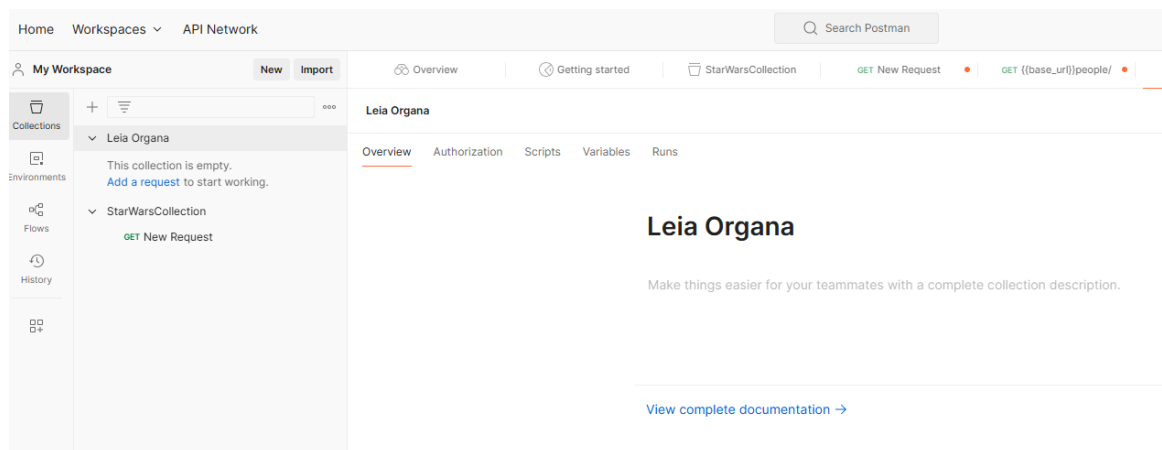


Практическая работа №4

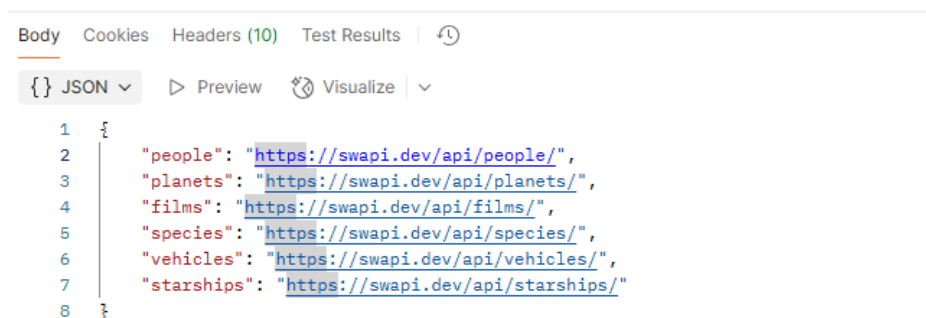
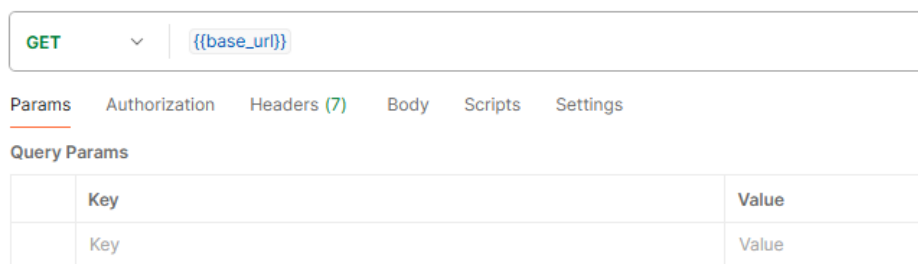
Цель работы: Получение знаний об особенностях тестирования REST API, а также базовых умений и навыков использования инструмента Postman.

Ход выполнения работы:

1.Создали коллекцию.



2.Создали запрос.



3. Запросили информацию о планетах. Выясните, что означают атрибуты `rotation_period` – период вращения планеты во круг своей оси в часах и `orbital_period` – период обращения планеты во круг звезды в сутках.

The screenshot shows the 'kspace' application interface. The URL bar displays `https://swapi.dev/api/planets/`. The method is set to `GET`. The response is shown in JSON format, with line numbers 1 through 35. The JSON structure is as follows:

```
1 {
2   "count": 60,
3   "next": "https://swapi.dev/api/planets/?page=2",
4   "previous": null,
5   "results": [
6     {
7       "name": "Tatooine",
8       "rotation_period": "23",
9       "orbital_period": "304",
10      "diameter": "10465",
11      "climate": "arid",
12      "gravity": "1 standard",
13      "terrain": "desert",
14      "surface_water": "1",
15      "population": "200000",
16      "residents": [
17        "https://swapi.dev/api/people/1/",
18        "https://swapi.dev/api/people/2/",
19        "https://swapi.dev/api/people/4/",
20        "https://swapi.dev/api/people/6/",
21        "https://swapi.dev/api/people/7/",
22        "https://swapi.dev/api/people/8/",
23        "https://swapi.dev/api/people/9/",
24        "https://swapi.dev/api/people/11/",
25        "https://swapi.dev/api/people/43/",
26        "https://swapi.dev/api/people/62/"
27      ],
28      "films": [
29        "https://swapi.dev/api/films/1/",
30        "https://swapi.dev/api/films/3/",
31        "https://swapi.dev/api/films/4/",
32        "https://swapi.dev/api/films/5/",
33        "https://swapi.dev/api/films/6/"
34      ],
35      "created": "2014-12-09T13:50:49.641000Z",
```

4. Запросили подробную информацию обо всех звездолетах

Leia Organa

GET New Request

StarWarsCollection

GET New Request

Leia Organa / New Request

GET {{base_url}} starships/

Params Authorization Headers (7) Body Scripts Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (10) Test Results

{} JSON

Preview Visualize

```
1 {
2   "count": 36,
3   "next": "https://swapi.dev/api/starships/?page=2",
4   "previous": null,
5   "results": [
6     {
7       "name": "CR90 corvette",
8       "model": "CR90 corvette",
9       "manufacturer": "Corellian Engineering Corporation",
10      "cost_in_credits": "3500000",
11      "length": "150",
12      "max_atmosphering_speed": "950",
13      "crew": "30-165",
14      "passengers": "600",
15      "cargo_capacity": "3000000",
16      "consumables": "1 year"
```

5. Теперь выполнили тот же запрос, но с параметром, позволяющим получить вторую страницу ответа от сервера. Проверьте с помощью теста, что запрос выполнен успешно

HTTP

Leia Organa / New Request

GET

▼

{{base_url}} starships/?page=2

Params

Authorization

Headers (7)

Body

Scripts

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	page	2
	Key	Value

Body

Cookies

Headers (10)

Test Results

🕒

{}

JSON

▼

▶ Preview

🔗 Visualize

▼

1

{

2

"count": 36,

3

"next": "https://swapi.dev/api/starships/?page=3",

4

"previous": "https://swapi.dev/api/starships/?page=1",

5

"results": [

6

{

7

"name": "Slave 1",

8

"model": "Firespray-31-class patrol and attack",

9

"manufacturer": "Kuat Systems Engineering",

10

"cost_in_credits": "unknown",

11

"length": "21.5",

12

"max_atmosphering_speed": "1000",

13

"crew": "1",

14

"passengers": "6",

15

"cargo_capacity": "70000",

16

"consumables": "1 month",

17

"hyperdrive_rating": "3.0",

18

"MGLT": "70",

19

"starship_class": "Patrol craft",

20

"pilots": [

21

"https://swapi.dev/api/people/22/"

22

],

23

"films": [

24

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

25

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

26

],

27

"created": "2014-12-15T13:00:56.332000Z",

28

"edited": "2014-12-20T21:23:10.807000Z"

6. Запросили все звездолеты, в названии или модели которых есть подстрока Star или star. Проверьте с помощью теста, пришел ли ответ за 300 мс.(не приходит запрос за 300 мс)

Leia Organa / New Request

GET {{base_url}} starships/?search=star

Params • Authorization Headers (7) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	search	star
	Key	Value

Body Cookies Headers (10) Test Results ↻

{ } JSON ▾ ▶ Preview 📄 Visualize ▾

```
1 {
2   "count": 15,
3   "next": "https://swapi.dev/api/starships/?search=star&page=2",
4   "previous": null,
5   "results": [
6     {
7       "name": "Star Destroyer",
8       "model": "Imperial I-class Star Destroyer",
9       "manufacturer": "Kuat Drive Yards",
10      "cost_in_credits": "150000000",
11      "length": "1,600",
12      "max_atmosphering_speed": "975",
13      "crew": "47,060",
14      "passengers": "n/a",
15      "cargo_capacity": "36000000",
16      "consumables": "2 years",
17      "hyperdrive_rating": "2.0",
18      "MGLT": "60",
19      "starship_class": "Star Destroyer",
20      "pilots": [],
21      "films": [
22        "https://swapi.dev/api/films/1/",
23        "https://swapi.dev/api/films/2/",
24        "https://swapi.dev/api/films/3/"
25      ],
26      "created": "2014-12-10T15:08:19.848000Z",
```

7. Запросили подробную информацию обо всех звездолетах на языке wookiee.

Leia Organa / New Request

GET

{{base_url}} starships/?format=wookiee

Params

Authorization

Headers (7)

Body

Scripts

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	format	wookiee
	Key	Value

Body

Cookies

Headers (10)

Test Results

🔄

{ } JSON

▶ Preview

🔗 Visualize

▼

1

{

2

"oaoohuwhao": 36,

3

"whwokao": "acaooakc://cohraakah.wawoho/raakah/caorarcacakahak/?akrarrwo=2&wwoorcsraao=ohooooorahwowa",

4

"akrcwohoahoohuc": whhuanan,

5

"rcwochuanaoc": [

6

{

7

"whrascwo": "CR90 oaoorchowaoaowo",

8

"scoowawoan": "CR90 oaoorchowaoaowo",

9

"scrawhhuwraaoahurcworc": "Coorcwoananahrawh Ewhirrahwhwoworcahwhrr Coorcakoorcraaoahoowh",

10

"oaoocao_ahwh_oarcwowaahaoc": "3500000",

11

"anwowhrraoac": "150",

12

"scrak_raaoscoocakacworcahwhrr_cakwowowa": "950",

13

"oarcwooh": "30-165",

14

"akraccworrhrrworcc": "600",

15

"oararcrrroo_oaraakraoahaoro": "3000000",

16

"oaoowhchuscrahanwoc": "1 roworarc",

17

"acroakworcwarcahhowo_rcraaoahwhrr": "2.0",

18

"MGLT": "60",

19

"caorarcacakahak_oaanracc": "oaoorchowaoaowo",

20

"akahanoaoac": [],

21

"wwahanscc": [

22

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

23

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

24

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

25

],

26

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

27

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

8. Проверили с помощью теста, что ответ от сервера содержит подстроку warcworawawhoohurracao.

GET {{base_url}} starships/?format=wookiee

Params • Authorization Headers (7) Body Scripts • Settings

Pre-request

Post-response •

```
1 pm.test("Ответ содержит warcworawawhoohurracao", function(){
2   | pm.expect(pm.response.text()).to.include("warcworawawhoohurracao")
3   | });
```

Body Cookies Headers (10) Test Results (1/1) | ↻

Testing your API endpoints one by one? Run your entire collection and test end-to-end workflows effortlessly. [Learn more](#)

Filter Results ▾

PASSED Ответ содержит warcworawawhoohurracao

GET https://swapi.dev/api/people/

Params Authorization Headers (7) Body Scripts • Settings

Pre-request

Post-response •

```
1 pm.test("Status code is 200", function () {
2   | pm.response.to.have.status(200);
3   | });
4
```

Body Cookies Headers (10) Test Results (1/1) | ↻

Testing your API endpoints one by one? Run your entire collection and test end-to-end workflows effortlessly. [Learn more](#)

Filter Results ▾

PASSED Status code is 200

Leia Organa / New Request

GET `{{base_url}} starships/?format=wookiee`

Params • Authorization Headers (7) Body Scripts • Settings

Pre-request

Post-response

```

1 pm.test("Ответ содержит warcworawawhoohurracao", function () {
2   pm.expect(pm.response.text()).to.include("warcworawawhoohurracao")
3 });
4 pm.test("Status code is 300", function () {
5   pm.response.to.have.status(300);
6 });
7 pm.test("Запрос выполнен успешно", function () {
8   pm.response.to.have.status(200);
9 });

```

Body Cookies Headers (10) Test Results (2/3) | 🔄

Filter Results ▾

PASSED Ответ содержит warcworawawhoohurracao

FAILED Status code is 300 | AssertionError: expected response to have status code 300 but got 200

PASSED Запрос выполнен успешно

9. Запустили вместе все три последних запроса с тестами.

Leia Organa - Run results

👤 Ran today at 05:46:20 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	2s 465ms	3	2085 ms

All Tests Passed (2) Failed (1) Skipped (0)

Iteration 1

GET New Request
https://swapi.dev/api/starships/?format=wookiee

- PASS Ответ содержит warcworawawhoohurracao
- FAIL Status code is 300 | AssertionError: expected response to have status code 300 but got 200
- PASS Запрос выполнен успешно

Ответы на контрольные вопросы:

1) Платформа Postman предназначена для разработки, тестирования и документирования API. Она позволяет отправлять запросы к API и анализировать ответы.

2) REST (Representational State Transfer) представляет собой архитектурный стиль, который используется для разработки веб-сервисов. Он основывается на принципах работы с ресурсами через стандартные HTTP-методы.

3) Наиболее часто используемые HTTP-методы в REST-запросах: GET, POST, PUT, DELETE.

4) Отличительная особенность метода GET заключается в том, что он используется для получения данных с сервера. Запросы GET не должны изменять состояние ресурса на сервере.

5) API (Application Programming Interface) — это интерфейс, который позволяет одной программе взаимодействовать с другой. Он предоставляет набор правил и протоколов для работы с различными программными компонентами.

6) Коллекции в Postman объединяют в себе связанные запросы, которые можно организовать в структуру, позволяя эффективно управлять тестами и документами.

7) Переменную коллекции в Postman можно использовать для хранения значений, которые могут быть доступны всем запросам в данной коллекции. Это позволяет динамически изменять параметры запросов.

8) Запрос GET с параметрами можно создать, добавив параметры в строку URL, используя параметры запроса в Postman или добавив их в разделе "Params".

9) Снимпеты нужны для упрощения работы с кодом. Они представляют собой заранее определенные фрагменты кода, которые можно быстро вставить в запросы или тесты.

10) Для запуска нескольких запросов вместе с тестами используется функционал коллекций и возможность их запуска через "Collection Runner" в Postman.

11) Тестировщику для начала тестирования API необходима информация о конечных точках API, HTTP-методах, формате запроса и ответа, параметрах, а также об спецификациях и ограничениях API.

12) Разница между методами POST и PUT состоит в том, что POST используется для создания нового ресурса, тогда как PUT используется для обновления существующего ресурса или создания его, если он не существует