САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №3
Документирование API

Выполнил:
Даньшин Семён
К3340

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Реализовать автодокументирование средствами swagger и документацию API средствами Postman.

Ход работы

1. Реализация Swagger документации

В проекте используется автоматическая генерация Swagger документации из protobuf файлов с помощью protoc-gen-openapiv2.

Генерация OpenAPI спецификации:

```
protoc --openapiv2_out=./api \
    --openapiv2_opt=logtostderr=true \
    --openapiv2_opt=json_names_for_fields=false \
    api/workouts/workouts.proto
```

2. Структура API документации

Swagger документация включает следующие разделы:

Основная информация:

```
title: Fitness Trainer API
version: 1.0.0
baseUrl: /api
description: API for fitness Trainer service
```

Сервисы и эндпоинты:

AuthService - аутентификация:

- `POST /v1/auth/login` - вход в систему
- `POST /v1/auth/refresh` - обновление токена
- `POST /v1/auth/logout` - выход из системы

UserService - управление пользователями:

- `GET /v1/users/me` - получение текущего пользователя
- `POST /v1/users` - создание пользователя
- `PUT /v1/users` - обновление данных пользователя
- `GET /v1/users/{userId}` - получение пользователя по ID

ExerciseService - управление упражнениями:

- `GET /v1/exercises` - получение списка упражнений
- `POST /v1/exercises` - создание нового упражнения
- `GET /v1/exercises/{exerciseId}` - получение деталей упражнения

- `GET /v1/exercises/{exerciseId}/alternatives` - альтернативные упражнения
- `GET /v1/exercises/{exerciseId}/history` - история выполнения

RoutineService - управление планами тренировок:

- `GET /v1/routines` - получение списка рутин
- `POST /v1/routines` - создание новой рутины
- `GET /v1/routines/{routineId}` - получение деталей рутины
- `PUT /v1/routines/{routineId}` - обновление рутины
- `DELETE /v1/routines/{routineId}` - удаление рутины

WorkoutService - управление тренировками:

- `GET /v1/workouts` - получение списка тренировок
- `POST /v1/workouts` - начало новой тренировки
- `GET /v1/workouts/active` - активные тренировки
- `GET /v1/workouts/{workoutId}` - получение тренировки
- `POST /v1/workouts/{workoutId}/complete` - завершение тренировки

3. Модели данных в Swagger

Документация включает полное описание всех моделей данных:

Пример модели User:

```
WorkoutUser:
  type: object
  properties:
    id:
      type: string
    createdAt:
      type: string
      format: date-time
    email:
      type: string
    firstName:
      type: string
    lastName:
      type: string
    height:
      type: number
      format: float
    weight:
      type: number
      format: float
```

4. Безопасность в документации

Документированы методы аутентификации:

- Bearer token authentication
- Refresh token mechanism
- Session management

5. Postman коллекция

Создана полная Postman коллекция с примерами запросов:

Структура коллекции:

```
Fitness Trainer API/
├── Auth/
│   ├── Login
│   ├── Refresh Token
│   └── Logout
├── Users/
│   ├── Get Current User
│   ├── Create User
│   ├── Update User
│   └── Get User by ID
├── Exercises/
│   ├── Get Exercises
│   ├── Create Exercise
│   ├── Get Exercise Details
│   └── Get Exercise History
├── Routines/
│   ├── Get Routines
│   ├── Create Routine
│   ├── Get Routine Details
│   └── Update Routine
└── Workouts/
    ├── Start Workout
    ├── Get Active Workouts
    ├── Get Workout Details
    └── Complete Workout
```

6. Примеры запросов в Postman

Создание пользователя:

```
POST /v1/users
{
  "email": "user@example.com",
  "password": "securepassword",
  "firstName": "John",
  "lastName": "Doe"
}
```

Начало тренировки:

```
POST /v1/workouts
{
  "routineId": "uuid-here",
  "generateWorkout": false
}
```

7. Переменные окружения в Postman

Настроены переменные для разных окружений:

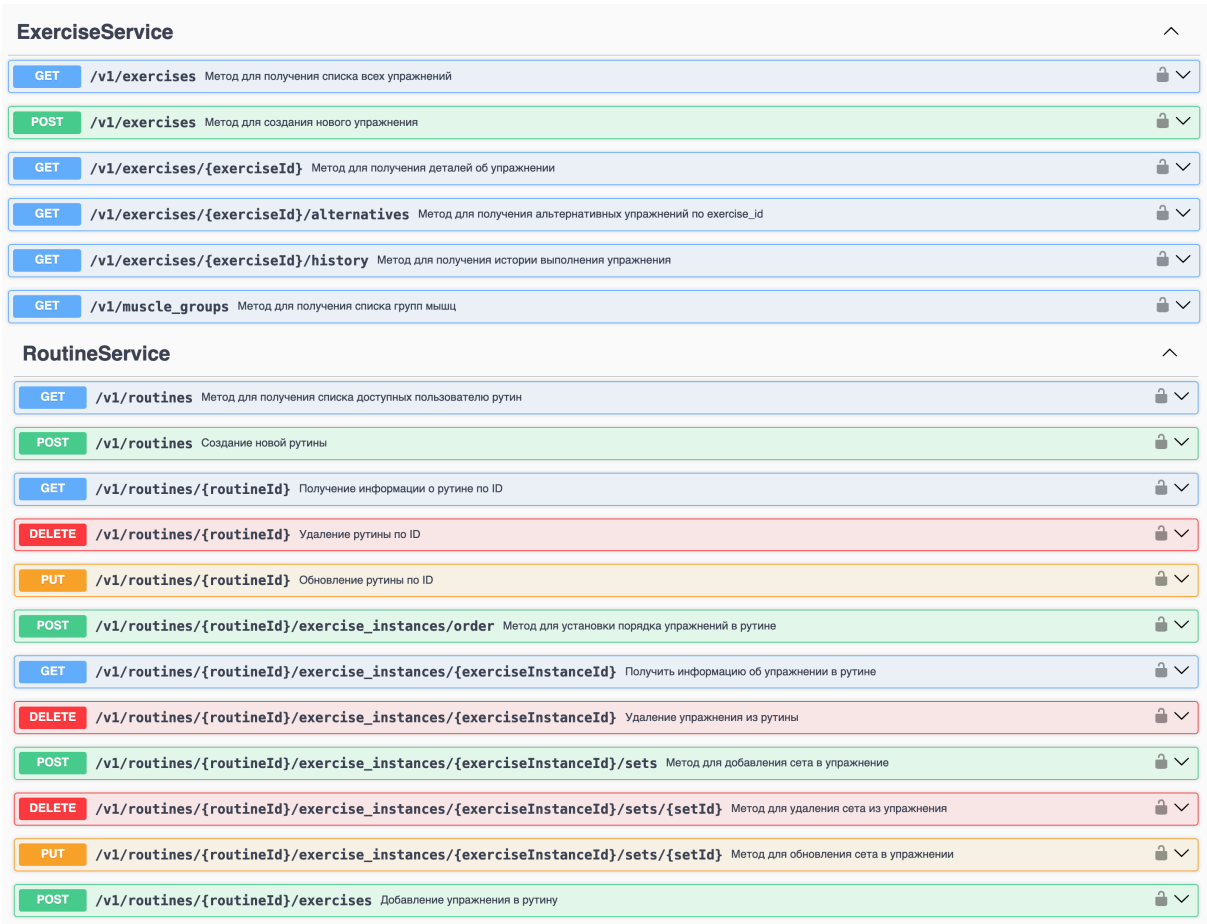
- `baseUrl` - базовый URL API

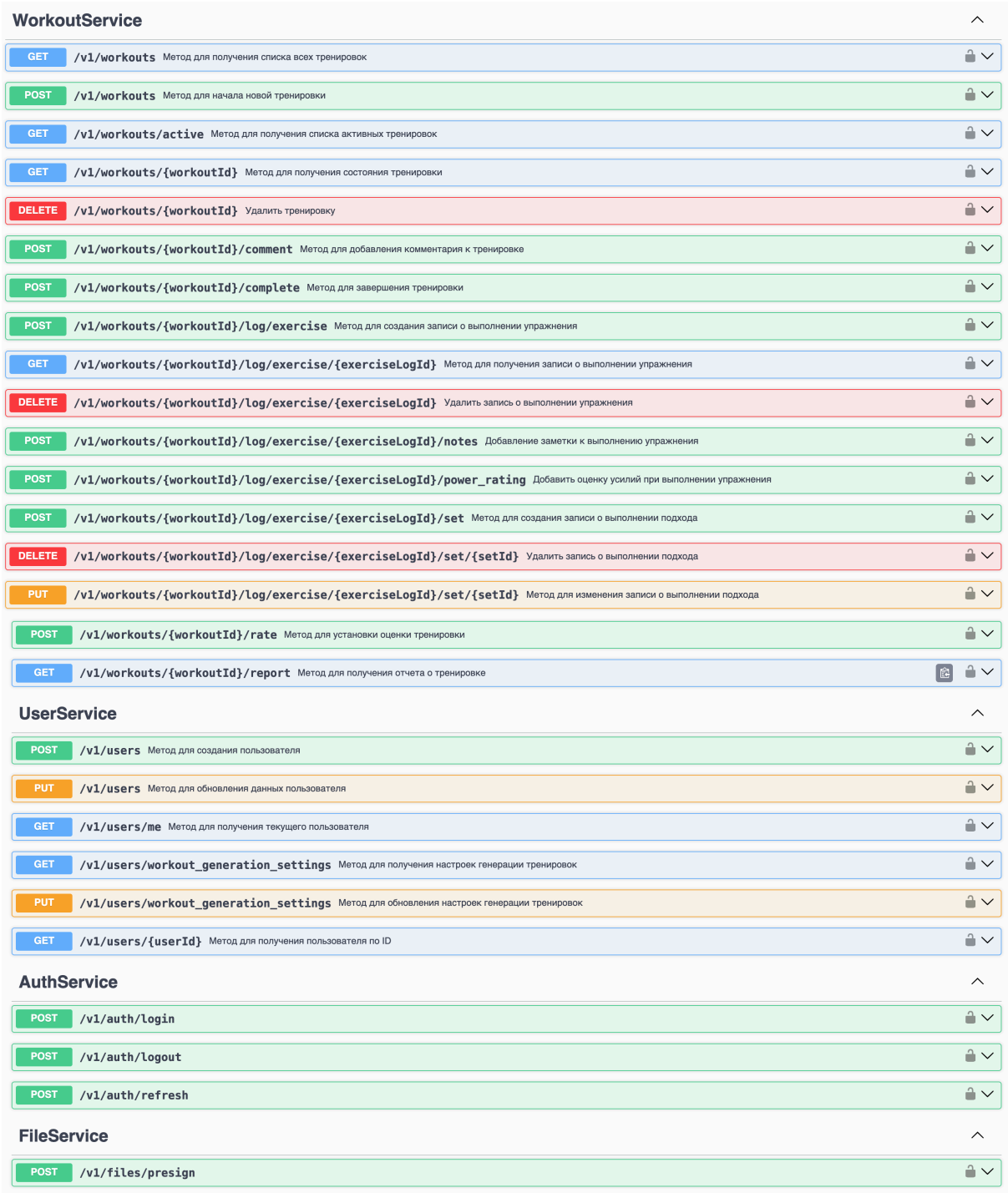
- `accessToken` - токен доступа
- `userId` - ID текущего пользователя

8. Автотесты в Postman

Добавлены базовые автотесты для проверки:

- Статус кодов ответов
- Структуры ответов
- Наличия обязательных полей





9. Доступ к документации

- Swagger UI доступен по адресу: `/api/docs`
- OpenAPI спецификация: `/api/openapi.json`
- Postman коллекция экспортирована в формате JSON

Вывод

Реализована полная документация API проекта:

1. **Автоматическая генерация** Swagger документации из protobuf схем
2. **Полное покрытие** всех эндпоинтов с описанием параметров и ответов
3. **Детальное описание** моделей данных и их валидации
4. **Postman коллекция** с примерами всех запросов и автотестами
5. **Настройка окружений** для разработки и продакшена
6. **Документирование безопасности** и методов аутентификации

Документация обеспечивает удобную разработку и интеграцию с API, а также служит справочным материалом для разработчиков.