

| | |
|--------------------------|--|
| Тема лабораторной работы | lb_7 |
| Выполняющий | Трехлеб Марина |
| Помогающий | |
| Ход работы | <div><div>1) В первом запросе, в котором я создала животное заменила id на <code>{{ \$randomInt }}</code>:</div><div><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></div><div><div>2) Написала скрипт, отправила запрос, появилась переменная и в ней значение:</div><div><pre>pm.collectionVariables.set("pet_id", pm.response.json().id)</pre><div><div><input checked="" type="checkbox"/></div><div>pet_id</div><div>177</div></div></div></div><div><div>3) Использовала сниппеты</div><div><ul style="list-style-type: none">• Status code: Code is 200• Response headers: Content-Type header check• Response time is less than 200ms (у себя заменила 200 на 2000)• Response body: JSON value check (вставила своё значение)</div></div></div> |

Отправила запрос и получила ответ:

The screenshot shows the Postman interface with a REST client request. The 'Test' tab is active, displaying the following JavaScript code for the tests:

```
1 pm.collectionVariables.set("pet_id", pm.response.json().id)
2
3 pm.test("Status code is 200", function () {
4   pm.response.to.have.status(200);
5 });
6
7 pm.test("Content-Type is present", function () {
8   pm.response.to.have.header("Content-Type");
9 });
10
11 pm.test("Response time is less than 2000ms", function () {
12   pm.expect(pm.response.responseTime).to.be.below(2000);
13 });
14
15 pm.test("Check that id in response equal pet_id", function () {
16   var jsonData = pm.response.json();
17   pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("pet_id"));
18 });
```

Below the code, the 'Test Results (4/4)' section shows four tests, all of which passed:

- PASS** Status code is 200
- PASS** Content-Type is present
- PASS** Response time is less than 2000ms
- PASS** Check that id in response equal pet_id

4) Дописала ещё три теста:

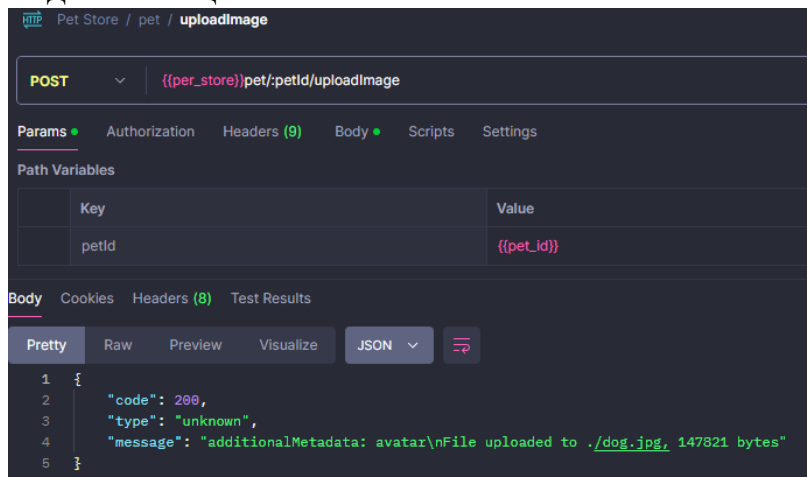
The screenshot shows the Postman interface with a REST client request. The 'Test' tab is active, displaying the following JavaScript code for the tests:

```
18 });
19
20 pm.test("Check that id equal 4444", function () {
21   var jsonData = pm.response.json();
22   pm.expect(jsonData.category.id).to.eql(4444);
23 });
24
25 pm.test("Check category name equal dogs", function () {
26   var jsonData = pm.response.json();
27   pm.expect(jsonData.category.name).to.eql("dogs");
28 });
29
30 pm.test("Check pet name equal Jack", function () {
31   var jsonData = pm.response.json();
32   pm.expect(jsonData.name).to.eql("Jack");
33 });
```

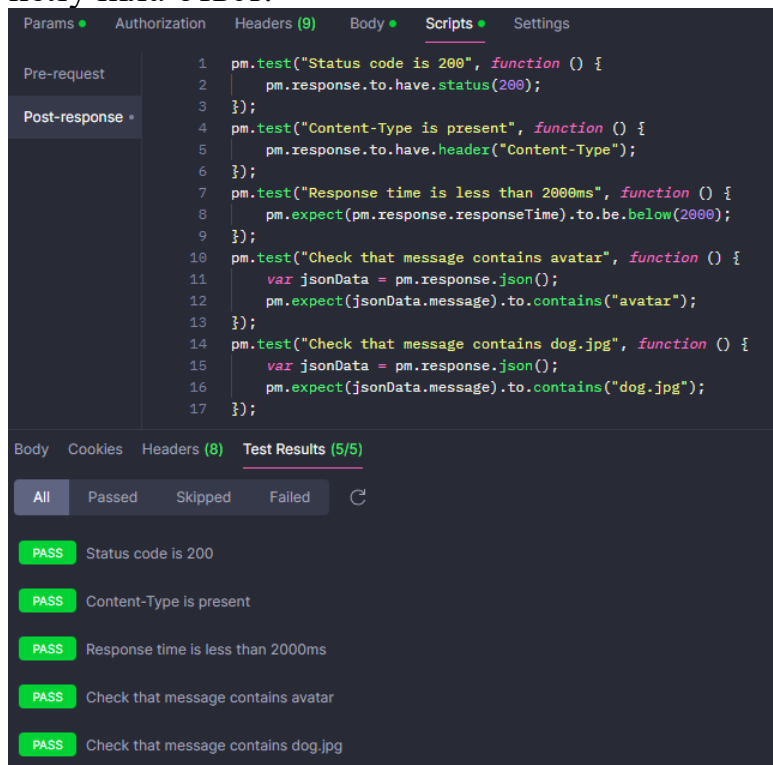
Below the code, the 'Test Results (7/7)' section shows seven tests, all of which passed:

- PASS** Status code is 200
- PASS** Content-Type is present
- PASS** Response time is less than 2000ms
- PASS** Check that id in response equal pet_id
- PASS** Check that id equal 4444
- PASS** Check category name equal dogs
- PASS** Check pet name equal Jack

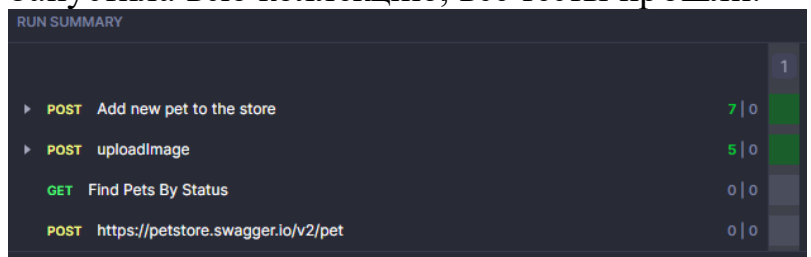
5) В другом методе создала параметр для задания айди питомца:



6) Используя сниппеты, написала скрипты, получила ответ:



7) Запустила всю коллекцию, все тесты прошли:



Результат

Я научилась писать скрипты для теста используя code snippet

Критерий оценивания

| | |
|---------------------|---|
| Отлично | Четко, лаконично, по факту, гитхаб на ура |
| Хорошо | Небольшие корректировки |
| Удовлетворительно | С вопросами, и комментариями |
| Неудовлетворительно | Ничего непонятно |