Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический

университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

Отчет о практической работе №4

«Тестирование REST API и знакомство с Postman»

по дисциплине

«Сопровождение программных систем»

Выполнили:

Студенты группы 045

Анохин В.А.

Вашкулатов Н.А.

Проверили:

асс. каф. ЭВМ Баранова С.Н.

**Цель работы**: получение знаний об особенностях тестирования REST API, а также базовых умений и навыков использования инструмента Postman.

**Ход работы**

1. Изучите теоретическую часть лабораторной работы.

2. Запустите Postman.

3. Создайте коллекцию для запросов к SWAPI.

4. Создайте переменную коллекции с именем base\_url и значением https://swapi.dev/api/. Далее используйте эту переменную в запросах к SWAPI. На рисунке 1 показана созданная переменная.

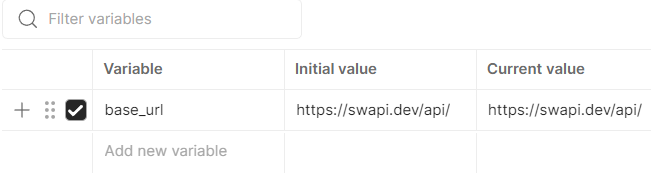


Рисунок 1 – Переменная коллекции

5. Запросите информацию о планетах. Выясните, что означают атрибуты rotation\_period и orbital\_period.

**Запрос:** GET {{base\_url}}/planets/

**Фрагмент ответа:**

{

    "count": 60,

    "next": "https://swapi.dev/api/planets/?page=2",

    "previous": **null**,

    "results": [

        {

            "name": "Tatooine",

            "rotation\_period": "23",

            "orbital\_period": "304",

            "diameter": "10465",

            "climate": "arid",

            "gravity": "1 standard",

            "terrain": "desert",

            "surface\_water": "1",

            "population": "200000",

            "residents": [

                "https://swapi.dev/api/people/1/",

                "https://swapi.dev/api/people/2/",

                "https://swapi.dev/api/people/4/",

                "https://swapi.dev/api/people/6/",

                "https://swapi.dev/api/people/7/",

                "https://swapi.dev/api/people/8/",

                "https://swapi.dev/api/people/9/",

                "https://swapi.dev/api/people/11/",

                "https://swapi.dev/api/people/43/",

                "https://swapi.dev/api/people/62/"

            ],

            "films": [

                "https://swapi.dev/api/films/1/",

                "https://swapi.dev/api/films/3/",

                "https://swapi.dev/api/films/4/",

                "https://swapi.dev/api/films/5/",

                "https://swapi.dev/api/films/6/"

            ],

            "created": "2014-12-09T13:50:49.641000Z",

            "edited": "2014-12-20T20:58:18.411000Z",

            "url": "https://swapi.dev/api/planets/1/"

        },

Параметр rotation\_period показывает количество часов в сутках на планете, а orbital\_period показывает длину года в сутках.

6. Запросите подробную информацию обо всех звездолетах.

**Запрос:** GET {{base\_url}}/starships

**Фрагмент ответа:**

{

    "count": 36,

    "next": "https://swapi.dev/api/starships/?page=2",

    "previous": **null**,

    "results": [

        {

            "name": "CR90 corvette",

            "model": "CR90 corvette",

            "manufacturer": "Corellian Engineering Corporation",

            "cost\_in\_credits": "3500000",

            "length": "150",

            "max\_atmosphering\_speed": "950",

            "crew": "30-165",

            "passengers": "600",

            "cargo\_capacity": "3000000",

            "consumables": "1 year",

            "hyperdrive\_rating": "2.0",

            "MGLT": "60",

            "starship\_class": "corvette",

            "pilots": [],

            "films": [

                "https://swapi.dev/api/films/1/",

                "https://swapi.dev/api/films/3/",

                "https://swapi.dev/api/films/6/"

            ],

            "created": "2014-12-10T14:20:33.369000Z",

            "edited": "2014-12-20T21:23:49.867000Z",

            "url": "https://swapi.dev/api/starships/2/"

        },

7. Теперь выполните тот же запрос, но с параметром, позволяющим получить вторую страницу ответа от сервера. Проверьте с помощью теста, что запрос выполнен успешно. На рисунке 2 показан результат теста.

**Запрос:** GET {{base\_url}}/starships?page=2

**Тест:**

pm.test("Status code is 200", function () {

    pm.response.to.have.status(200);

});

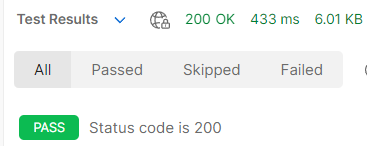


Рисунок 2 – Результат теста

8. Запросите все звездолеты, в названии или модели которых есть подстрока Star или star. Проверьте с помощью теста, пришел ли ответ за 300 мс. На рисунке 3 показан результаты теста.

**Запрос:** GET {{base\_url}}/starships?search=star

**Фрагмент ответа:**

{

    "count": 15,

    "next": "https://swapi.dev/api/starships/?search=star&page=2",

    "previous": **null**,

    "results": [

        {

            "name": "Star Destroyer",

            "model": "Imperial I-class Star Destroyer",

            "manufacturer": "Kuat Drive Yards",

            "cost\_in\_credits": "150000000",

            "length": "1,600",

            "max\_atmosphering\_speed": "975",

            "crew": "47,060",

            "passengers": "n/a",

            "cargo\_capacity": "36000000",

            "consumables": "2 years",

            "hyperdrive\_rating": "2.0",

            "MGLT": "60",

            "starship\_class": "Star Destroyer",

            "pilots": [],

            "films": [

                "https://swapi.dev/api/films/1/",

                "https://swapi.dev/api/films/2/",

                "https://swapi.dev/api/films/3/"

            ],

            "created": "2014-12-10T15:08:19.848000Z",

            "edited": "2014-12-20T21:23:49.870000Z",

            "url": "https://swapi.dev/api/starships/3/"

        },

**Тест:**

pm.test("Response time is less than 300ms", function () {

    pm.expect(pm.response.responseTime).to.be.below(300);

});

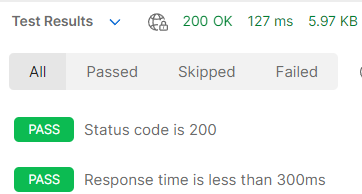


Рисунок 3 – Результат теста

9. Запросите подробную информацию обо всех звездолетах на языке wookiee.

**Запрос:** GET {{base\_url}}/starships?format=wookiee

**Фрагмент ответа:**

{

    "oaoohuwhao": 36,

    "whwokao": "acaoaoakc://cohraakah.wawoho/raakah/caorarccacahakc/?akrarrwo=2&wwoorcscraao=ohooooorahwowo",

    "akrcwohoahoohuc":whhuanan,

    "rcwochuanaoc": [

        {

            "whrascwo": "CR90 oaoorchowoaoaowo",

            "scoowawoan": "CR90 oaoorchowoaoaowo",

            "scrawhhuwwraoaaohurcworc": "Coorcwoananahrawh Ewhrrahwhwoworcahwhrr Coorcakoorcraaoahoowh",

            "oaoocao\_ahwh\_oarcwowaahaoc": "3500000",

            "anwowhrraoac": "150",

            "scrak\_raaoscoocakacworcahwhrr\_cakwowowa": "950",

            "oarcwooh": "30-165",

            "akraccwowhrrworcc": "600",

            "oararcrroo\_oaraakraoaahaoro": "3000000",

            "oaoowhchuscrarhanwoc": "1 roworarc",

            "acroakworcwarcahhowo\_rcraaoahwhrr": "2.0",

            "MGLT": "60",

            "caorarccacahak\_oaanracc": "oaoorchowoaoaowo",

            "akahanooaoc": [],

            "wwahanscc": [

                "acaoaoakc://cohraakah.wawoho/raakah/wwahanscc/1/",

                "acaoaoakc://cohraakah.wawoho/raakah/wwahanscc/3/",

                "acaoaoakc://cohraakah.wawoho/raakah/wwahanscc/6/"

            ],

            "oarcworaaowowa": "2014-12-10T14:20:33.369000Z",

            "wowaahaowowa": "2014-12-20T21:23:49.867000Z",

            "hurcan": "acaoaoakc://cohraakah.wawoho/raakah/caorarccacahakc/2/"

        },

10. Проверьте с помощью теста, что ответ от сервера содержит подстроку warcworawawhoohurracao. На рисунке 4 показан результаты теста.

**Тест:**

pm.test("Body matches string", function () {

    pm.expect(pm.response.text()).to.include("warcworawawhoohurracao");

});

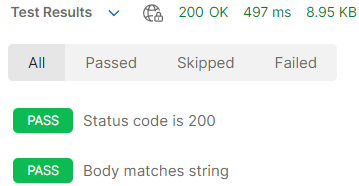


Рисунок 4 – Результат теста

11. Запустите вместе все три последних запроса с тестами. На рисунке 5 показаны результаты запуска коллекции запросов.

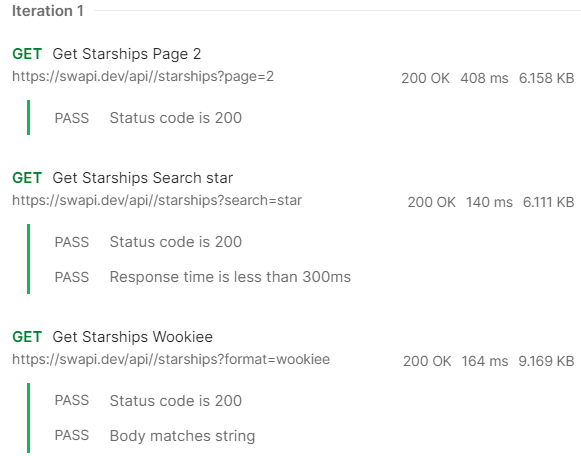


Рисунок 5 – Результат теста

12. Выберите персонажа из таблицы согласно своему номеру варианта задания. Дальнейшие пункты выполняются для заданного персонажа. По согласованию с преподавателем возможен выбор какого-либо другого персонажа, представленного в SWAPI.

Вариант 8: Ayla Secura.

13. Найдите персонажа с помощью запроса с параметром. Узнайте рост, вес и цвет глаз своего персонажа.

**Запрос:** GET {{base\_url}}/people/?search=Ayla Secura

**Ответ:**

{

    "count": 1,

    "next": **null**,

    "previous": **null**,

    "results": [

        {

            "name": "Ayla Secura",

            "height": "178",

            "mass": "55",

            "hair\_color": "none",

            "skin\_color": "blue",

            "eye\_color": "hazel",

            "birth\_year": "48BBY",

            "gender": "female",

            "homeworld": "https://swapi.dev/api/planets/37/",

            "films": [

                "https://swapi.dev/api/films/4/",

                "https://swapi.dev/api/films/5/",

                "https://swapi.dev/api/films/6/"

            ],

            "species": [

                "https://swapi.dev/api/species/15/"

            ],

            "vehicles": [],

            "starships": [],

            "created": "2014-12-20T09:48:01.172000Z",

            "edited": "2014-12-20T21:17:50.409000Z",

            "url": "https://swapi.dev/api/people/46/"

        }

    ]

}

Рост персонажа: 178, масса: 55, цвет глаз: карий. На рисунке 6 показан вывод этих данных в тесте.

**Тест:**

var jsonData = pm.response.json();

pm.test("People is Ayla Secura. Info: height: " + jsonData.results[0].height +

    ", mass: " + jsonData.results[0].mass +

    ", eye color: " + jsonData.results[0].eye\_color, function () {

    pm.expect(jsonData.results[0].name).to.eql("Ayla Secura");

});



Рисунок 6 – Результат теста

14. Определите с помощью запроса название родной планеты своего персонажа. Определите следующие сведения об этой планете: длительность суток в часах, длительность года в сутках, климат.

Название родной планеты: Ryloth. На рисунке 7 показан результат теста, проверяющий название планеты.

**Тест:**

pm.sendRequest(jsonData.results[0].homeworld, function (err, response) {

    var planet = response.json();

    pm.test("Homeworld planet is Ryloth", function () {

        pm.expect(planet.name).to.eql("Ryloth");

    });

});



Рисунок 7 – Результат теста

Длительность суток на планете: 30 часов, длительность года: 305 суток, климат: умеренный, засушливый, субарктический. На рисунке 9 показаны результаты тестов, проверяющие данные показатели.

**Тесты:**

var jsonData = pm.response.json();

pm.test("Planet rotation period: 30", function () {

    pm.expect(jsonData.rotation\_period).to.eql("30");

});

pm.test("Planet orbital period: 305", function () {

    pm.expect(jsonData.orbital\_period).to.eql("305");

});

pm.test("Planet climate: temperate, arid, subartic", function () {

    pm.expect(jsonData.climate).to.eql("temperate, arid, subartic");

});

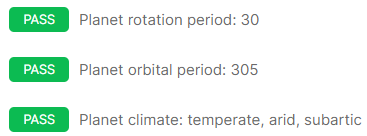


Рисунок 8 – Результат теста

15. Узнайте, в скольких фильмах участвовал персонаж.

Персонаж участвовал в трёх фильмах. На рисунке 9 показан результат теста, проверяющий количество фильмов.

**Тест:**

pm.test("Number of films - 3", function () {

    pm.expect(jsonData.results[0].films.length).to.eql(3);

});



Рисунок 9 – Результат теста

16. Определите с помощью запроса родной язык у персонажа, если о нем есть сведения.

Родной язык персонажа: Twi'leki. На рисунке 10 показан результат теста, проверяющий родной язык.

**Тест:**

pm.sendRequest(jsonData.results[0].species[0], function (err, response) {

    var specie = response.json();

    pm.test("Language is Twi'leki", function () {

        pm.expect(specie.language).to.eql("Twi'leki");

    });

});



Рисунок 10 – Результат теста

17. Определите с помощью запроса максимальную скорость в атмосфере для одного из транспортных средств персонажа.

Выбранный персонаж не имеет транспортных средств. На рисунке 11 показан результат теста, проверяющий отсутствие любого транспорта.

**Тест:**

pm.test(“People does not have transport”, function () {

    pm.expect(jsonData.results[0].vehicles.length

    + jsonData.results[0].starships.length).to.eql(0);

});



Рисунок 11 – Результат теста

18. Создайте с помощью сниппетов по 3 теста для каждого запроса о персонаже. Запустите все тесты для запросов о персонаже вместе и убедитесь в их успешном завершении. На рисунке 12 показаны результаты запуска коллекции запросов.

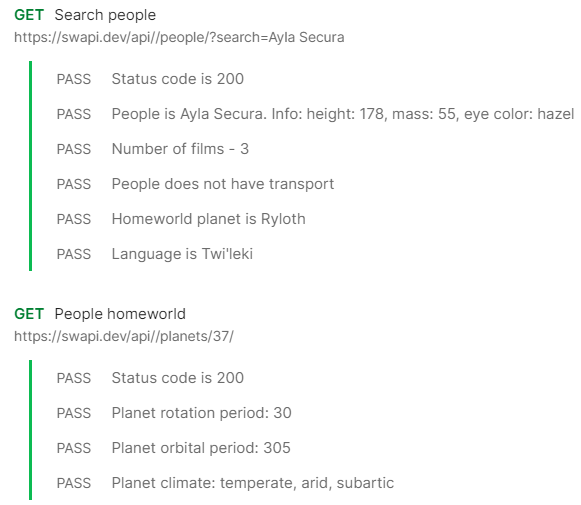


Рисунок 12 – Результат теста

**Вывод**: в ходе выполнения работы были получены знания об особенностях тестирования REST API, а также базовые умения и навыки использования инструмента Postman.