

Postman

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. Вашим 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 ваших коллекций сэкономят время на ещё одну "полезную фишку". 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.

Collection

Коллекция — отправная точка для нового API. Можно рассматривать коллекцию, как файл проекта. Коллекция объединяет в себе все связанные запросы. Обычно API описывается в одной коллекции, но если вы желаете, то нет никаких ограничений сделать по-другому. Коллекция может иметь свои скрипты и переменные.

Folder

Папка — используется для объединения запросов в одну группу внутри коллекции. К примеру, вы можете создать папку для первой версии своего API — "v1", а внутри сгруппировать запросы по смыслу выполняемых действий — "Order & Checkout", "User profile" и т. п. Всё ограничивается лишь вашей фантазией и потребностями. Папка, как и коллекция может иметь свои скрипты, но не переменные.

Request

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

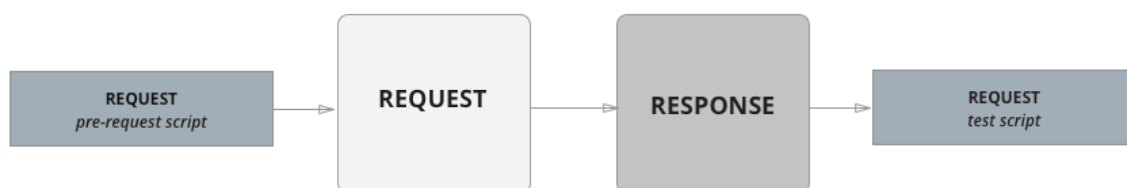
запросы с помощью всех стандартных HTTP методов, все параметры запроса под вашим контролем. Вы с лёгкостью можете поменять или добавить необходимые вам заголовки, cookie, и тело запроса. У запроса есть свои скрипты. Обратите внимание на вкладки "Pre-request Script" и "Tests" среди параметров запроса. Они позволяют добавить скрипты перед выполнением запроса и после. Именно эти две возможности делают Postman мощным инструментом помогающим при разработке и тестировании.

Pre-request Script

Pre-request скрипты пишутся на JavaScript и выполняются перед отправкой запросов. Используются для того, чтобы провести какие-то действия прямо перед тем, как отправить запрос (например, добавить timestamp или какие-то вычисляемые данные в ваши запросы, можно сделать запрос к другой системе и использовать результат его выполнения в основном запросе).

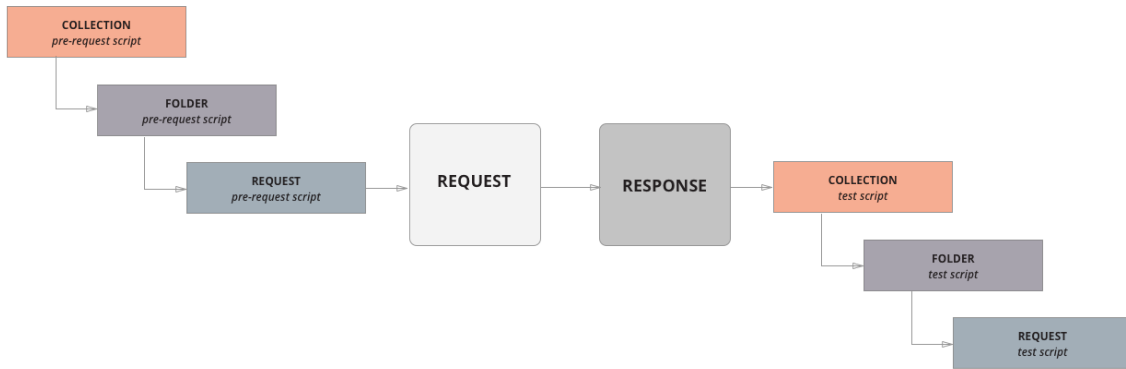
Tests

Во вкладке Tests находятся скрипты, которые выполняются во время запроса. Тесты позволяют проверить API и убедиться, что все работает так, как это было задумано. Используется для написания тестов, проверки результатов, и при необходимости их сохранения в переменные.



Последовательность выполнения запроса

Помимо скриптов на уровне запроса, мы можем создавать скрипты на уровне папки, и, даже, на уровне коллекции. Они называются также — "Pre-request Script" и "Tests", но их отличие в том, что они будут выполняться перед каждым и после каждого запроса в папке, или, как вы могли догадаться, во всей коллекции.



Последовательность выполнения запроса со скриптами папок и коллекций

Endpoint

Endpoint можно рассматривать как средство, с помощью которого API может получить доступ к ресурсам, необходимым им с сервера для выполнения своей задачи. Endpoint API - это, по сути, модное слово для обозначения URL-адреса сервера или службы. API-интерфейсы работают через "запросы" и "ответы". И когда API запрашивает доступ к данным из веб-приложения или сервера, ответ всегда отправляется обратно. Местоположение, в которое API отправляет запрос и откуда исходит ответ, называется Endpoint.

Body

Body -это фактическое тело запроса (обычно оно определяет полезную нагрузку запроса или данные, которые должны быть обработаны запросом).