

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3

Выполнила:

Гусейнова Марьям

БР 1.2

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- реализовать автодокументирование средствами swagger;
- реализовать документацию API средствами Postman.

Ход работы

Для реализации автодокументирования средствами Swagger было сделано следующее:

1. Применены следующие команды к проекту (использовался swagger-jsdoc):

```
npm install swagger-ui-express swagger-jsdoc --save
```

```
npm install @types/swagger-ui-express @types/swagger-jsdoc --save-dev
```

2. В главный файл (index.ts) были добавлены следующие строчки:

```
import swaggerUi from 'swagger-ui-express';
import swaggerJsdoc from 'swagger-jsdoc';

// Настройки Swagger
const options = {
  definition: {
    openapi: '3.0.0',
    info: {
      title: 'Fitness Platform API',
      version: '1.0.0',
      description: 'API для платформы фитнес-тренировок',
    },
  },
  components: {
    securitySchemes: {
      bearerAuth: {
        type: 'http',
        scheme: 'bearer',
        bearerFormat: 'JWT'
      }
    }
  },
  tags: [
    { name: 'Authentication', description: 'Авторизация и регистрация' },
    { name: 'Users', description: 'Управление пользователями' },
    { name: 'CurrentProgress', description: 'Управление общим прогрессом' },
    { name: 'WorkoutPlans', description: 'Управление планами тренировок' },
    { name: 'PlanProgress', description: 'Управление прогрессом планов' },
    { name: 'Workouts', description: 'Управление тренировками' },
    { name: 'WorkoutInPlans', description: 'Управление тренировками в планах' },
    { name: 'Exercises', description: 'Управление упражнениями' },
  ],
};
```

```

        { name: 'ExerciseWorkouts', description: 'Управление упражнениями в
тренировках' },
        { name: 'ExerciseProgress', description: 'Управление прогрессом упражнений' },
        { name: 'BlogPosts', description: 'Управление блогом' }
    ]
},
apis: [
    './src/routes/*.ts',
    './src/swagger/*.ts'
]
};

```

```

const swaggerSpec = swaggerJsdoc(options);
console.log(swaggerSpec);

// Подключение Swagger UI к маршруту /api-docs
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpec));

```

3. Был добавлен файл src/swagger/swagger-schemas.ts

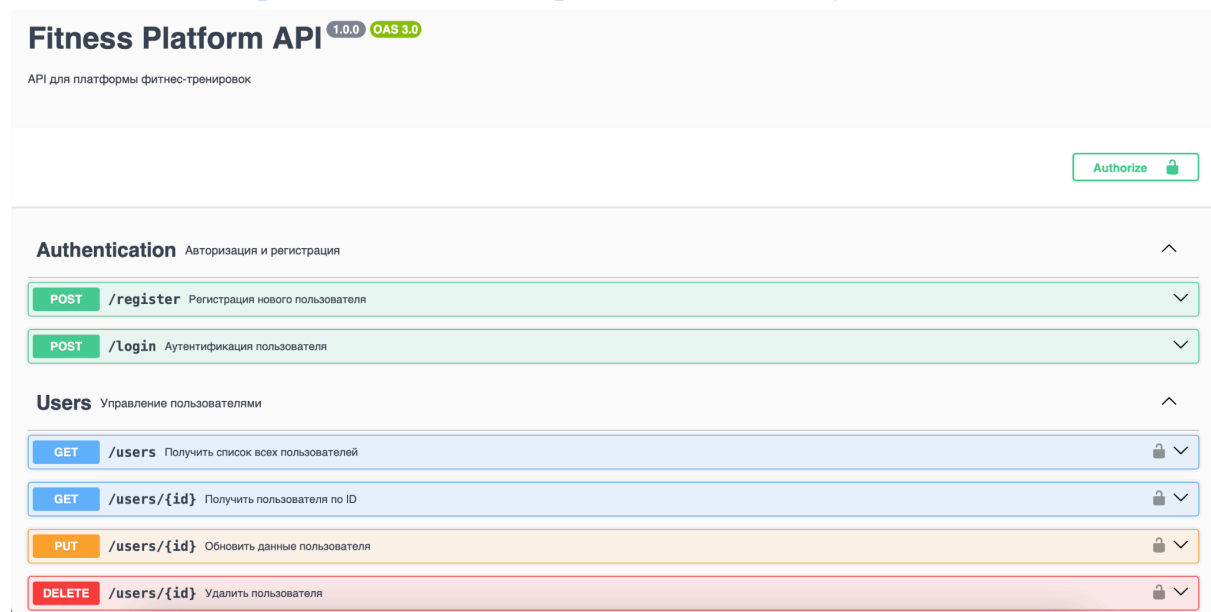
4. В файлы ./routes/*.ts были добавлены @openapi-аннотации, пример:

```

/**
 * @openapi
 * /register:
 *   post:
 *     tags: [Authentication]
 *     summary: Регистрация нового пользователя
 *     requestBody:
 *       required: true
 *       content:
 *         application/json:
 *           schema:
 *             $ref: '#/components/schemas/RegisterUser'
 *     responses:
 *       201:
 *         description: Пользователь успешно зарегистрирован
 *         content:
 *           application/json:
 *             schema:
 *               $ref: '#/components/schemas/AuthResponse'
 *       400:
 *         description: Пользователь с таким email уже существует
 */

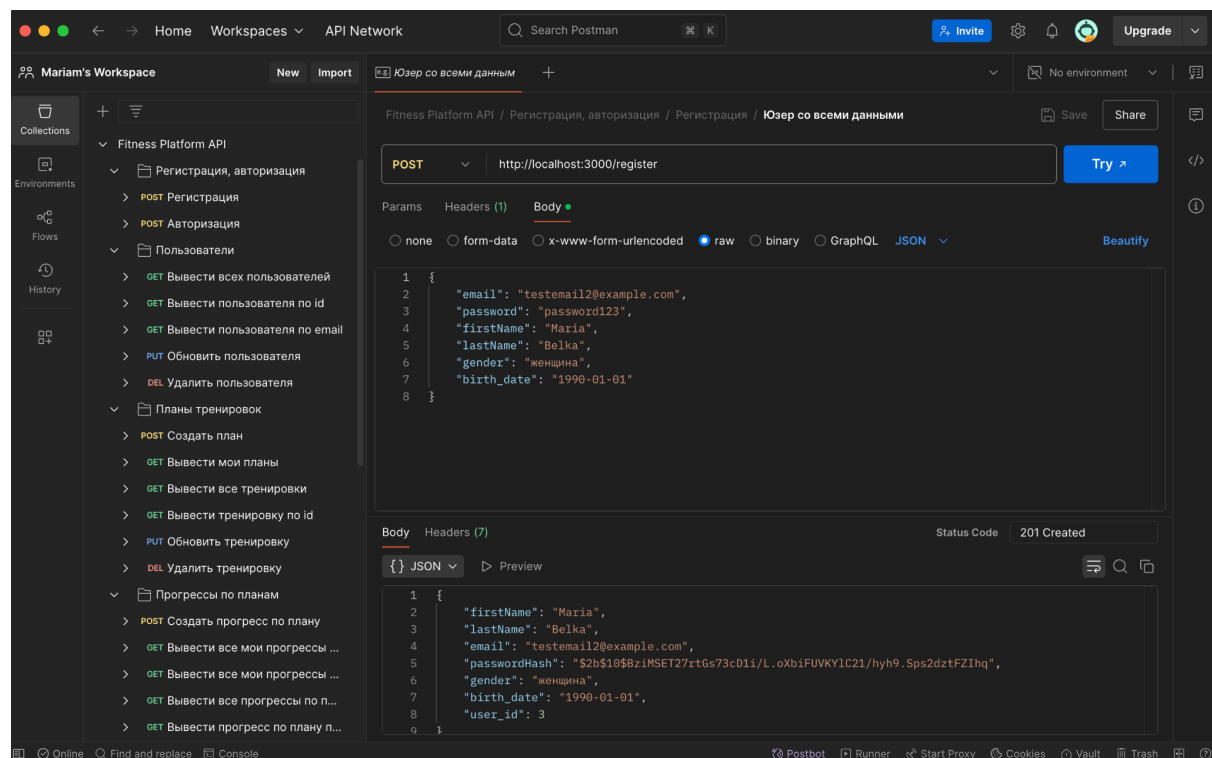
```

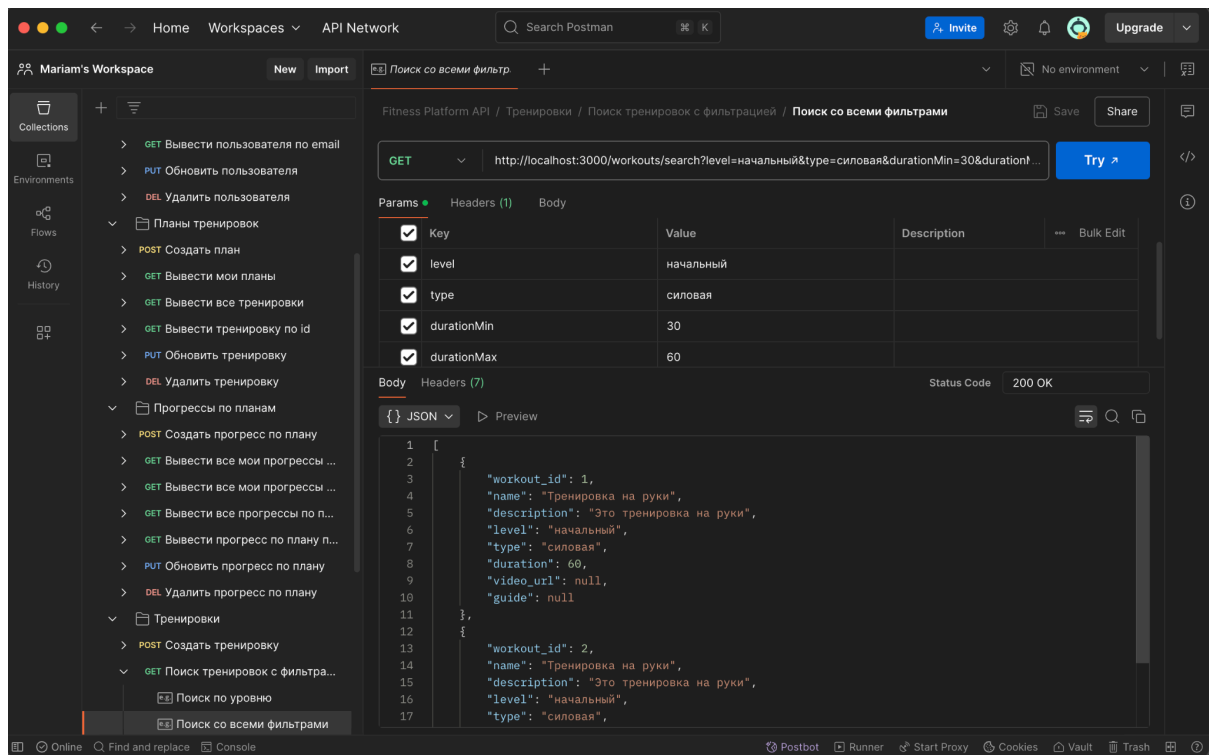
5. По ссылке <http://localhost:3000/api-docs/> было получено:



Для реализации документации средствами Postman было сделано следующее:

1. Создана коллекция Fitness Platform API
2. Добавлены запросы в коллекцию
3. Добавлено описание к коллекции





Вывод

В ходе данной лабораторной работы были реализованы документации средствами swagger и Postman.