

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашнее задание №4

Выполнил:
Григорян Самвел Ашотович
группа К3340

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задание

1. Реализовать тестирование API средствами Postman
2. Написать тесты внутри Postman для всех эндпоинтов системы управления недвижимостью

Ход работы

1. Создание коллекции тестов Postman

1.1 Структура коллекции

Создана коллекция "RealtyRent API Tests".

```
{
  "info": {
    "name": "RealtyRent API Tests",
    "description": "Полный набор API-тестов для системы управления недвижимостью",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "Health Check",
      "item": [
        {
          "name": "Health Check",
          "request": {
            "method": "GET",
            "header": [],
            "url": {
              "raw": "{{baseUrl}}/../health",
              "host": ["{{baseUrl}}"],
              "path": [".", "health"]
            }
          },
          "response": [],
          "event": [
            {
              "listen": "test",
              "script": {
                "exec": [
                  "pm.test(\"Status code is 200\", function () {",
                  "    pm.response.to.have.status(200);",
                  "});",
                  ""
                ],
                "prerequest": [
                  "pm.test(\"Response has status field\", function () {",

```

```

        "    const jsonData = pm.response.json();",
        "    pm.expect(jsonData).to.have.property('status');",
        "    pm.expect(jsonData.status).to.eql('OK');",
        "});",
        "",
        "pm.test(\"Response has timestamp field\", function () {",
        "    const jsonData = pm.response.json();",
        "    pm.expect(jsonData).to.have.property('timestamp');",
        "});"
    ]
  }
}
]
},
...
}

```

1.2 Настройка переменных окружения

Создан файл окружения

[RealtyRent_API_Environment.postman_environment.json](#) с переменными:

```

{
  "id": "realty-rent-env",
  "name": "RealtyRent API Environment",
  "values": [
    {
      "key": "baseUrl",
      "value": "http://localhost:3000/api",
      "type": "default",
      "enabled": true
    },
    {
      "key": "authToken",
      "value": "",
      "type": "secret",
      "enabled": true
    },
    {
      "key": "testUserId",
      "value": "",
      "type": "default",
      "enabled": true
    },
    {
      "key": "testBuildingId",
      "value": "",

```

```
    "type": "default",
    "enabled": true
  },
  {
    "key": "testApartmentId",
    "value": "",
    "type": "default",
    "enabled": true
  },
  {
    "key": "testContractId",
    "value": "",
    "type": "default",
    "enabled": true
  },
  {
    "key": "adminUsername",
    "value": "admin",
    "type": "default",
    "enabled": true
  },
  {
    "key": "adminPassword",
    "value": "admin123",
    "type": "secret",
    "enabled": true
  }
],
"_postman_variable_scope": "environment",
"_postman_exported_at": "2024-01-01T00:00:00.000Z",
"_postman_exported_using": "Postman/10.0.0"
}
```

2. Документация тестов

2.1 Создание README для тестов

Создан файл [API_TESTS_README.md](#) с подробной инструкцией по:

- Установке и настройке Postman
- Импорту коллекции и окружения
- Запуску тестов
- Интерпретации результатов
- Устранению неполадок

2.2 Примеры тестовых данных

Создан файл `test_data_examples.json` с примерами:

- Валидных данных для всех сущностей
- Невалидных данных для тестирования ошибок
- Сценариев тестирования
- Ожидаемых ответов

Вывод

В рамках выполнения домашнего задания №4 была успешно реализована комплексная система тестирования API с использованием Postman. Созданы тесты для всех эндпоинтов системы (Users, Buildings, Apartments, Contracts, Auth) с проверкой как успешных сценариев, так и обработки ошибок. Система тестов легко адаптируется для разных окружений (development, staging, production) через переменные окружения.

Проект готов к использованию в реальной разработке и может быть легко расширен дополнительными тестами по мере развития API.