АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ API

AUTOMATED API TESTING

УДК 004.9

Сарычева Юлия Юрьевна, студент 4 курс, факультет «Информатика и управление», КФ МГТУ им. Н.Э. Баумана, Россия, г. Калуга

Липатова Софья Евгеньевна, студент 4 курс, факультет «Информатика и управление», КФ МГТУ им. Н.Э. Баумана, Россия, г. Калуга

Sarycheva Yu.Yu.

Lipatova S.E.

Аннотация: В данной работе рассматриваются особенности автоматизированного тестирования программных продуктов и приведены примеры реализации такого тестирования с помощью программы Postman. Тестирование программных продуктов является неотъемлемым этапом их создания и разработки. Наиболее простым способом тестирования является мануальное (ручное), однако оно способно быть неэффективным, поскольку напрямую зависит от классификации, внимания, профессионализма, логики и опыта тестировщика. Кроме того, ручное тестирование даже простых функций приложения занимает достаточно большое количество времени, поскольку тестировщик проверяет и анализирует каждый тестов кейс, разрабатывает их, строит схемы зависимостей. С целью снижения человеческого фактора при тестировании приложений и ускорения этого процесса была реализована возможность автоматизированного тестирования.

Annotation: This article is dedicated to the features of automated testing of software products and provides examples of the implementation of such testing using Postman. Testing is an integral stage of the creation of software and its development. The simplest way of testing is manual testing, but it can be ineffective since it directly depends on the classification, attention, professionalism, logic, and experience of the tester. In addition, manual testing of even simple application functions takes a lot of time, since the tester checks and analyzes each test case, develops them, and builds dependency diagrams. To reduce the human factor when testing applications and to speed up this process, the possibility of automated testing was implemented.

Ключевые слова: тестирование, автоматизация, API, протокол HTTP, Postman.

Key words: testing, automation, API, HTTP protocol, Postman.

Автоматизированное тестирование – метод тестирования программного обеспечения, при котором проверку ожидаемого результата с реальным выполняет не человек (как в мануальном тестировании), а специальная программа.

Автоматизированное тестирование – это эффективный по затратам метод регрессионного тестирования, несмотря на то, что изначально на автоматизацию требуется больше ресурсов. Но когда тестовые сценарии готовы, они смогут выполняться с большой скоростью и одинаковой точностью. Так что его стоимость будет постепенно окупаться. Также преимуществами автоматизированного тестирования являются отсутствие человеческого фактора при тестировании, благодаря этому будет повышаться качество его кода; автоматическое сохранение отчетов о результатах тестирования; а также возможность повторного использование тестов при изменении кода продукта [1, с. 158].

Процесс автоматизации состоит из следующих этапов:

1. Подготовка – выбор функционала, который подлежит автоматизированному тестированию; определение требований, сроков, инструментов; оценка рисков;

2. Проведение – запуск авто-тестов и проведение регрессионного тестирования;

3. Отчет – составление документации с обнаруженными багами и предложениями по улучшению функционала.

К автоматизации тестирования прибегают в нескольких случаях: данные требуют сложных математических расчетов; труднодоступность мест в системе; функциональность с высоким уровнем риска на баги; необходимость совершать множество однотипных операций, например, проверка валидации полей и ситуаций, когда программный продукт уже отлажен и сильные доработки не планируются [2, с. 33].

Существует несколько основных подходов в автоматизации:

–GUI-автоматизация (автоматизация тестирования графического пользовательского интерфейса (GUI));

–Тестирование кода;

–Автоматизированные тесты, взаимодействующие с API.

В рамках данной статьи будет рассмотрено тестирование API.

API – связующее звено между разработчиком и той средой, где оно должно взаимодействовать, оно разрабатывается или для клиента, или для внутреннего пользования. В основном, API – это часть сервера, получающая запросы и отправляющая в качестве ответов на них только данные [3, с. 54].

Существует ряд проблем, связанных с тестированием API:

1. Отсутствие GUI;

2. Обязательность проверки обработки исключений;

3. Необходимость знаний в программировании для тестировщика.

API обладает четким форматом передачи данных, что упрощает процесс обмена информацией.

Для этой цели используют запросы HTTP.

HTTP – протокол прикладного уровня модели OSI, используемый для передачи данных. Изначально с его помощью была возможна передача лишь гипертекстовых документов в формате HTML, однако со временем данное условие было доработано до передачи произвольных данных. Протокол HTTP является основой сети Интернет, обеспечивая клиент-серверное взаимодействие приложений.

Взаимодействие клиентской и серверной частей приложения осуществляется посредством отправки запросов и обработкой полученных на них ответов.

HTTP-запрос состоит из метода HTTP, пути к ресурсу, версии HTTP - протокола, заголовка и тела запроса (необязательно). HTTP-ответ содержит также версию протокола, код состояния (ошибки), сообщение состояния, заголовок и тело ответа [6, с. 49].

HTTP-метод – слово, определяющее операцию, выполняемую клиентом. Обычно является глаголом (GET, POST, и др.), однако может быть и существительным (OPTIONS, HEAD)

Наиболее используемыми методами HTTP являются:

–GET - получение и чтение данных;

–PUT - загрузка информации на указанный в запросе URI;

–POST - передача данных от пользователя к ресурсу;

–DELETE - удаляет ресурс;

–OPTIONS - определяет возможности веб-сервера [4].

Рассмотрим выполнение запросов с помощью приложения Postman. Postman – приложение, предназначенное для тестирования различных API, а также отправки различных запросов на сервер [5]. Важным свойством программы является возможность создания коллекций запросов к API, что значительно ускоряет разработку и оптимизирует время и ресурсы, зачастую тратимые на поиск запроса в памяти или ручное заполнение параметров. С Postman достаточно легко работать, поэтому многие тестировщики выбирают

его, как инструмент автоматизации. Основное окно приложения представлено на рисунке 1.

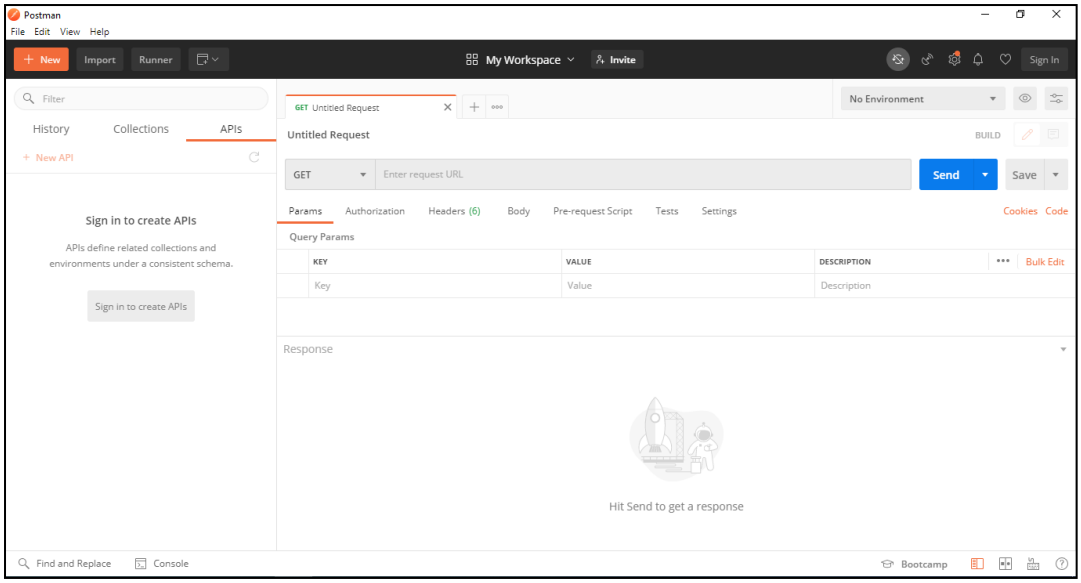


Рисунок 1 – Окно приложения Postman.

С целью понимания структуры выполнения запроса в Postman как при стандартном выполнении, так и при проведении тестирования, рассмотрим ключевые понятия, используемые в нем. К ним относят Коллекции (Collections), Папки (Folders) и Запросы (Requests). Коллекция – это файл проекта, в который входят все запросы, в ней есть как свои скрипты, так и свои переменные. Также Коллекции сохраняют историю прошлых запросов. Папка объединяет запросы в одну группу внутри коллекции, у нее могут быть свои скрипты, но не переменные. Запросы создаются в конструкторе, но также у них есть собственные скрипты. Схема взаимодействия этих элементов между собой представлена на рисунке 2.

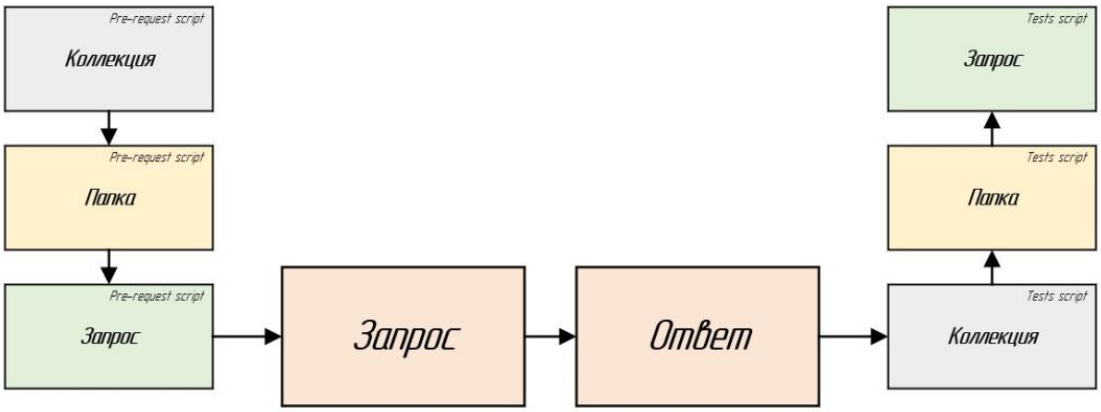


Рисунок 2 – Схема зависимостей в Postman.

Рассмотрим выполнение запросов в Postman. Для этого используем метод getNameByBookId, использующийся для получения названия произведения по его идентификатору. В качестве параметров запроса передается пользовательский ключ (userKey) и идентификатор произведения (bookId). В качестве ответа ожидается строка, содержащая название произведения.

Пример выполнения GET-запроса представлен на рисунке 3.

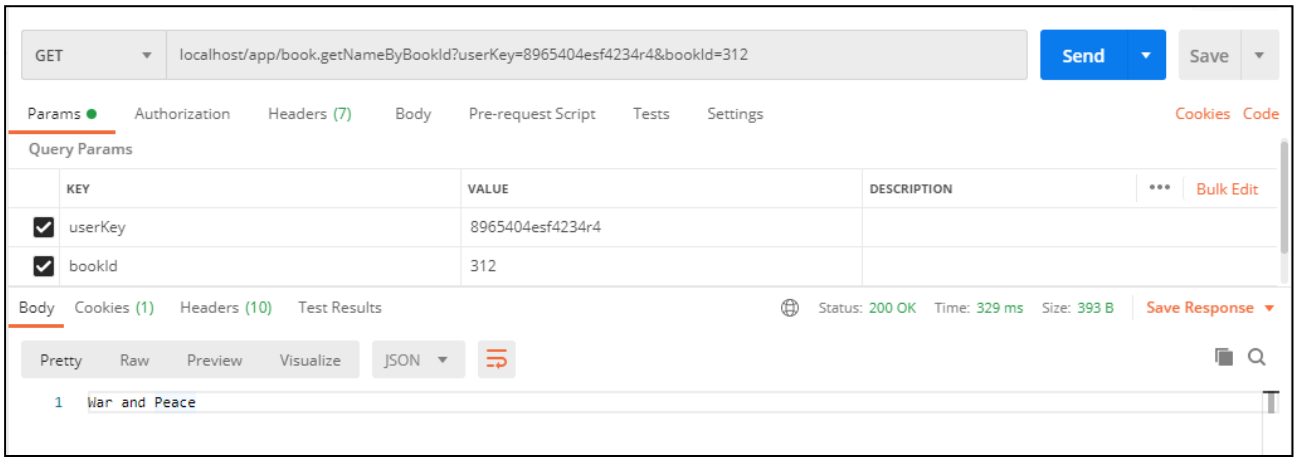


Рисунок 3 – Пример выполнения GET-запроса.

Помимо выполнения запросов Postman позволяет автоматизировать тестирование методов API, что позволяет значительно ускорить процесс разработки программного продукта.

Для выполнения запроса и автоматизированного теста необходимо:

1. Вынести URL и токен как переменные окружения;
2. Проставить необходимый для запроса метод;

3. Указать параметры запроса;

4. Во вкладке Tests прописать код, выполняющий тестирование метода (его результатов).

5. Отправить запрос;

6. По выполнению запроса просмотреть результаты: ответ (Body) и результаты тестирования (Test Results).

Рассмотрим создание и выполнение теста на примере ранее рассмотренного метода getNameByBookId.

Одним из наиболее простых и понятных тестов является проверка статуса выполненного запроса. При успешном выполнении запроса ответ будет содержать код ошибки: 200.

Листинг теста:

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

При успешном выполнении такого теста пользователь может увидеть статус теста PASS, а также сообщение «Status code is 200».

Указываем данных листинг в разделе Tests страницы запроса в Postman и отправляет запрос на сервер. После его выполнения можно увидеть, что Вкладка Test Results подсвечивает «(1/1)», показывая, что тест был выполнен. При переходе можно увидеть результат успешного выполнения теста (рисунок 4).

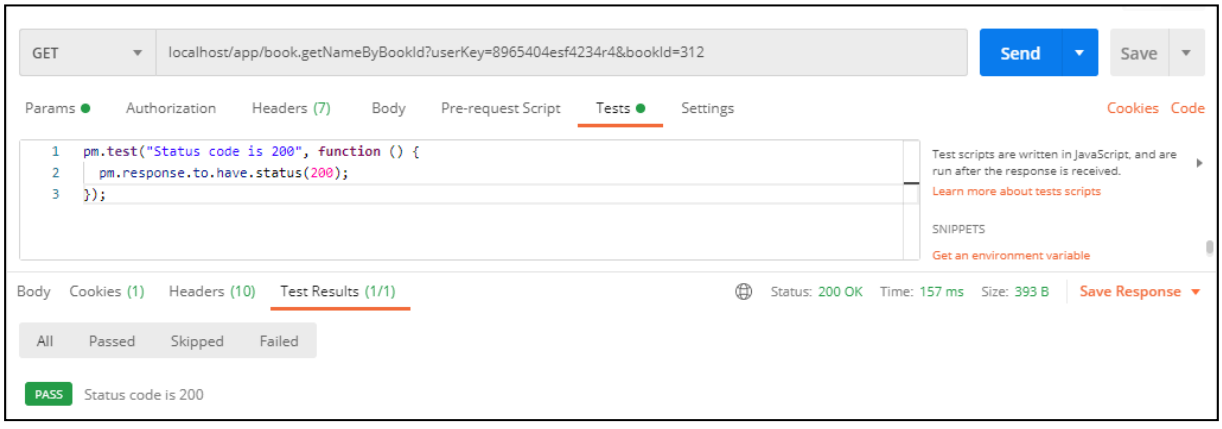


Рисунок 4 – Результат успешного выполнения теста.

Рассмотрим вариант, когда тест был провален. Для этого изменим листинг теста так, чтобы тест ожидал, что запрос вернется с кодом ошибки 500. Однако сам запрос при этом вернет код 200. Результат такого теста представлен на рисунке 5.

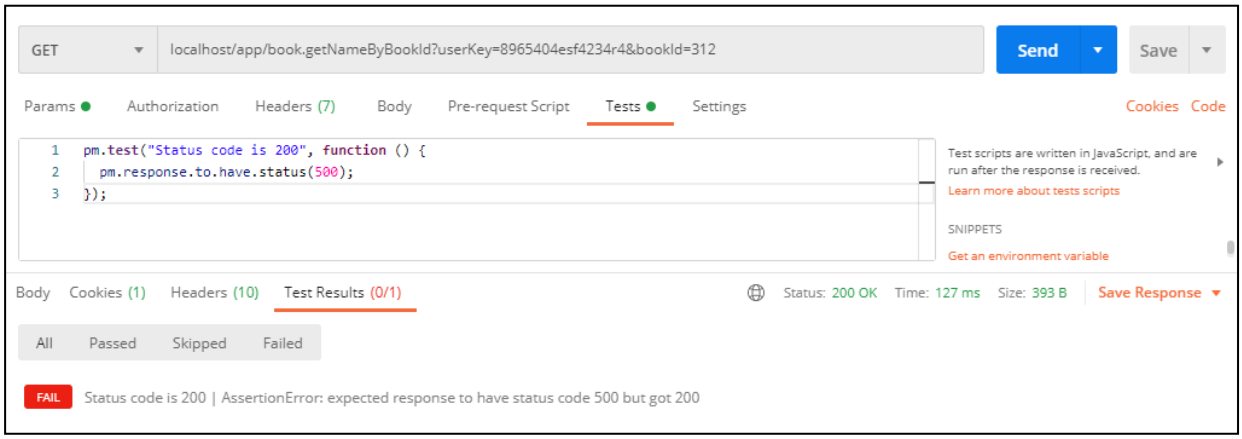


Рисунок 5 – Вид окна Postman при провале теста.

По рисунку 5 можно увидеть, что вкладка Test Results теперь подсвечивает «(0/1)», показывая, что тест был провален, статус теста изменился на «FAIL», а рядом с сообщением, показывающим, что тест был выполнен, можно увидеть комментарий, сообщающий об ошибке, которая помешала успешному выполнению теста.

Важно отметить, что такие тесты в Postman могут иметь более сложную структуру. В них может происходить проверка нескольких условий, анализ данных и т.д. Однако, несмотря на сложность составления таких тестов их важным преимуществом является значительное ускорение процесса разработки, тестирования, а также минимизация человеческого фактора при проверке ошибок.

Таким образом, можно сделать вывод, что автоматизированное тестирование API имеет больше преимуществ, чем недостатков, так как его стоимость довольно быстро окупается, а скорость превосходит скорость ручного тестирования. Единственный недостаток – необходимость умения

программировать для тестировщика, но для этого существует множество инструментов, которые могут помочь в создании тестов. Одним из них можно назвать программу Postman, которая помогает создавать, тестировать и документировать приложение с помощью одного ресурса.

Список литературы

1. Лайза Криспин, Джанет Грегори. Agile-тестирование. Обучающий курс для всей команды – МИФ, 2019 г., 530 стр.
2. Элфрид Дастин, Джефф Рэшка, Джон Пол. Автоматизированное тестирование программного обеспечения – Лори, 2003 г., 592 стр.
3. Старолетов С.М. Основы тестирования и верификации программного обеспечения – Лань, 2018 г., 344 стр. [Электронный ресурс]. – URL: https://e.lanbook.com/reader/book/110939/#1
4. What Is Automation Testing (Ultimate Guide To Start Test Automation) [Электронный ресурс]//Software Testing Help: сайт – Режим доступа: https://www.softwaretestinghelp.com/automation-testing-tutorial-1/
5. Postman [Электронный ресурс]//Официальный сайт Postman - Режим доступа: https://www.getpostman.com/
6. Барри Поллард. HTTP/2 в действии – ДМК-Пресс, 2021 г., 424 стр.

Bibliography

1. Lisa Crispin, Janet Gregory. Agile testing. Training course for the whole team - MYTH, 2019, 530 pp.
2. Elfrid Dustin, Jeff Raschka, John Paul. Automated Software Testing - Laurie 2003, 592 pp.
3. Staroletov S.M. Fundamentals of software testing and verification - Lan, 2018, 344 pages [Electronic resource]. - URL: https://e.lanbook.com/reader/book/110939/#1
4. What Is Automation Testing (Ultimate Guide To Start Test Automation) [Electronic resource] // Software Testing Help: website - Access mode: https://www.softwaretestinghelp.com/automation-testing-tutorial-1/

5. Postman [Electronic resource] // Postman official website - Access mode: https://www.getpostman.com/
6. Barry Pollard. HTTP / 2 in Action - DMK-Press, 2021, 424 pp.