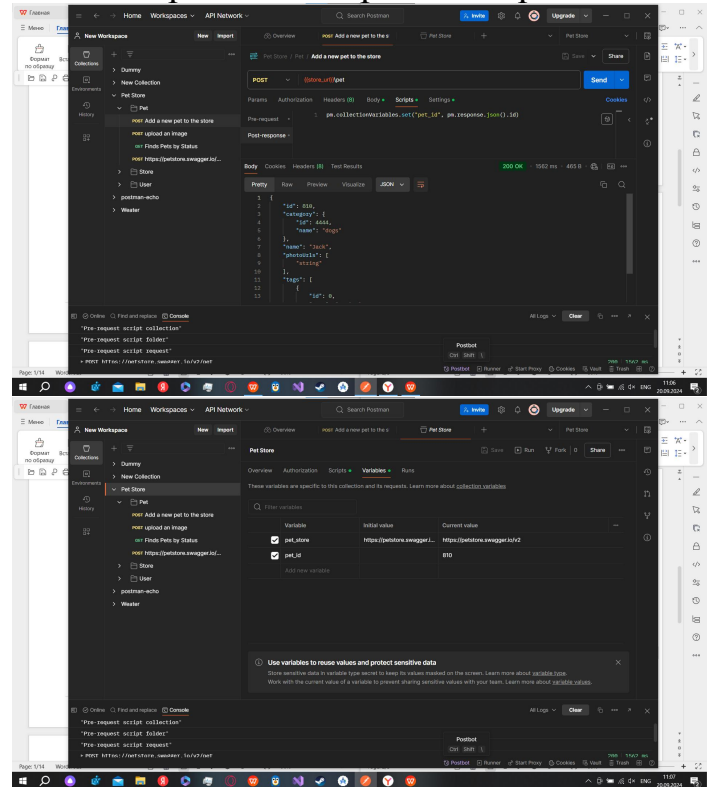


Тема лабораторной работы	lb_7
Выполняющий	Кутенков Константин
Помогающий	Наглов Виктор
Ход работы	<ol style="list-style-type: none"> 1. После ознакомления с лабораторной работой, мы приступили к ее выполнению 2. Заходим в ранее созданную коллекцию Pet Sotre и меняем id на генератор случайного числа в запросе, который добавляет новых животных <div data-bbox="710 580 1396 958" data-label="Image"> </div> 3. Далее на вкладке Tests добавляем строку: <div data-bbox="801 1088 1374 1176" data-label="Text"> <pre>pm.collectionVariables.set(«pet_id», pm.response.json().id)</pre> </div> <div data-bbox="710 1169 1374 1541" data-label="Image"> </div> 4. Также убедимся, что наша переменная pet_store единственная <div data-bbox="710 1668 1361 2031" data-label="Image"> </div>

5. Сохраним и отправим запрос



И у нас сгенерировался id 810

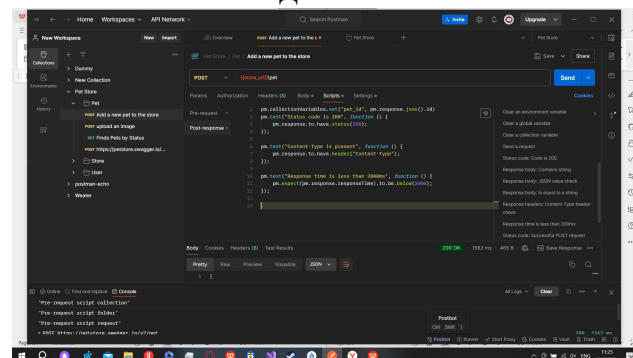
6. Теперь попробуем сделать автотест, пользуясь готовым сниппетам.

Первый код проверяет возвращаемый код, и если он не 200, то тест провален.

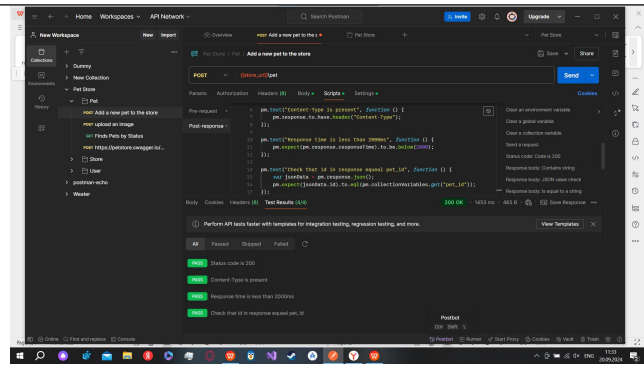
Второй код проверяет наличие заголовка “Content-Type”, если его нет, то тест провален

Третий код проверяет время ответа сервера, если оно превысит 2 секунды, то тест провален

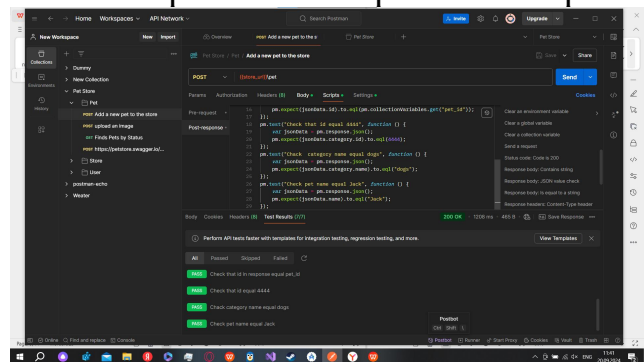
Вот итоговый код



7. Добавим еще сниппет для сравнения pet_id, отредактируем код и отправляем

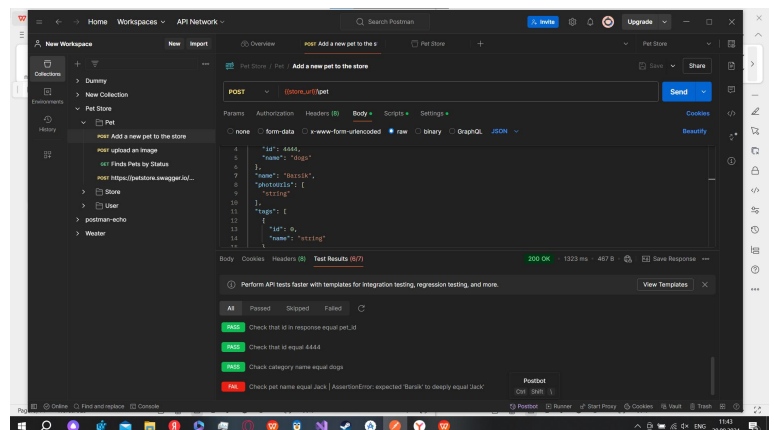


8. Аналогично добавим проверку для названия категории и имени питомца. Сохраняем и отправляем запрос.



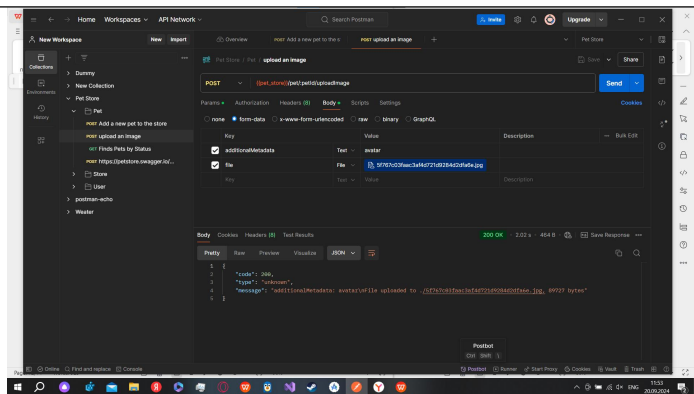
Как видим все тесты успешно выполнились

9. Теперь попробуем намеренно провалить тест. Поменяем имя Jack на Barsik.



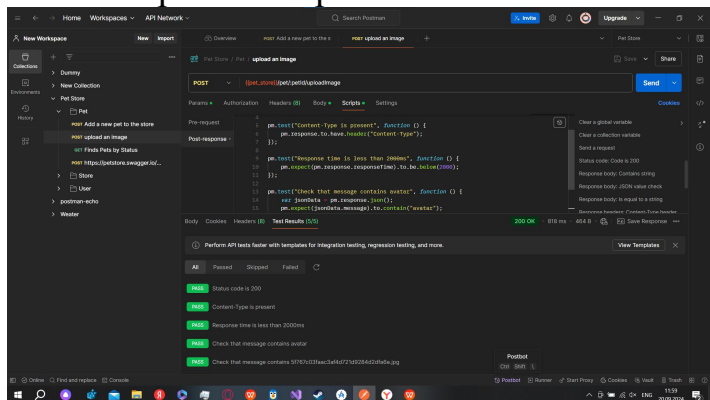
Тест провален.

10. Следуя инструкции, редактируем запрос upload an image, заменив вручную записанный id на :petId и он отобразился в Path Variables. Ему в качестве значения передаем переменную pet_id, сохраним и отправим запрос.



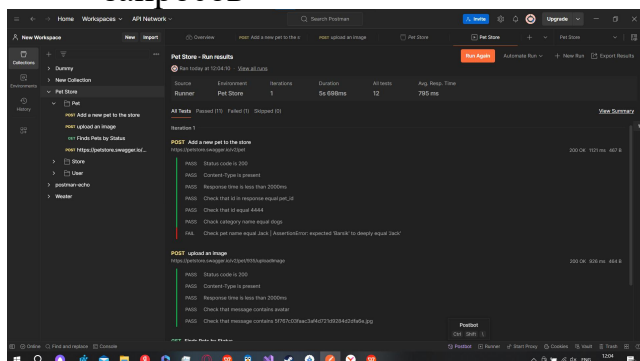
Все прекрасно отрабатывает

11. Добавим автотесты и для этого запроса. Проверим возвращаемый код, наличие заголовка, время ответа сервера, наличие аватара, наличие изображения. И теперь отправим запрос



И все пять тестов были пройдены успешно

12. И напоследок, запустим всю коллекцию запросов



И все тесты выполнились успешно, кроме проверки имени Jack, которого мы заменили на Barsik

Результат

После проделанной нами работы, мы:
1. Используя знания, полученные ранее, отредактировали запросы Pet Store, используя

	<p>вкладку scripts для проведения автотестов запроса.</p> <p>2. Выяснили, что используя данные автотестов мы можем проверить качество отправки запроса и значения переменных, которые отправляются в запросе</p>
--	--

Программный коды:

Add a new pet to the store:

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

pm.test("Check that id equal 4444", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.category.id).to.eql(4444);
});

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

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

upload an image:

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

pm.test("Check that message contains 5f767c03faac3af4d721d9284d2dfa6e.jpg", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.message).to.contain("5f767c03faac3af4d721d9284d2dfa6e.jpg");
});
```

