

Тема лабораторной работы	lb_7
Выполнил	Ржевский Глеб
Помогал	
Ход работы	<p>1) Вместо id моего питомца пишу строчку, чтобы каждый раз было случайное число</p> <pre> 1  { 2    "id": {{\$randomInt}}, 3    "category": { 4      "id": 777, 5      "name": "cats" 6    }, 7    "name": "Barsik", 8    "photoUrls": [ 9      "string" 10   ], 11   "tags": [ </pre> <p>2) Во вкладке Test вставляю эту строку, чтобы она устанавливала значение переменной pet_id равным значению id из json</p> <pre> 1  pm.collectionVariables.set("pet_id", pm.response.json().id) </pre> <p>3) Видим, что числа выводятся случайно</p> <pre> 1  { 2    "id": 757, 3    "category": { 4      "id": 777, 5      "name": "cats" 6    }, 7    "name": "Barsik", 8    "photoUrls": [ 9      "string" 10   ], </pre> <p>4) Выбрал в Snippets Status code: Code is 200 и добавилась эта строка</p> <pre> 1  pm.collectionVariables.set("pet_id", pm.response.json().id) 2  pm.test("Status code is 200", function () { 3    pm.response.to.have.status(200); 4  }); 5 </pre> <p>Она проверяет вернул ли сервер статус 200</p> <p>5) Далее я добавил ещё snippets, которые проверяют есть ли в ответе сервера заголовок Content type, ответил ли сервер менее чем за 2 секунды и равен ли id переменной pet_id</p>

```

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("Your test name", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.value).to.eql(100);
});

```

6) Изменил последний snippet

```

pm.test("Check that id in response equals pet_id", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.value).to.eql(pm.collectionVariables.get(
        "pet_id"));
});

```

7) Проверяем

```

PASS Status code is 200

PASS Content-Type is present

PASS Response time is less than 2000ms

PASS Check that id in response equals pet_id

```

8) Создаю ещё 3 одинаковых snippetа и пишу в них данные моего питомца

```

pm.test("Check that id equals 777", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.id).to.eql(777);
});

pm.test("Check that category name equals cats", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.name).to.eql("cats");
});

pm.test("Check that name equals Barsik", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.name).to.eql("Barsik");
});

```

9) Нажал Send. Получил результат

- PASS** Status code is 200
- PASS** Content-Type is present
- PASS** Response time is less than 2000ms
- PASS** Check that id in response equals pet\_id
- PASS** Check that id equals 777
- PASS** Check that category name equals cats
- PASS** Check that name equals Barsik

10) Вместо id пишу :petId, а в значение ключа пишу {{pet\_id}}

The screenshot shows the Postman interface for a POST request. The URL is `{{pet_store}}/pet/:petId/uploadImage`. The request body is set to `form-data` with two fields: `Avatar` (Text) and `file` (File, pointing to `logo.png`). The response status is `200 OK`. The response body is shown in JSON format:

```

1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "additionalMetadata: null\nFile uploaded to ./logo.png, 1786694 bytes"
5 }

```

11) Добавляю сниппеты для запроса и запускаю

```

pm.test("Check that message contains avatar", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.message).to.contains("avatar");
});

pm.test("Check that message contains logo.png", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.message).to.contains("logo.png");
});

```

12) Запускаю и видим, что всё работает(кроме время выполнения запроса, плохой инет)

```
PASS Status code is 200

PASS Content-Type is present

FAIL Response time is less than 2000ms | AssertionError: expected 3184 to be below 2000

PASS Check that message contains avatar

PASS Check that message contains logo.png
```

13) Запускаю всю коллекцию и вижу результат(все прошло кроме выполнения запроса, плохой инет)

		1
POST	Add a new pet to the store	7   0
POST	upload an Image	4   1
Pass Status code is 200		
Pass Content-Type is present		
Fail Response time is less than 2000ms		
Pass Check that message contains avatar		
Pass Check that message contains logo.png		
POST	https://petstore.swagger.io/v2/pet	0   0
GET	Find Pets by status	0   0
GET	https://petstore.swagger.io/v2/pet/666	0   0
POST	UPDATE PET	0   0
PUT	Need add to store	0   0

Результат

Я научился работать со Snippets, проверять с помощью тестов запросы.