

## Использование POSTMAN в REST технологиях

*Маринчук Александр Сергеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Научный руководитель:*

*Глаголев Владимир Александрович*

*Приамурский государственный университет имени Шолом-Алейхема*

*к.г.н., доцент кафедры информационных систем, математики и методик обучения*

### **Аннотация**

В данной статье описано использование программы POSTMAN в REST технологиях с рассмотрением примеров. Были рассмотрены понятия, касающиеся данной темы, и сделаны выводы касательно полезности и применимости POSTMAN. Также данная программа была протестирована на сервисе Яндекс Словарь и в результате ее выполнения были описаны полученные результаты.

**Ключевые слова:** POSTMAN, REST технологии, API, коллекции, запросы.

## Using POSTMAN in REST Technologies

*Marinchuk Alexander Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Scientific adviser:*

*Glagolev Vladimir Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*Ph. D., associate Professor of the Department of Information Systems, Mathematics and Legal Informatics*

### **Abstract**

This article describes the use of the POSTMAN program in REST technologies with a discussion of examples. Concepts related to this topic were reviewed and conclusions were made regarding the usefulness and applicability of POSTMAN. Also, this program was tested on the Yandex Dictionary service and, as a result of its execution, the results were described.

**Keywords:** POSTMAN, REST technologies, API, collections, requests..

## **1. Введение**

### **1.1 Актуальность исследования**

На сегодняшний день становится все более востребованным проведение тестирования различных API приложений. Тесты помогают выявить недоработки и несовершенства различных систем, начиная от простых сайтов и заканчивая гигантскими проектами. С последними как раз и возникают сложности в плане количества проводимых тестов. Количество запросов бывает просто огромно, иногда даже необходимо повторять некоторые из них, изменяя различные параметры. Для начала следует разобраться, что же такое REST.

Говоря простым языком REST это архитектура, то есть принципы построения распределенных гипермедиа систем, того что другими словами называется World Wide Web, включая универсальные способы обработки и передачи состояний ресурсов по HTTP. В данной работе было рассмотрено использование POSTMAN именно в REST API, но прежде чем приступить к практической части необходимо рассмотреть термин API [1].

API (программный интерфейс приложения, интерфейс прикладного программирования) — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой [2].

### **1.2 Обзор исследований**

В своей статье Р.С. Поскребышев и В.Г. Тарасов рассмотрели основные подходы к построению сервисов SOAP и REST, а также относительно новый инструмент - Swagger, и проанализировали простоту и удобство в их использовании [3]. Привел обзор методов композиции RESTful сервисов, провел анализ подходов реализации сервисов в архитектуре REST, а также рассмотрел высокоуровневые методы на основе семантических аннотаций в своей статье А.Л. Чеглаков [4]. В статье Н.Н. Гурьева, А.А. Кочеткова, А.А. Удалова рассмотрены особенности построения синхронных и асинхронных запросов, основные аспекты построения веб-приложений с архитектурой REST, а также способы доступа из веб-приложения к удаленным ресурсам и сервисам с помощью технологии XMLHttpRequest и с использованием стека технологий AJAX [5]. Статья В.В. Федотова рассказывает о возможностях применения протокола SOAP и метода взаимодействия REST при проектировании распределённого Веб-приложения для интеграции информационных ресурсов об объектах природного и культурного наследия [6]. В своей статье Ю.Р. Акинин, А.В. Барабанов, Н.И. Гребенникова рассмотрели пример быстрого создания REST API сервиса на основе облачных технологий Microsoft Azure. Отличительной особенностью решения являются простота использования, инсталляции и поддержки, а также гибкость и масштабируемость [7].

### 1.3 Цель исследования

Целью данной статьи является знакомство с POSTMAN в REST технологиях и выполнение различных запросов к существующим системам, включающих REST API.

## 2. Методы исследования

Инструментов для выполнения запросов станет бесплатное приложение POSTMAN. POSTMAN – это мощный набор инструментов тестирования API, ставший необходимым для многих разработчиков. POSTMAN используется более чем миллионом разработчиков по всему миру, и это число постоянно растет.

Для начала работы следует скачать и установить приложение с официального сайта (рис. 1).

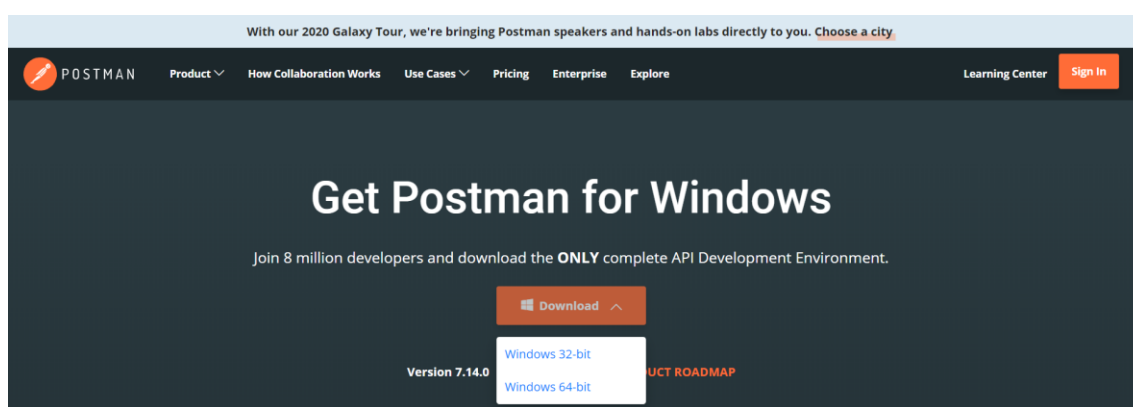


Рисунок 1 – Скачивание приложения

После установки программа попросит пройти простую регистрацию в которую нужно указать почту, логин, пароль. После этого откроется окно главного меню (рис. 2).

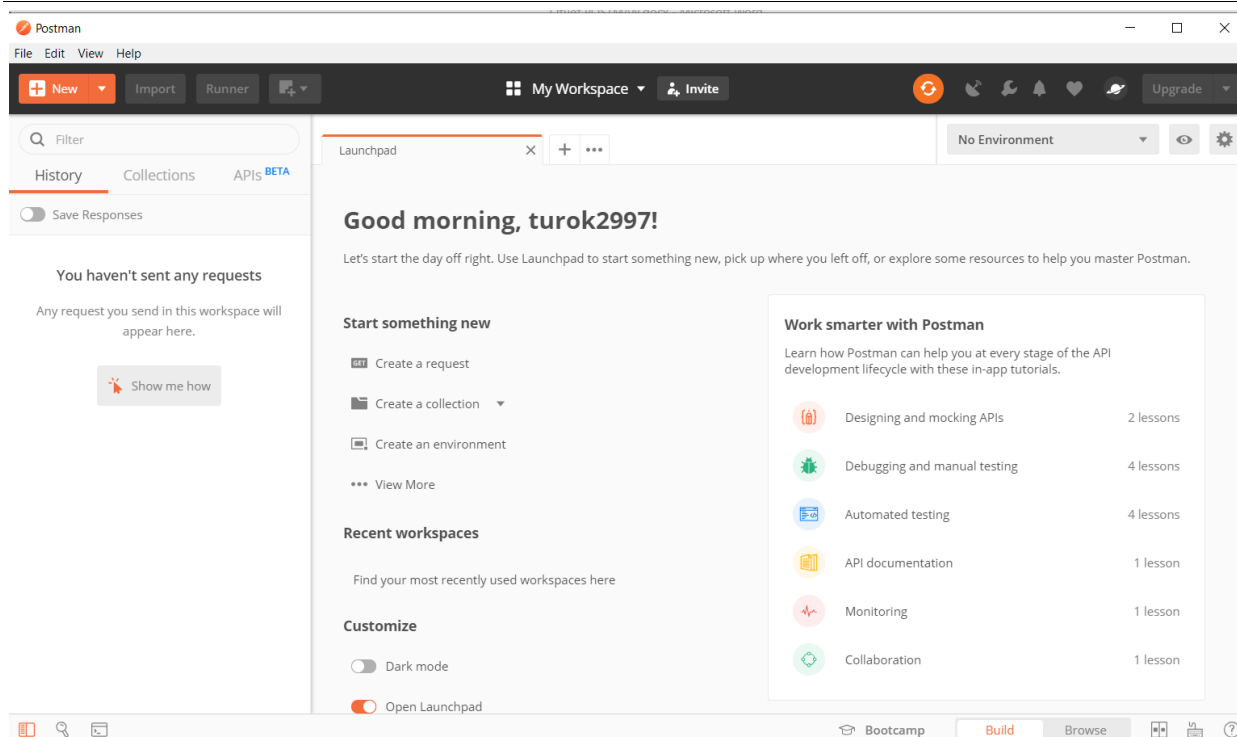


Рисунок 2 – Окно работы с POSTMAN

Главные понятия, которыми оперирует Postman это Collection (коллекция) на верхнем уровне, и Request (запрос) на нижнем. Вся работа начинается с коллекции и сводится к описанию вашего API с помощью запросов (рис. 3)

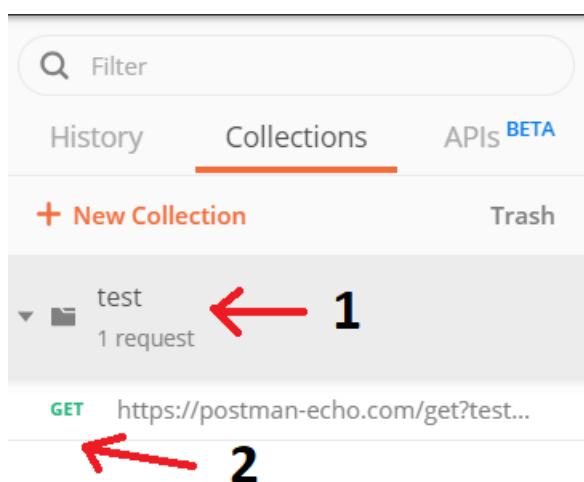


Рисунок 3 – Коллекции (1) и запросы (2)

Коллекция — отправная точка для нового API. Можно рассматривать коллекцию, как файл проекта. Коллекция объединяет в себе все связанные запросы.

Запрос — основная составляющая коллекции, то ради чего все и затевалось. Запрос создается в конструкторе. Конструктор запросов это главное пространство, с которым придётся работать.

Рассмотрим простой GET-запрос к сервису Postman Echo, который служит для пользователей, только начавших обучение в тестировании. Передадим переменную test равную 123 (рис. 4)

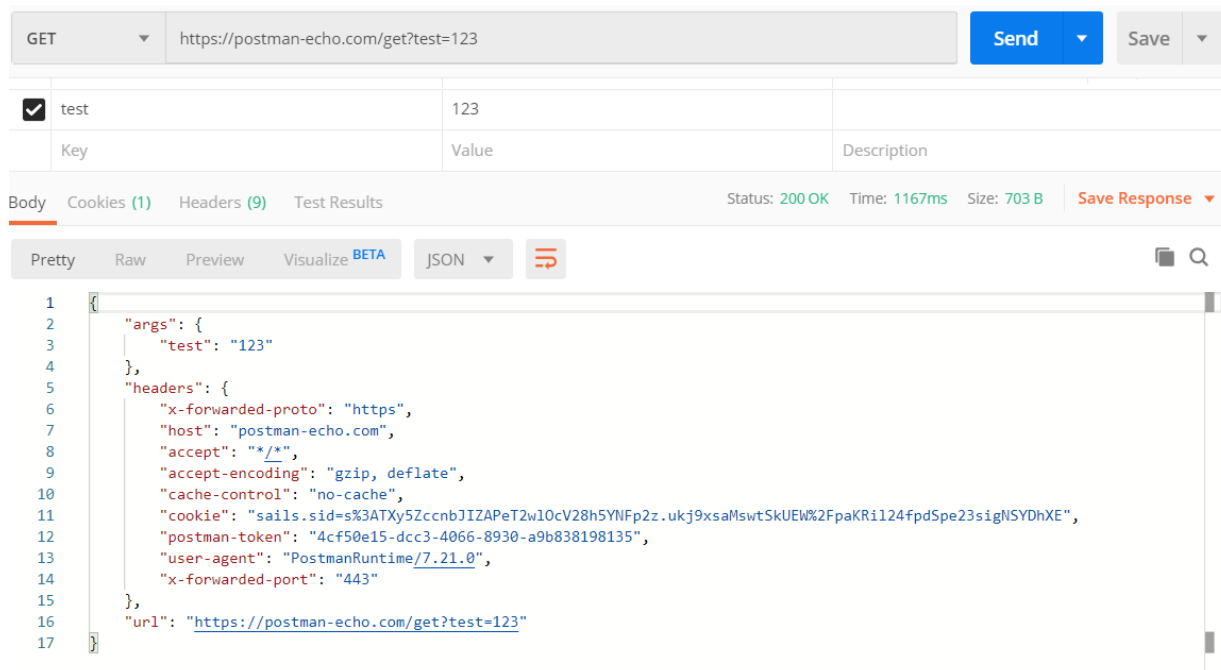


Рисунок 4 – Запрос к серверу и ответ

Сразу же получаем ответ в формате JSON, где можно посмотреть заголовки, cookie, аргументы и url. Кратко опишем основные поля, существующие в POSTMAN.

1. Раздел Body, где выводится весь ответ, полученный от сервера.
2. Раздел Cookies служит для представления в удобной форме переданных сервером cookie (рис. 5).

Name	Value	Domain	Path	Expires	HttpOnly	Secure
sails.sid	s%3Awp4eqM0k1zCrMXAI0kAQ12Q__0Huzzh.MwOGkjNsXaAByKuHx%2BleVCNEkKyLC79oBm%2FdEFEWLN8	postman-echo.com	/	Never	true	false

Рисунок 5 – Раздел Cookie

3. Раздел Headers, служащий для вывода заголовков в виде пары столбцов key и value (рис. 6).

KEY	VALUE
Content-Encoding	gzip
Content-Type	application/json; charset=utf-8
Date	Tue, 24 Dec 2019 01:24:03 GMT
ETag	W/"1b0-eWp1WQyaH9kc9mjDPQN+C7Hrves"
Server	nginx
set-cookie	sails.sid=s%3Avp4eqM0k1zCrMXAl0kAQ12Q__-0Huzzh.MwOGkjNsXaAByKu...
Vary	Accept-Encoding
Content-Length	329
Connection	Close

Рисунок 6 – Раздел Headers

4. Раздел Test Result предназначен для написания скриптов, которые автоматизируют тестирование (рис. 7).

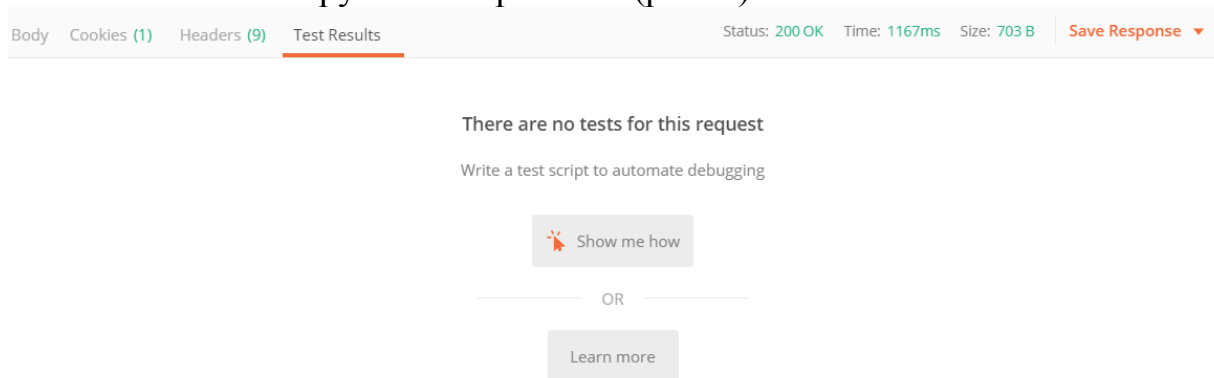


Рисунок 7 – Раздел Test Result

Теперь попробуем написать простой запрос для определения части речи слова, обращаясь к сервису Яндекс Словарь. Для начала следует получить API-ключ в пару кликов, перейдя по ссылке [1]. Далее вбиваем запрос в строку POSTMAN, передавая сервису полученный API-ключ, язык с которым будем работать и само слово для проверки. (рис. 8).

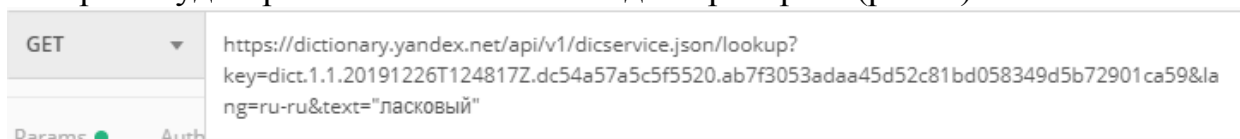


Рисунок 8 – Формирование запроса

Затем переходим на вкладку Tests, которая служит для написания кода в привычном для нас виде. Используя метод вложенных условий проверим код ответа сервиса, пустое ли тело ответа и соответствует ли выбранное слово части речи прилагательное (рис. 9).

```
1  if (responseCode.code === 200)
2  {
3      tests["Код ответа 200"] = true;
4      if (responseBody !== ''){
5          tests ["Тело ответа не пустое"] = true;
6          var testsStr = 'adjective';
7          var responseBodyJSON = JSON.parse(responseBody);
8          var strfromJSON = responseBodyJSON.def[0].pos;
9          if(testsStr === strfromJSON){
10             tests["Ласковый - это часть речи прилагательное"] = true;
11
12             }
13             else
14             {
15                 tests["Ласковый - это часть речи не прилагательное"] = false;
16             }
17         }
18         else
19         {
20             tests ["Тело ответа пустое"] = false;
21         }
22     }
23     else
24     {
25         tests["Код ответа не 200"];
26     }
```

Рисунок 9 – Код для получения ответа от сервиса

10) Нажимая кнопку Send, мы получим результаты в поле Test Results (рис.

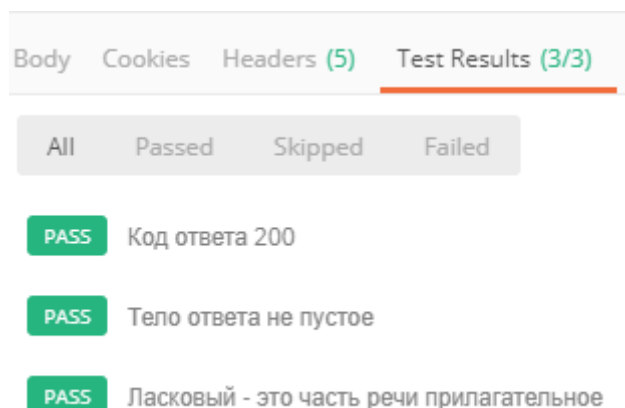


Рисунок 10 – Ответ от сервиса

Также протестируем запрос на вывод ошибок, отправив слово “машина”, которое является существительным (рис. 11).



Рисунок 11 – Ошибка, получаемая от сервиса

### 3. Выводы

POSTMAN - это мощный инструмент, как для тестирования, так и для описания запросов к API проекта. Даже если вы не будете использовать средства тестирования, которые предоставляет Postman, то иметь описание запросов в виде коллекции будет весьма полезно. Коллекции Postman это живая, интерактивная документация к API, которая сэкономит множество времени, и в разы ускорит разработку, отладку и тестирование

### Библиографический список

1. POSTMAN URL: <https://www.getpostman.com/> (дата обращения: 13.01.2020).
2. API // Википедия URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 13.01.2020).
3. Поскребышев Р.С., Тарасов В.Г. API на основе SOAP и REST // Молодые ученые - ускорению научно-технического прогресса в XXI веке. Ижевск: ИННОВА, 2016. С. 404-410.
4. Чеглаков А.Л. Композиция WEB-сервисов на основе архитектуры REST // Инновационная наука. 2016. №12-2 . С. 118-120.
5. Гурьев Н.Н., Кочетков А.А., Удалов А.А. Особенности построения синхронных и асинхронных запросов при работе с REST API // Молодежный научно-технический вестник. 2017. №2. С. 14.
6. Федотов В.В. Применение SOAP и REST для интеграции ресурсов об объектах природного и культурного наследия // Байкальские встречи - VIII: историко-культурное наследие региона как фактор социально-экономического развития. Улан-Удэ: Восточно-Сибирский государственный институт культуры, 2014. С. 763-769.
7. Акинин Ю.Р., Барабанов А.В., Гребенникова Н.И. Быстрое создание REST API сервиса на основе облачных технологий Azure // Вестник Воронежского государственного технического университета. 2012. №12-1. С. 66-68.