МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра программных систем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**  
  
по лабораторной работе №6

«Тестирование API»

по дисциплине «Тестирование и отладка ПО»

Обучающаяся в группе 6301-020302D Соколова Алёна

Преподаватель Лобанков Антон Алексеевич

Самара 2024

**Цель работы**

В процессе выполнения заданий ознакомиться с тестированием API в Postman.

**Задание**

1. Скачать и установить Postman;

2. Создайте environment, в котором пропишите переменные, необходимые для формирования URL (хост, порт, протокол);

3. Создайте коллекцию для тестирования ресурса User программного интерфейса зоомагазина <https://petstore.swagger.io/#/user>. Добавьте в нее реквесты для тестирования всех доступных енд-поинтов;

4. Создайте pre-request script, для коллекции, который на основе переменных из environment будет формировать URL. Во всех реквестах заменить захардкоженную строку подключения, на переменную, которая будет формироваться в pre-request скрипте.

Решение

1. Для начала работы был скачан и установлен Postman. Это приложение используется для тестирования API, упрощая процесс отправки запросов и анализа ответов.

2. В Postman был создан environment с переменными, необходимыми для формирования URL:

* protocol: протокол, используемый для соединения (например, https);
* host: доменное имя или IP-адрес API (petstore.swagger.io/v2);
* port: порт для соединения.

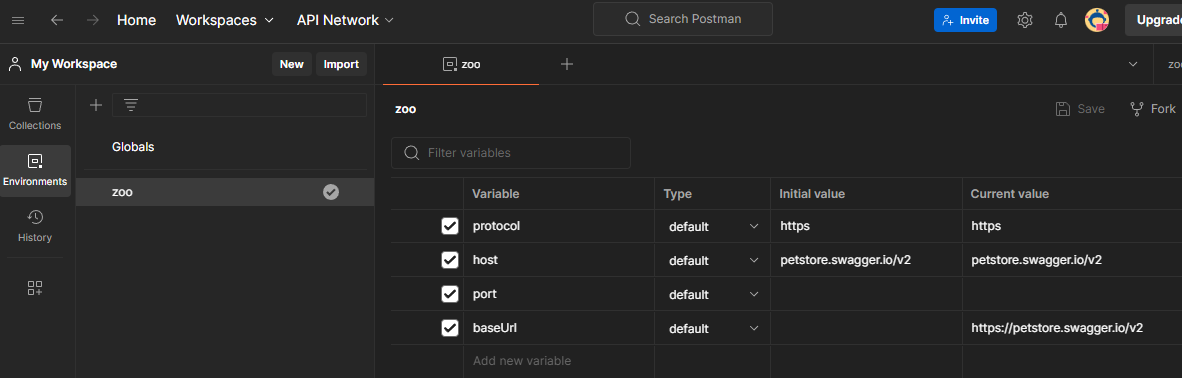


Рисунок 1 – Environment с переменными

3. В Postman была создана коллекция для тестирования ресурса User из API зоомагазина. В коллекцию добавлены запросы для следующих эндпоинтов:

* POST /user: создание пользователя;

{

"id": 11,

"username": "user1",

"firstName": "al",

"lastName": "sok",

"email": "alyona123@gmail.com",

"password": "qwerty",

"phone": "89012345678",

"userStatus": 1

}

* POST /user/createWithArray: создание пользователей;

[

{

"id": 1,

"username": "user2",

"firstName": "firstName1",

"lastName": "lastName1",

"email": "email1@gmail.com",

"password": "password1",

"phone": "1234567890",

"userStatus": 1

},

{

"id": 2,

"username": "user3",

"firstName": "firstName2",

"lastName": "lastName2",

"email": "email2@gmail.com",

"password": "password2",

"phone": "0987654321",

"userStatus": 1

}

]

* GET /user/user2: получение информации о пользователе;
* PUT /user/user1: обновление информации о пользователе;

{

"id": 11,

"username": "user1",

"firstName": "alyona",

"lastName": "sokolova",

"email": "al123@gmail.com",

"password": "qwerty",

"phone": "88011234567",

"userStatus": 1

}

* DELETE /user/user1: удаление пользователя;
* GET /user/ login?username=user1&password=qwerty: вход пользователя;
* GET /user/logout: выход пользователя.

Каждый запрос настроен в соответствии с документацией API, включая параметры, заголовки и тело запросов.

4. Для упрощения изменения параметров соединения был создан скрипт, формирующий URL на основе переменных из environment. Скрипт был добавлен на уровне коллекции, чтобы применяться ко всем запросам. Скрипт и список запросов отображены на рисунке 2.

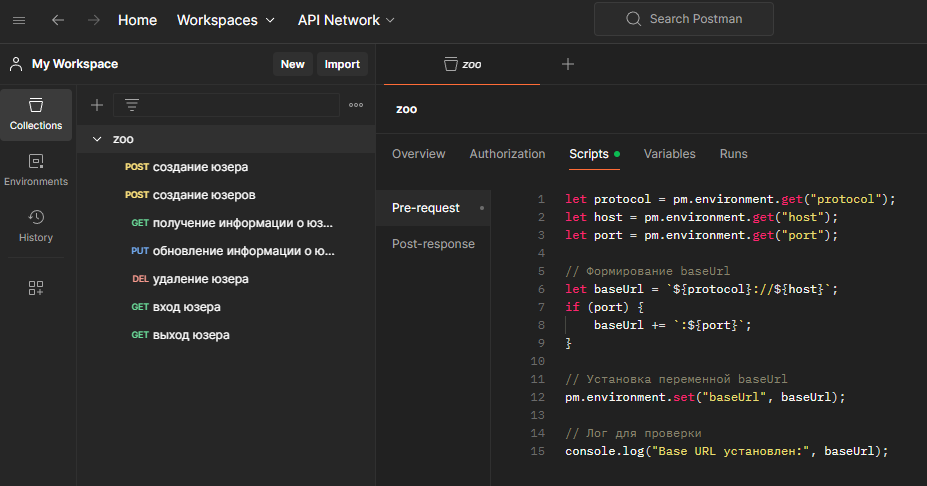


Рисунок 2 – Список запросов и Pre-request в коллекции zoo

После настройки был выполнен тестовый запуск запросов для проверки их работоспособности. Все эндпоинты корректно обработали запросы, а ответы соответствовали документации API.

Настройка Postman для тестирования ресурса User API зоомагазина завершена. Созданная коллекция и окружение обеспечивают удобство и гибкость в работе с API.

Результат работы отображен на рисунке 3.

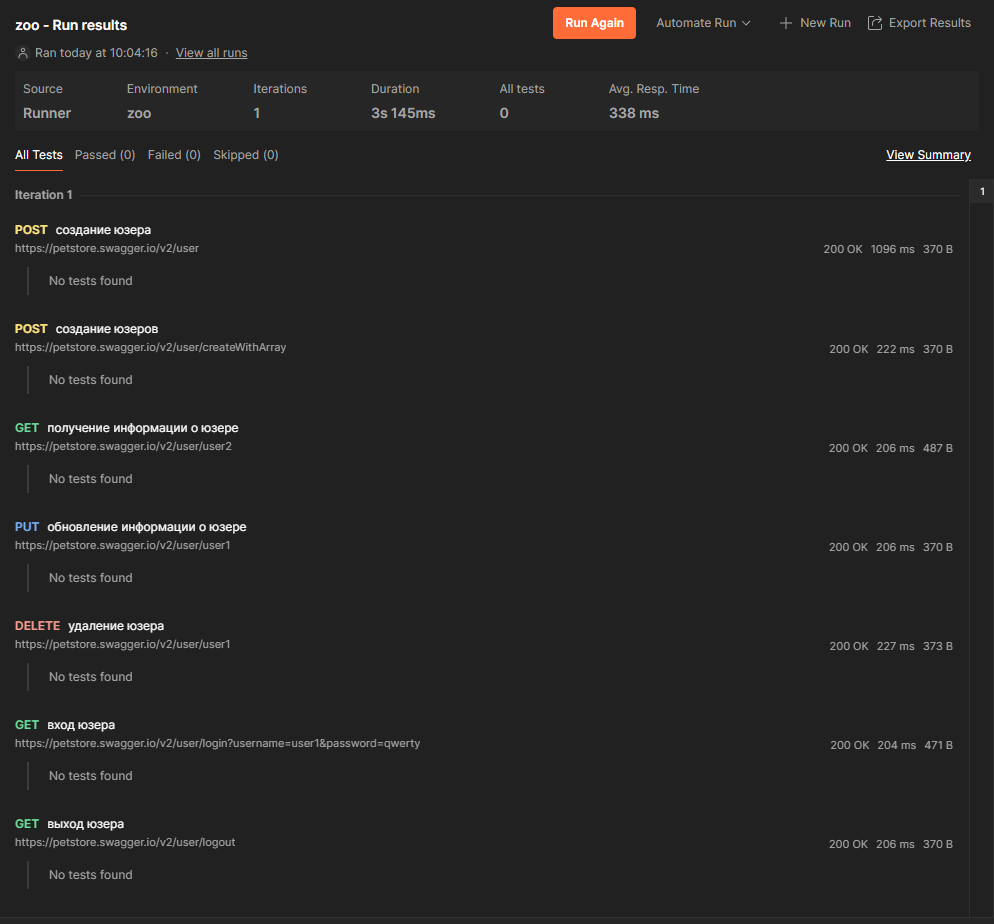


Рисунок 3 – Результат работы программы

# Заключение

В процессе выполнения лабораторных заданий был освоен инструмент Postman для тестирования API. Были созданы environment и коллекция для тестирования ресурсов API зоомагазина, что позволило эффективно настроить переменные и реализовать запросы к различным енд-поинтам. В результате выполнения задания также был разработан pre-request script, который динамически формирует URL, используя значения из environment, что повысило гибкость и удобство работы с API. Полученные результаты позволяют улучшить подход к тестированию веб-ресурсов, оптимизируя процессы взаимодействия с API и улучшая автоматизацию тестирования.