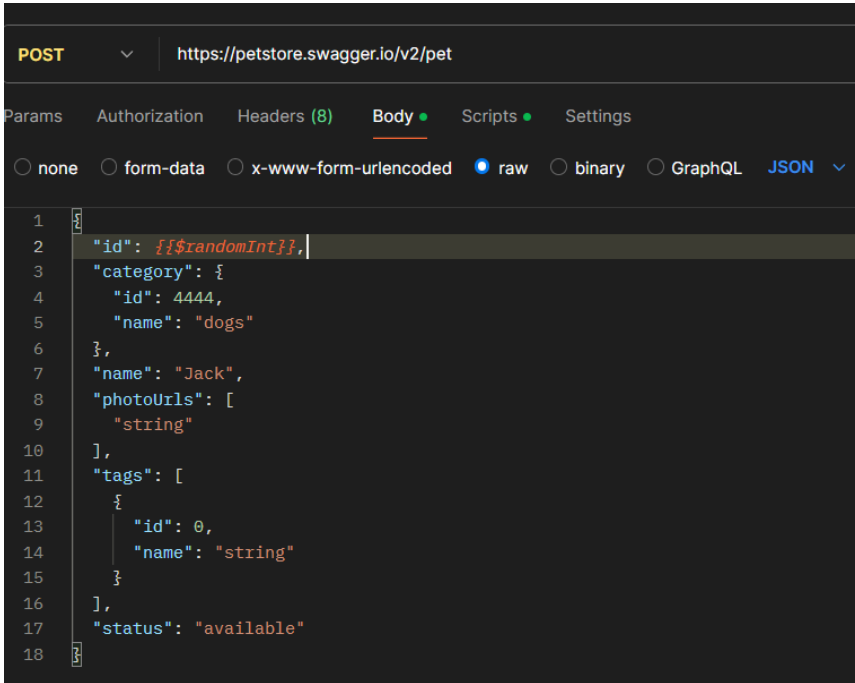
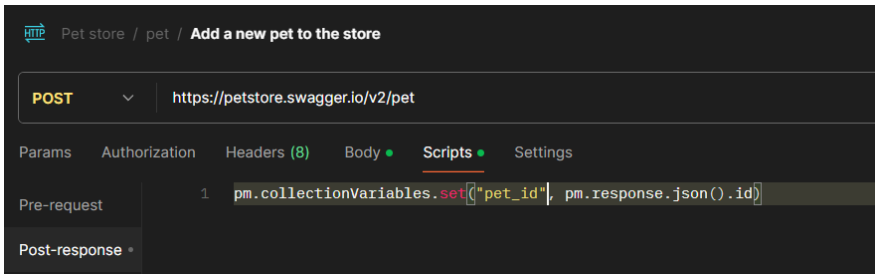


Тема лабораторной работы	lb_7
Выполняющий	Холоднов Вадим
Помогающий	
Ход работы	<p>1. В запросе Add a new pet to the store меняю id питомца на случайно сгенерированное.</p>  <pre> 1 { 2 "id": {{\$randomInt}}, 3 "category": { 4 "id": 4444, 5 "name": "dogs" 6 }, 7 "name": "Jack", 8 "photoUrls": [9 "string" 10], 11 "tags": [12 { 13 "id": 0, 14 "name": "string" 15 } 16], 17 "status": "available" 18 } </pre> <p>2. Там же в Post-response пишу скрипт, который определит id питомца и запишет его в переменную pet_id.</p>  <pre> 1 pm.collectionVariables.set("pet_id", pm.response.json().id) </pre> <p>3. Отправляю запрос. К питомцу сгенерировалось id, и автоматически создалась переменная pet_id.</p>

```

1  {
2      "id": 490,
3      "category": {
4          "id": 4444,
5          "name": "dogs"
6      },
7      "name": "Jack",
8      "photoUrls": [
9          "string"
10     ],
11     "tags": [
12         {
13             "id": 0,
14             "name": "string"
15         }
16     ],
17     "status": "available"
18 }

```

Pet store

Overview Authorization Scripts **Variables** Runs

These variables are specific to this collection and its requests. [Learn more about collection variables](#)

Filter variables

	Variable	Initial value	Current value
+ <input checked="" type="checkbox"/>	pet_id		490
Add new variable			

4. Для того же запроса вставил 3 готовых снippets, которые проверят завершён ли запрос успешно, не превысило ли ожидание ответа 2 секунды и проверяет заголовок.

Pet store / pet / Add a new pet to the store

POST https://petstore.swagger.io/v2/pet Send

Params Authorization Headers (8) Body **Scripts** Settings Cookies

Pre-request

```

1 pm.test("Response time is less than 2000ms", function () {
2     pm.expect(pm.response.responseTime).to.be.below(2000);
3 });

```

Post-response

```

4 pm.collectionVariables.set("pet_id", pm.response.json().id);
5
6
7 pm.test("Content-Type is present", function () {
8     pm.response.to.have.header("Content-Type");
9 });
10
11 pm.test("Status code is 200", function () {
12     pm.response.to.have.status(200);
13 });

```

- Set a collection variable
- Clear an environment variable
- Clear a global variable
- Clear a collection variable
- Send a request
- Status code: Code is 200
- Response body: Contains string
- Response body: JSON value check
- Response body: Is equal to a string
- Response headers: Content-Type header check
- Response time is less than 200ms
- Status code: Successful POST request
- Status code: Code name has string
- Response body: Convert XML body to a

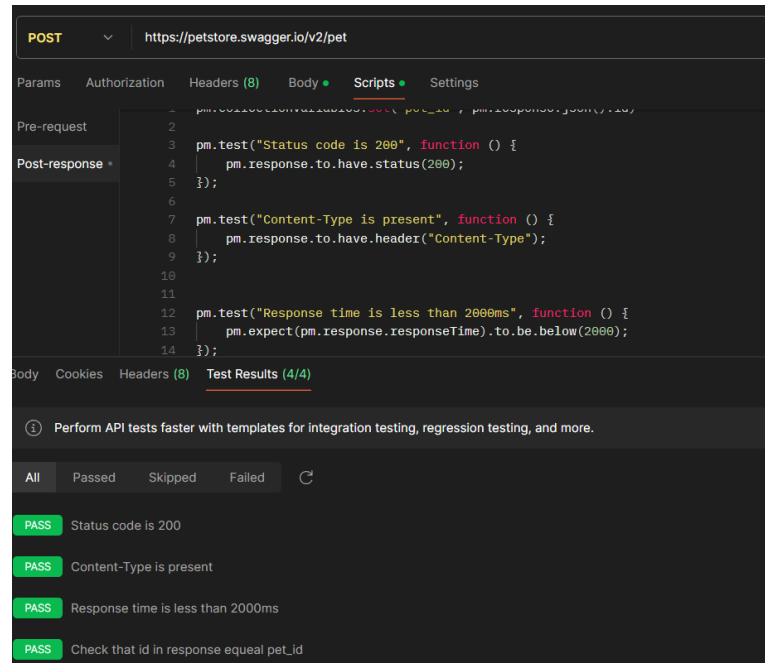
5. Используя снippet Response body: JSON value check, написал тест, который проверяет id.

```

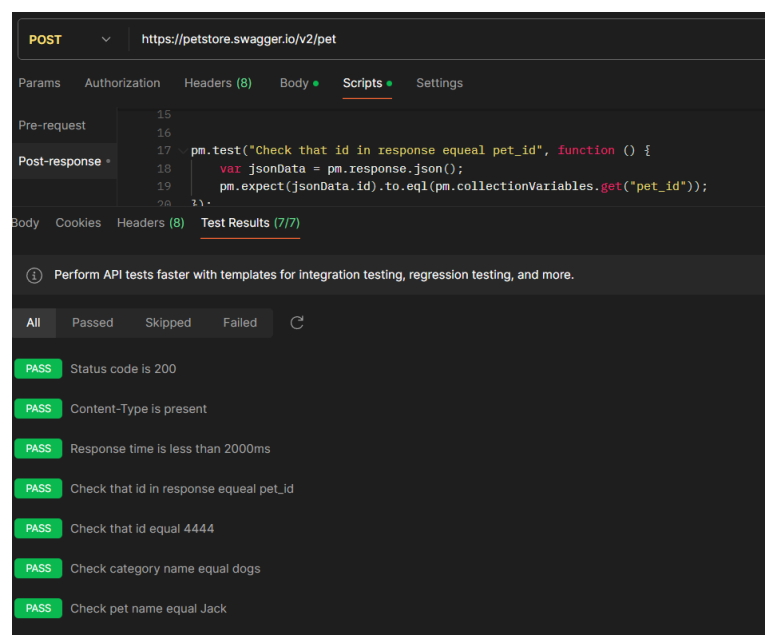
32 pm.test("Check pet name equal Jack", function () {
33     var jsonData = pm.response.json();
34     pm.expect(jsonData.name).to.eql("Jack");
35 });

```

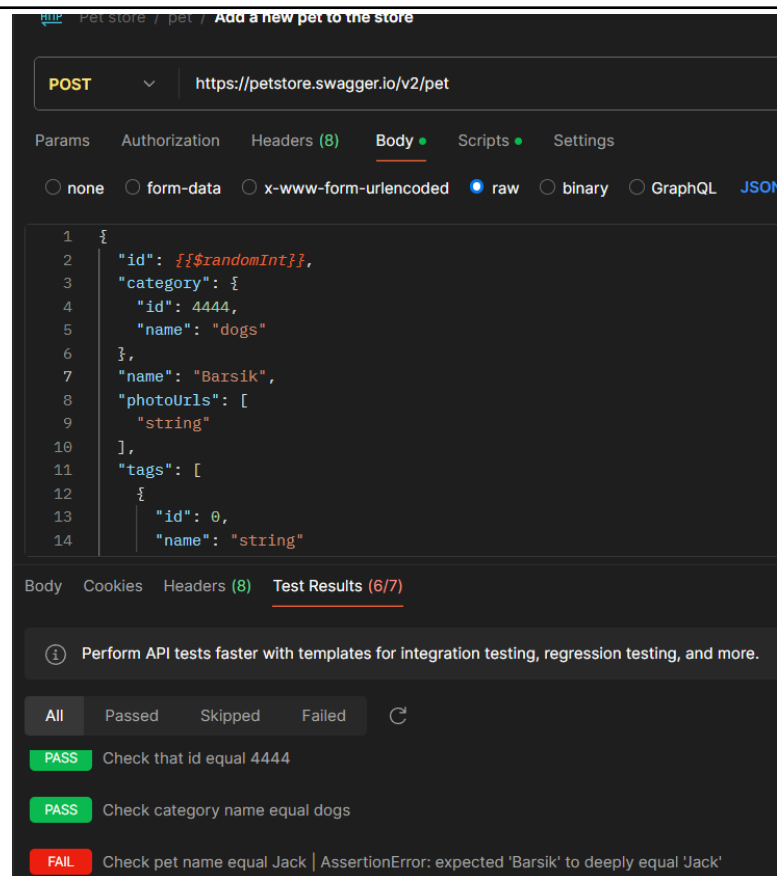
6. Проверка прошла успешно.



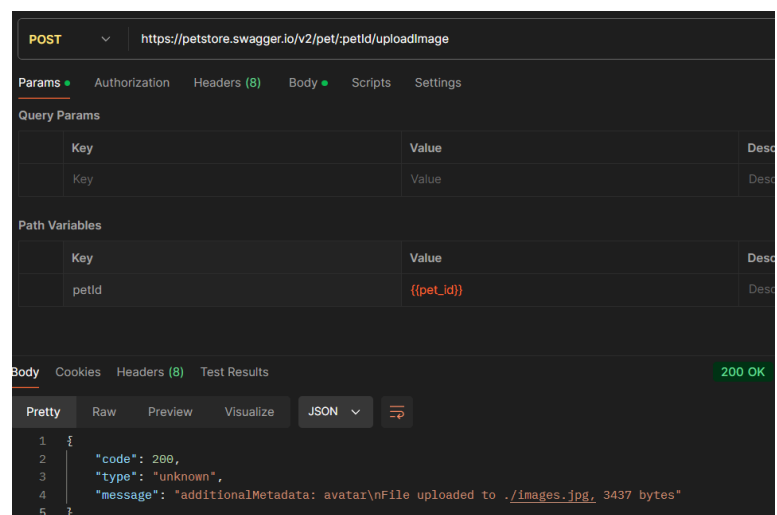
7. По такому же принципу написал тесты, которые проверяют id категории 4444, имя на Jack, категорию на Dogs.



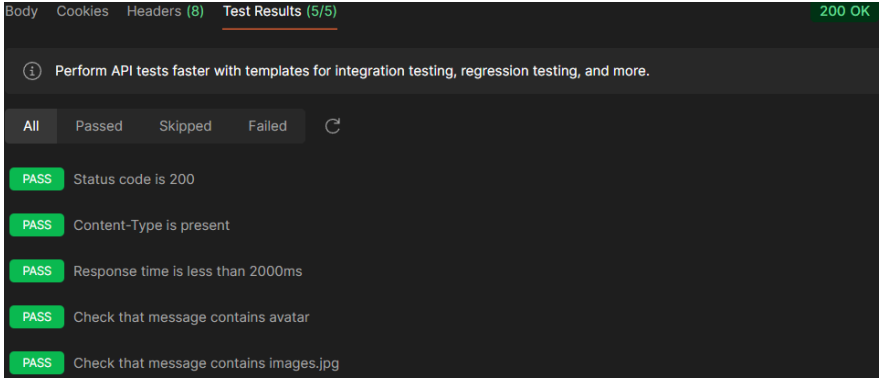
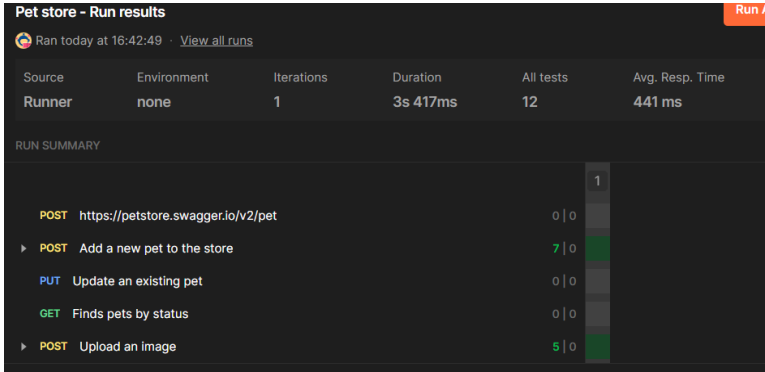
8. Поменял имя питомца на Barsik и один тест упал.



9. В запросе UploadImage поменял id животного на :petId. Автоматически создался параметр.



10. Также для UploadImage написал 5 тестов. 3 первых теста уже были в предыдущем запросе. Четвертый проверяет

	<pre>1 pm.test("Status code is 200", function () { 2 pm.response.to.have.status(200); 3 }); 4 5 pm.test("Content-Type is present", function () { 6 pm.response.to.have.header("Content-Type"); 7 }); 8 9 10 pm.test("Response time is less than 2000ms", function () { 11 pm.expect(pm.response.responseTime).to.be.below(2000); 12 }); 13 14 pm.test("Check that message contains avatar", function () { 15 var jsonData = pm.response.json(); 16 pm.expect(jsonData.message).to.contains("avatar"); 17 }); 18 19 pm.test("Check that message contains images.jpg", function () { 20 var jsonData = pm.response.json(); 21 pm.expect(jsonData.message).to.contains("images.jpg"); 22 });</pre> <p>11.Проверка прошла успешно.</p>  <p>12. Под конец запустил проверку всей сборки.</p> 
Результат	В ходе лабораторной работы я научился с помощью готовых снипетов создавать автоматические тесты для запросов.
Оценка	

