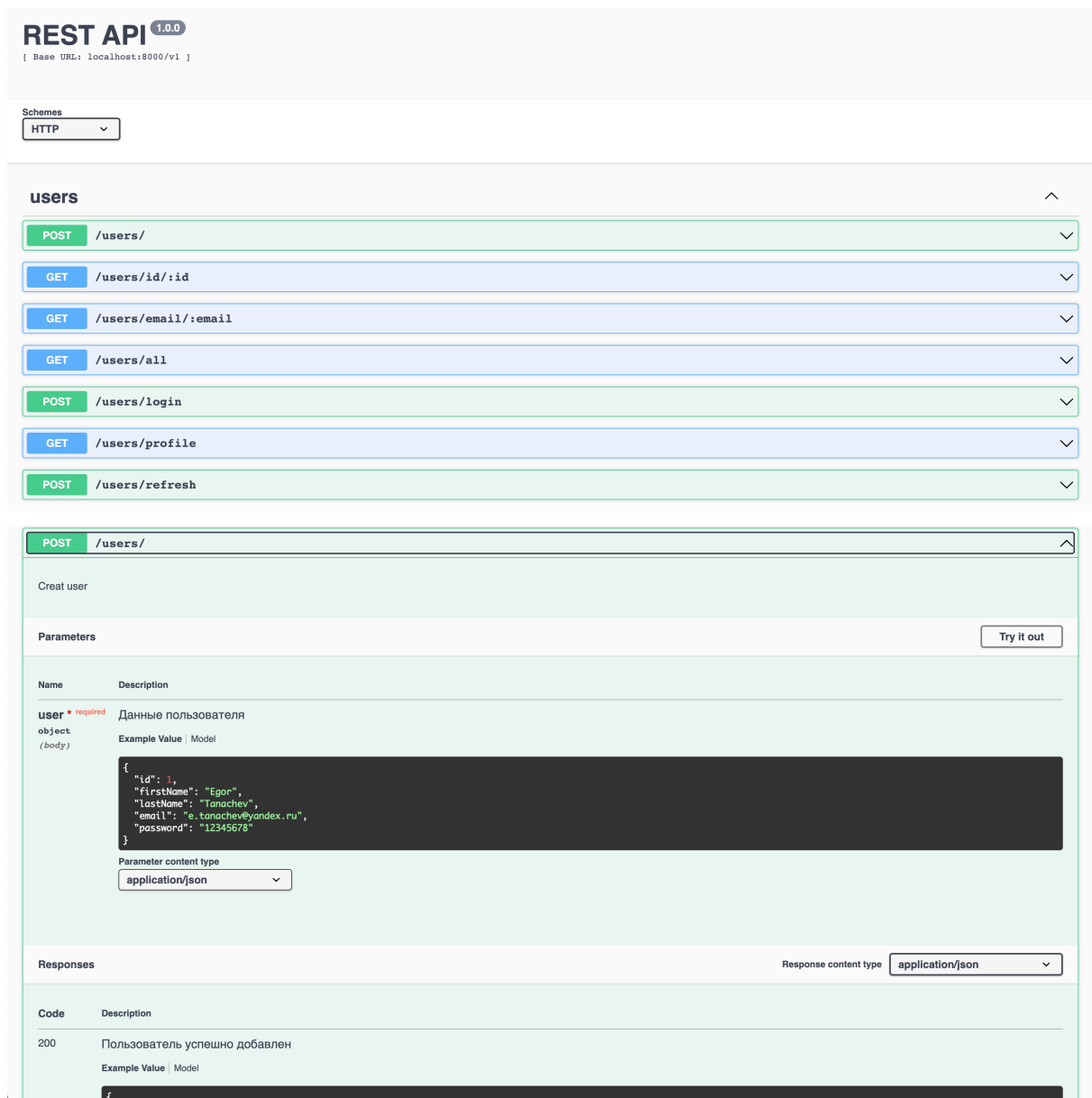


Рисунок 3 – Users

products

POST

/products

GET

/products

GET

/products/:id

DELETE

/products/:id

products

POST

/products

Add product

Parameters

Try it out

Name	Description
<b>product</b> <small>required</small> object (body)	Данные товара Example Value   Model

```
{
  "id": 1,
  "name": "Компьютерная мышь Logitech",
  "price": 3500
}
```

Parameter content type  
application/json

Responses

Response content type application/json

Code	Description
200	Товар успешно добавлен Example Value   Model

Рисунок 4 – Products

supplies

POST

/supplies/supply

GET

/supplies/supply

GET

/supplies/supply/:id

DELETE

/supplies/supply/:id

GET

/supplies/amount?product-id=:id

supplies

POST /supplies/supply

Add supply

Parameters
Try it out

Name	Description
<b>supply</b> <span>required</span> object (body)	Данные поставки Example Value   Model

```

{
  "id": 1,
  "count": 20,
  "productId": 2,
  "dateSupply": "2022-03-10T19:42:38.731Z"
}

```

Parameter content type  
application/json

Responses
Response content type application/json

Code	Description
200	Поставка успешно добавлена

Example Value | Model

Рисунок 5 – Supplies

realizations

POST /realizations/realization

GET /realizations/realization

GET /realizations/realization/:id

DELETE /realizations/realization/:id

realizations

POST /realizations/realization

Add realization

Parameters
Try it out

Name	Description
<b>realization</b> <span>required</span> object (body)	Данные продажи Example Value   Model

```

{
  "id": 1,
  "count": 3,
  "price": 12500,
  "productId": 2,
  "dateRealization": "2022-05-13T19:42:38.731Z"
}

```

Parameter content type  
application/json

Responses
Response content type application/json

Code	Description
200	Продажа успешно добавлена

Example Value | Model

Рисунок 6 – Realizations

