Тема лабораторной работы	lb_7
Выполняющий	Кирюшкин Артем
Помогающий	Солунин Евгений
Ход работы	1) Для начала нам нужно поменять параметр id у животного

```
1  {
2    "id": {{$randomInt}},
3    "category": {
4      "id": 444,
5      "name": "string"
6    },
7    "name": "bigl",
8    "photoUrls": [
9      "string"
10   ],
11   "tags": [
12     {
13       "id": 0,
14       "name": "string"
15     }
16   ],
17   "status": "available"
```

2) В вкладке test нужно вписать новую команду

Params | Authorization | Headers (8) | Body • | Scripts • | Settings

Pre-request •

Post-response •

```
1  pm.collectionVariables.set("pet_id", pm.response.json().id)
```

3) У нас сгенерировалось новое айди

Filter variables

	Variable	Initial value	Current value
☑	per_store	https://petstore.swagger.io/v2	https://petstore.swagger.io/v2
☑	pet_id		195
	Add new variable		

4) Далее нам нужно зайти в вкладку test и начать разбираться с снипетами
5) Добавим нужные нам снипеты и внесём в них

собственные изменения(Листинг №1)

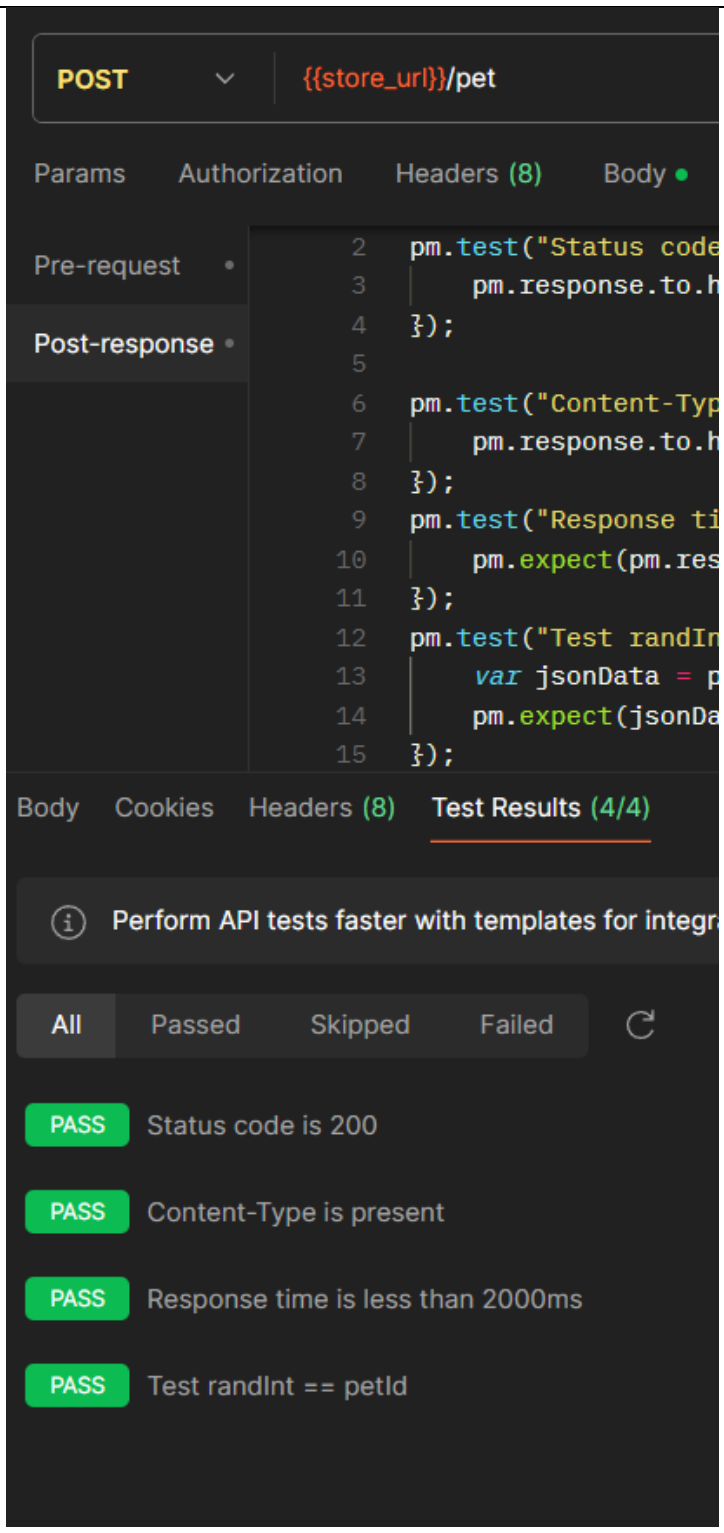
Pre-request

Post-response

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

pm.test("Content-Type is present", function () {
    pm.response.to.have.header("Content-Type");
});
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(4000);
});
pm.test("Test randInt == petId", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("pet_id"));
});
```

6) Теперь проверим работоспособность наших тестов и результаты в них



7) Теперь вернёмся в body поменяем id и имя

```
1  {
2    "id": {{$randomInt}},
3    "category": {
4      "id": 4444,
5      "name": "dogs"
6    },
7    "name": "lord",
8    "photoUrls": [
9      "dogs"
10   ],
11   "tags": [
```

8) И добавим новые снипеты (листинг №2)

```
pm.test("Test category.id == 4444", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.id).to.eql(4444);
});
pm.test("Test category.name == dogs", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.name).to.eql("dogs");
});
pm.test("Test name == lord", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.name).to.eql("lord");
});
```

9) Проверяем работоспособность

Params | Authorization | Headers (8) | Body | Scripts | Settings

Pre-request | Post-response

```
15     pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("pet_id"));
16 });
17 pm.test("Test category.id == 4444", function () {
18     var jsonData = pm.response.json();
19     pm.expect(jsonData.category.id).to.eql(4444);
20 });
21 pm.test("Test category.name == dogs", function () {
22     var jsonData = pm.response.json();
23     pm.expect(jsonData.category.name).to.eql("dogs");
24 });
25 pm.test("Test name == lord", function () {
26     var jsonData = pm.response.json();
27     pm.expect(jsonData.name).to.eql("lord");
28 });
```

Body | Cookies | Headers (8) | Test Results (7/7) | 200 OK

Perform API tests faster with templates for integration testing, regression testing, and more.

All | Passed | Skipped | Failed

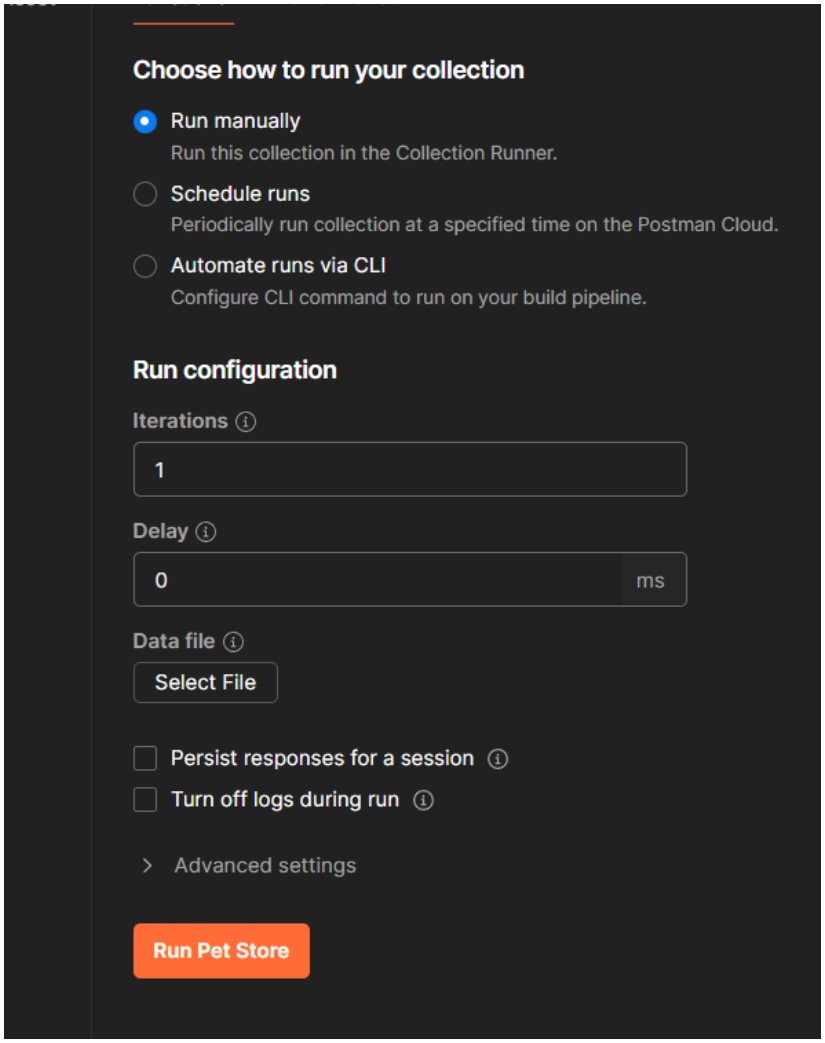
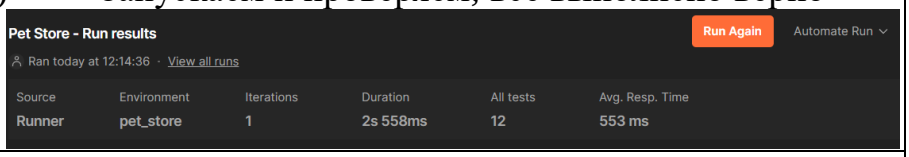
PASS Response time is less than 2000ms

PASS Test randInt == petId

PASS Test category.id == 4444

PASS Test category.name == dogs

PASS Test name == lord

	10)Подобным же образом добавляем последующие (листинг№3) 11)нужные нам тесты и проверяем их все с помощью run collection  10) Запускаем и проверяем, всё выполнено верно 
Результат	Мы углубились в postman, научились полноценно работать с тестированием и создавать собственные запросы на тестирование.

Листинг№1

```
pm.collectionVariables.set("pet_id", pm.response.json().id)
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Content-Type is present", function () {
```

```
    pm.response.to.have.header("Content-Type");
});
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(4000);
});
pm.test("Test randInt == petId", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("pet_id"));
});
```

Листинг№2

```
pm.collectionVariables.set("pet_id", pm.response.json().id)
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Content-Type is present", function () {
    pm.response.to.have.header("Content-Type");
});
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});

pm.test("Test randInt == petId", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("pet_id"));
});
pm.test("Test category.id == 4444", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.id).to.eql(4444);
});
pm.test("Test category.name == dogs", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.name).to.eql("dogs");
});
pm.test("Test name == lord", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.name).to.eql("lord");
});
```

Листинг№3

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

pm.test("Content-Type is present", function () {
```

```
    pm.response.to.have.header("Content-Type");
});
pm.test("Response time is less than 4000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(4000);
});

pm.test("check that message contains avatar", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.message).to.contain("avatar");
});
pm.test("check that message contains logo.jpg", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.message).to.contain("logo.jpg");
```