

# Postman. Endpoint, pre-request script, body, test.

**Postman** - это масштабируемый инструмент тестирования API, который быстро интегрируется в конвейер CI/CD.

Он был создан индийским программистом Abhinav Asthana (сейчас живет и работает в Силиконовой долине) в 2012 году как побочный проект для упрощения рабочего процесса при разработке и тестировании API.

Основное предназначение приложения — создание коллекций с запросами к вашему API. Любой разработчик или тестировщик, открыв коллекцию, сможет с лёгкостью разобраться в работе вашего сервиса. Ко всему прочему, Postman позволяет проектировать дизайн API и создавать на его основе Mock-сервер. Разработчикам больше нет необходимости тратить время на создание "заглушек". Реализацию сервера и клиента можно запустить одновременно. Тестировщики могут писать тесты и производить автоматизированное тестирование прямо из Postman. А инструменты для автоматического документирования по описаниям из коллекций сэкономят время на ещё одну "полезную фичу". Есть кое-что и для администраторов — авторы предусмотрели возможность создания коллекций для мониторинга сервисов.

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.

Сегодня Postman — супер-популярный инструмент. Им пользуются более 8 миллионов разработчиков и тестировщиков. И вот почему:

## Плюсы Postman

- Бесплатный. Postman — бесплатный инструмент.
- Простой в использовании. Очень просто начать пользоваться — Postman интуитивно понятный. Уже через несколько минут после скачивания и установки вы сможете отправить ваш первый запрос.
- Поддерживает разные API. С помощью Postman можно выполнять разные типы запросов к любым API (REST, SOAP, GraphQL (по тестированию GraphQL с помощью Postman у нас есть отдельная статья))
- Расширяемый. Postman можно настроить под ваши конкретные нужды с помощью Postman API.
- Интегрируемый. Можно легко интегрировать наборы тестов в ваш любимый CI/CD инструмент с помощью Newman (CLI collection runner — позволяет запускать Postman-коллекции в командной строке)
- Имеет большое комьюнити. Postman очень популярный и, как следствие, имеет большое комьюнити, которое подскажет ответы на большинство вопросов.

## Endpoint

Адрес, на который посылаются сообщения называется Endpoint.

Обычно это URL (например, название сайта) и порт. Если я хочу создать веб сервис на порту 8080 Endpoint будет выглядеть так:

`http://andreyolegovich.ru:8080`

Слово адрес нужно понимать в общем смысле - как при отправке бумажного письма нужно написать на конверте физический адрес, так и при обращении к сервису у которого больше одного интерфейса нужно указать нужный.

Если моему Web сервису нужно будет отвечать на различные сообщения я создам сразу несколько URL (interfaces) по которым к сервису можно будет обратиться. Например:

`https://andreyolegovich.ru:8080 /resource1/status`

`https://andreyolegovich.ru:8080 /resource1/getserviceInfo`

`https://andreyolegovich.ru:8080 /resource1/putID`

`http://andreyolegovich.ru:8080 /resource1/eventslist`

...

Как видите у моих эндпойнтов (Endpoints) различные окончания. Такое окончание в Endpoint называются Resource, а начало Base URL.

Такое определение Endpoint и Resource используется, например, в SOAP UI для RESTful интерфейсов

`https://andreyolegovich.ru:8080` - это Base URL

`/resource1/status` - это Resource

Endpoint = Base URL + Resource

Понятие Endpoint может использоваться в более широком смысле. Можно сказать, что какой-то определённый роутер или компьютер является Endpoint. Например, в понятии Endpoint Management под Endpoint имеют в виду конечное устройство. Обычно это понятно из контекста.

Также следует обратить внимание на то, что понятие Endpoint выходит за рамки RESTful и может использовать как в SOAP так и в других протоколах.

На программистском сленге Endpoint иногда называют ручкой.

Сделать какой-то запрос, например HTTP, на сленге будет - дёрнуть ручку

В протоколе HTTP предусмотрено несколько способов отправить запрос на один и тот же Endpoint.

Они называются:

- CONNECT
- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT
- TRACE

Когда мы знаем какие методы с какими Endpoint можно использовать составить запросы не составит труда.

Например:

GET http://andreyolegovich.ru:8080 /resource1/status

GET http://andreyolegovich.ru:8080 /resource1/getserviceInfo

PUT http://andreyolegovich.ru:8080 /resource1/putID

GET http://andreyolegovich.ru:8080 /resource1/eventslist

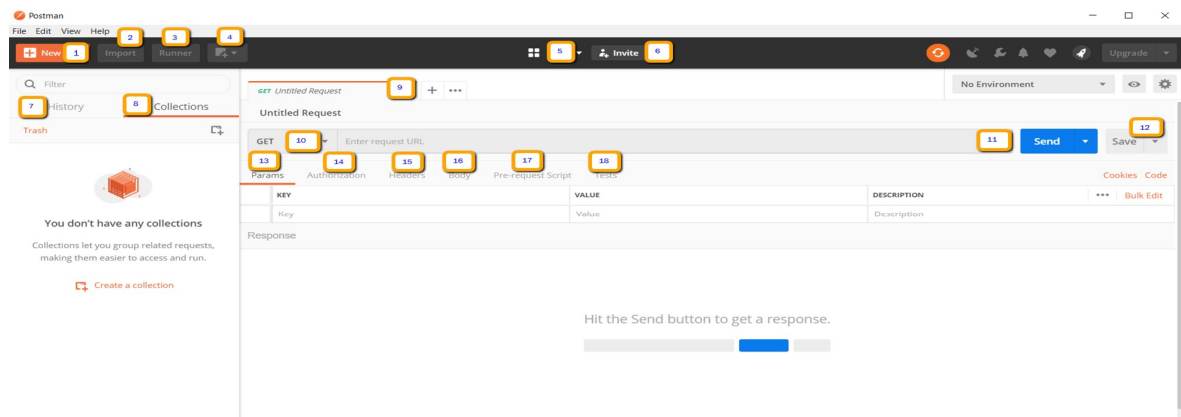
POST http://andreyolegovich.ru:8080 /resource1/eventslist

...

Таким образом простейший запрос состоит из метода и Endpoint

Request = Method + Endpoint

## Как использовать Postman для выполнения запросов к API?



1. New – Здесь вы создадите новый запрос, коллекцию или среду.
2. Import – Это используется для импорта коллекции или среды. Существуют такие опции, как импорт из файла, папки, ссылка или вставка необработанного текста.
3. Runner – Тесты автоматизации могут быть выполнены с помощью программы сбора данных. Это будет обсуждаться далее в следующем блоке.
4. Open New – Откройте новую вкладку, Окно Postman или Окно Runner, нажав эту кнопку.
5. My Workspace – Вы можете создать новое рабочее пространство индивидуально или в команде
6. Invite – Сотрудничайте в рабочей области, приглашая членов команды.
7. History – Прошлые запросы, которые вы отправляли, будут отображаться в истории. Это позволяет легко отслеживать выполненные вами действия.
8. Collections – Организуйте свой набор тестов, создавая коллекции. Каждая коллекция может содержать вложенные папки и несколько запросов. Запрос или папка также могут быть продублированы.
9. Request tab – Здесь отображается заголовок запроса, над которым вы работаете. По умолчанию “Untitled Request” будет отображаться для запросов без названий.
10. HTTP Request – При нажатии на эту кнопку отобразится раскрывающийся список различных запросов, таких как GET, POST, COPY, DELETE, etc. В Postman API тестировании, наиболее часто используемые запросы - GET и POST.
11. **Request URL (Endpoint)** – Также известный как конечная точка, это место, где вы определите ссылку, с которой будет взаимодействовать API.

12. **Save** – Если в запросе есть изменения, необходимо нажать кнопку Сохранить, чтобы новые изменения не были потеряны или перезаписаны.
13. **Params** – Здесь вы напишете параметры, необходимые для запроса, такие как значения ключей.
14. **Authorization** – Для доступа к API-интерфейсам необходима авторизация. Это может быть в виде имени пользователя и пароля, токена на предъявителя (bearer token) и т.д.
15. **Headers** – Вы можете задать заголовки, такие как тип контента JSON, в зависимости от запроса.
16. **Body** – Именно здесь можно настроить детали в запросе, обычно используемом в запросе POST.
17. **Pre-request Script** – Это сценарии, которые будут выполняться перед запросом. Обычно сценарии предварительного запроса для среды настройки используются для обеспечения того, чтобы тесты выполнялись в правильной среде.
18. **Tests** – Это сценарии, выполняемые во время запроса. Важно иметь тесты, поскольку он устанавливает контрольные точки, чтобы проверить, в порядке ли статус ответа, получены ли ожидаемые данные и другие тесты.