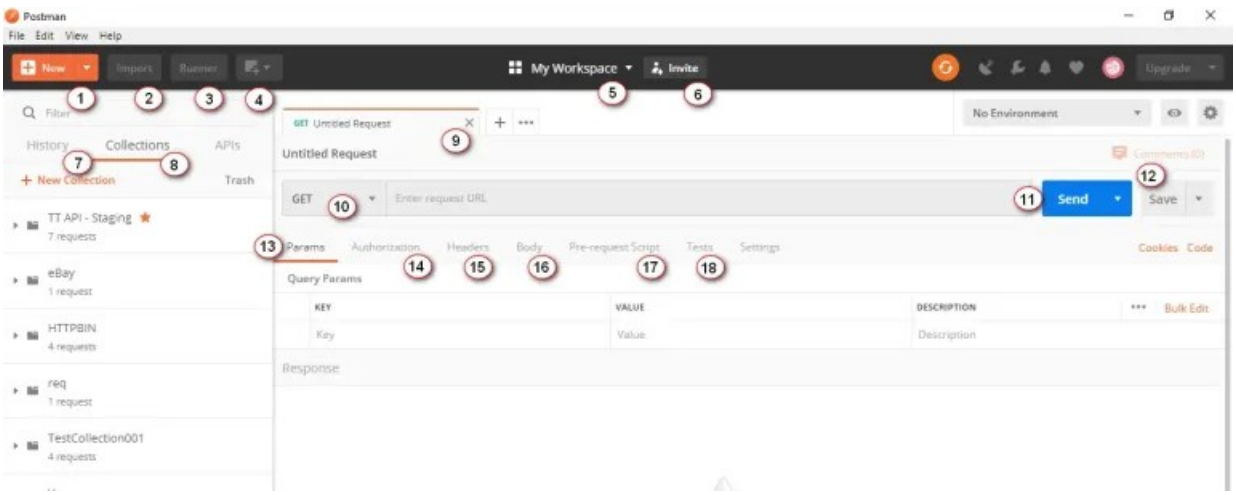
Postman

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.

Интерфейс Postman



New: С помощью этой кнопки можно создать новый запрос (Request), коллекцию (Collection) или окружение (Environment).

Import: С помощью этой кнопки можно импортировать коллекцию или окружение. По нажатию откроется окно, где вы сможете выбрать одну из нескольких опций для импорта: импорт из файла, папки или по ссылке. Также можно просто вставить данные для импорта в текстовое поле.

Runner: По нажатию на кнопку запускается Collection Runner, который выполняет коллекции запросов.

Open New: По нажатию открывается новое окно Postman или новое окно запуска коллекций.

My Workspace: Моя рабочая область. С помощью этой кнопки можно создать новую рабочую область (workspace). Рабочая область предоставляет общий контекст для работы с API. Может использоваться для совместной работы внутри команды (ее можно расшарить с коллегами).

Invite: С помощью этой кнопки можно пригласить других членов команды для совместной работы внутри рабочей области (workspace-а)

History: Все запросы и ответы попадают во вкладку «History» (История). Это позволяет вернуться к предыдущим запросам.

Collections: В этой вкладке хранятся коллекции запросов. Коллекции используются для группировки запросов по каким-либо признакам.

Request Tab: Вкладка запроса. Название вкладки по умолчанию — Untitled Project. Хорошая практика — называть вкладку по названию запроса.

HTTP Request: С помощью этого выпадающего списка можно выбрать тип запроса: GET, POST, PUT, PATCH, DELETE и т.п.

Request URL: URL API запроса.

Save: По нажатию на кнопку Save можно сохранить запрос (или перезаписать, если запрос уже был сохранен ранее)

Params: Параметры, необходимые для выполнения запроса.

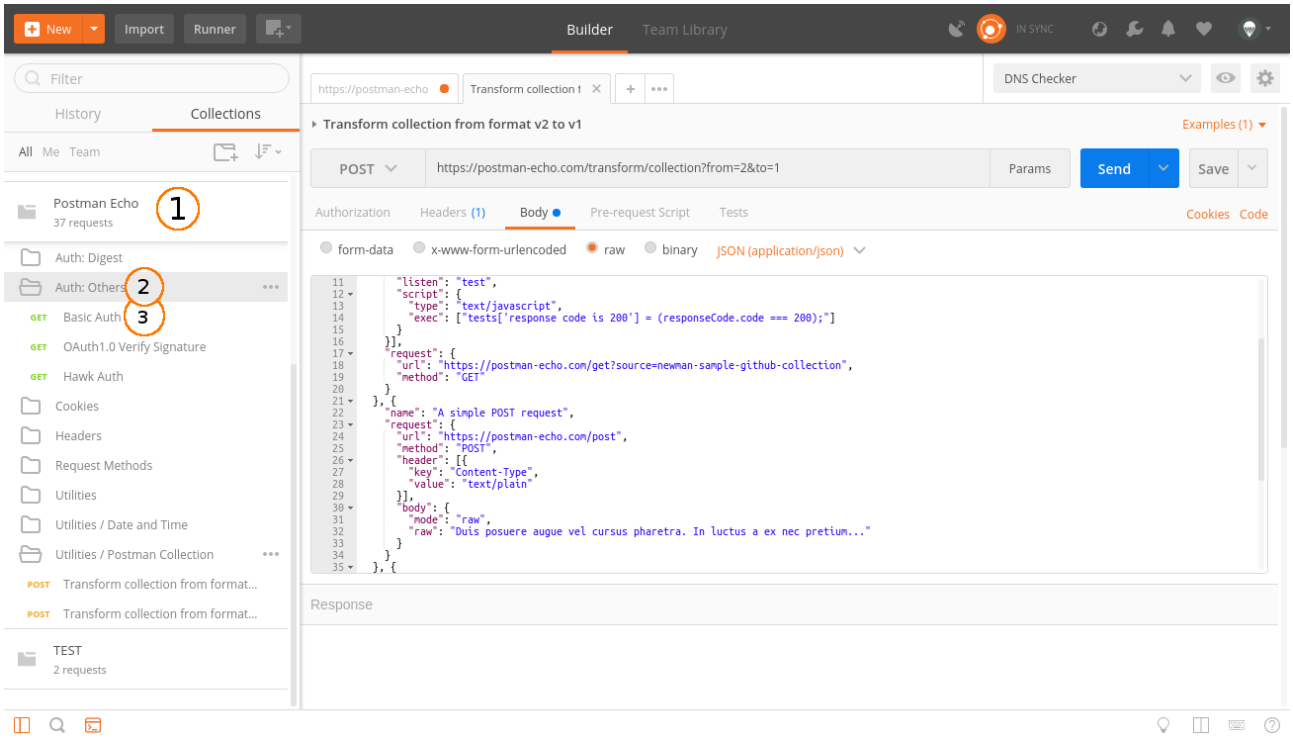
Authorization: API используют авторизацию, чтобы убедиться, что клиент имеет доступ к запрашиваемым данным. В этой секции описываются параметры авторизации: например, username, password, bearer-токен и т.п.

Headers: Для работы с некоторыми API с каждым запросом необходимо отправлять специальные хедеры. Это нужно для того, чтобы добавить дополнительные данные о типе операции, которую вы хотите провести. Хедеры можно указать в этой секции.

Body: В этой вкладке указываются данные, которые должны быть отправлены вместе с запросом.

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)

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.



1 — коллекция, 2 — папка, 3 — запрос

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. Давайте рассмотрим подробнее всё по порядку.

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, которые мы рассмотрим позже.

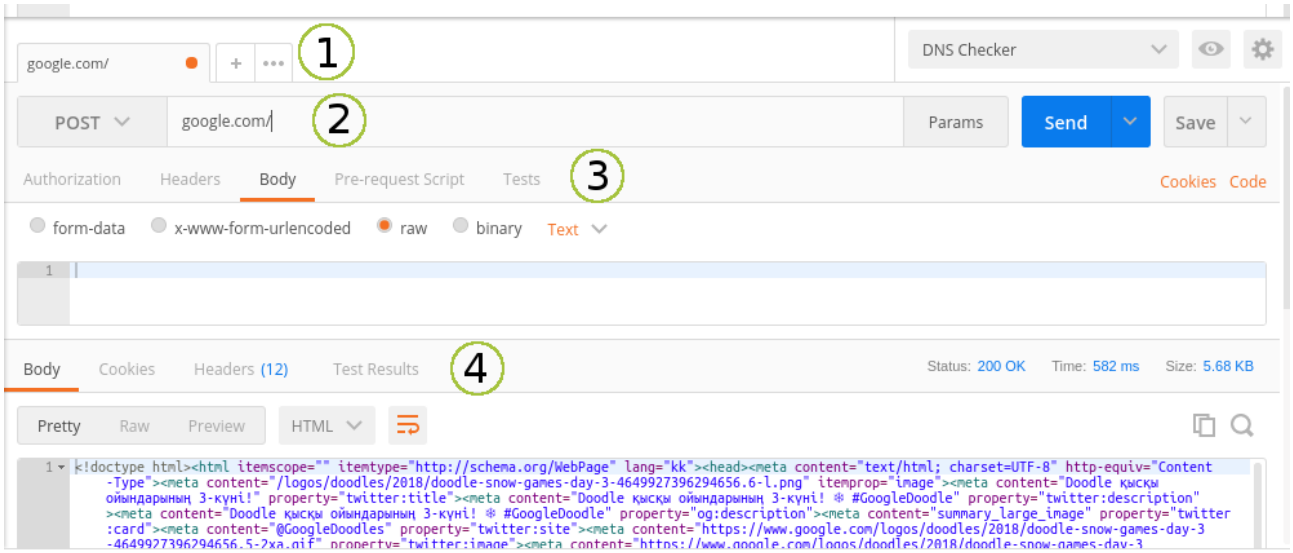
Folder

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.



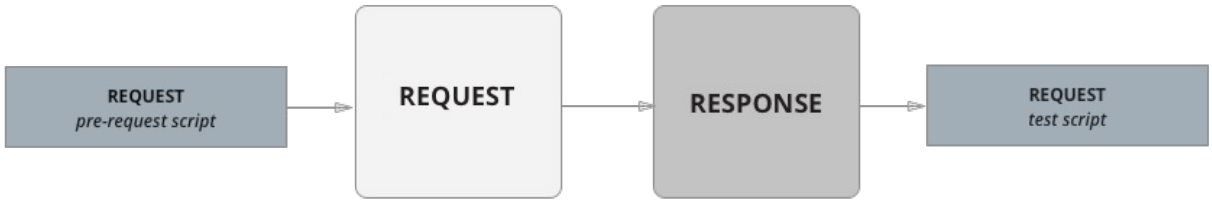
1 — вкладки с запросами, 2 — URL и метод, 3 — параметры запроса, 4 — параметры ответа

URL, к которому мы обращаемся в запросе, называется **Endpoint**. В некотором смысле это конечный объект, с которым мы будем взаимодействовать.

Скрипты

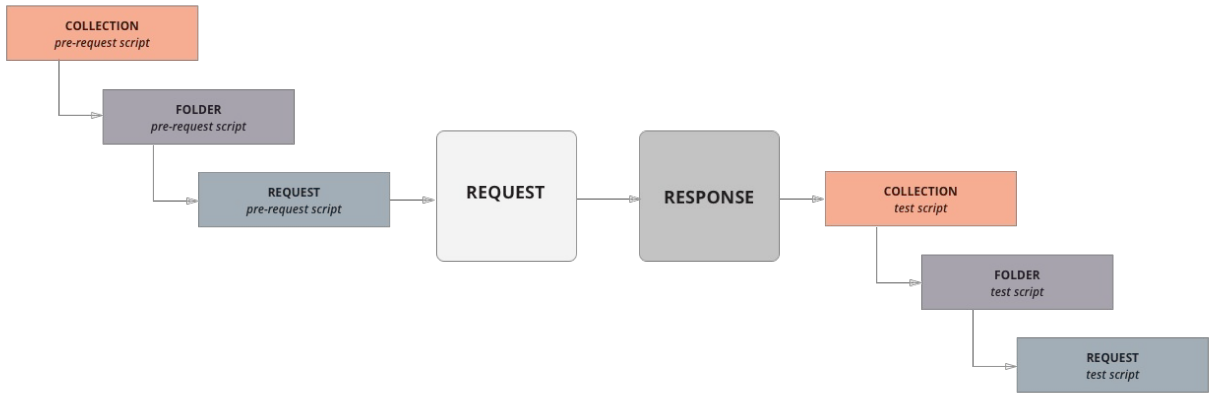
"Postman Sandbox" это среда исполнения JavaScript доступная при написании "Pre-request Script" и "Tests" скриптов. "Pre-request Script" используется для

проведения необходимых операций перед запросом, например, можно сделать запрос к другой системе и использовать результат его выполнения в основном запросе. "Tests" используется для написания тестов, проверки результатов, и при необходимости их сохранения в переменные.



Последовательность выполнения запроса (из оф. документации)

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.



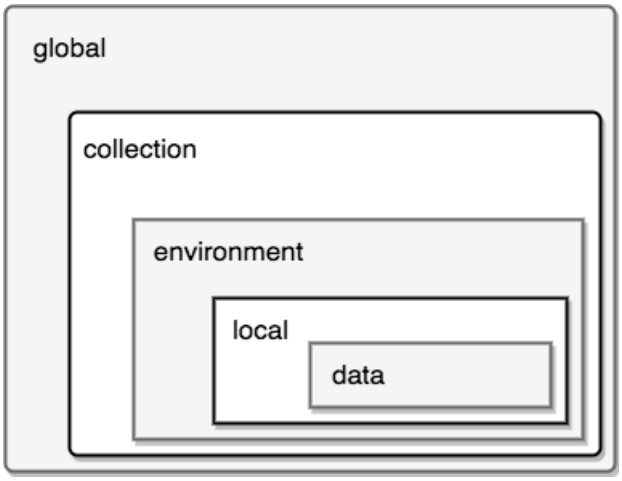
Последовательность выполнения запроса со скриптами папок и коллекций (из оф. документации)

Переменные

Postman имеет несколько пространств и областей видимости для переменных:

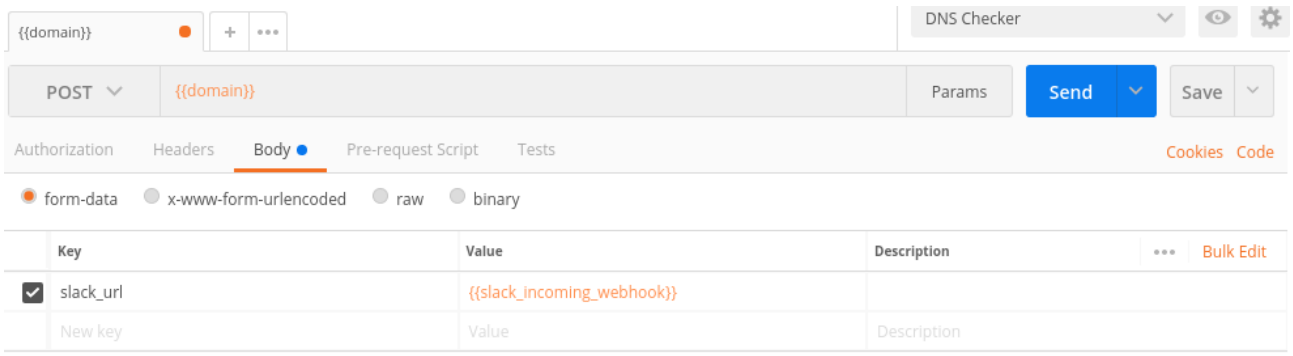
- Глобальные переменные
- Переменные коллекции
- Переменные окружения
- Локальные переменные
- Переменные уровня данных

Глобальные переменные и переменные окружения можно создать, если нажать на шестеренку в правом верхнем углу программы. Они существуют отдельно от коллекций. Переменные уровня коллекции создаются непосредственно при редактировании параметров коллекции, а локальные переменные из выполняемых скриптов. Также существуют переменные уровня данных, но они доступны только из Runner, о котором мы поговорим позже.



Приоритет пространств переменных (из оф. документации)

Особенностью переменных в Postman является то, что вы можете вставлять их в конструкторе запроса, в URL, в POST параметры, в Cookie, всюду, используя фигурные скобки в качестве плейсхолдера для подстановки.



{{domain}} и {{slack_incoming_webhook}} — переменные окружения DNS Checker будут заменены на значения во время выполнения запроса

Из скриптов переменные тоже доступны, но получить их поможет вызов стандартного метода встроенной библиотеки pm:

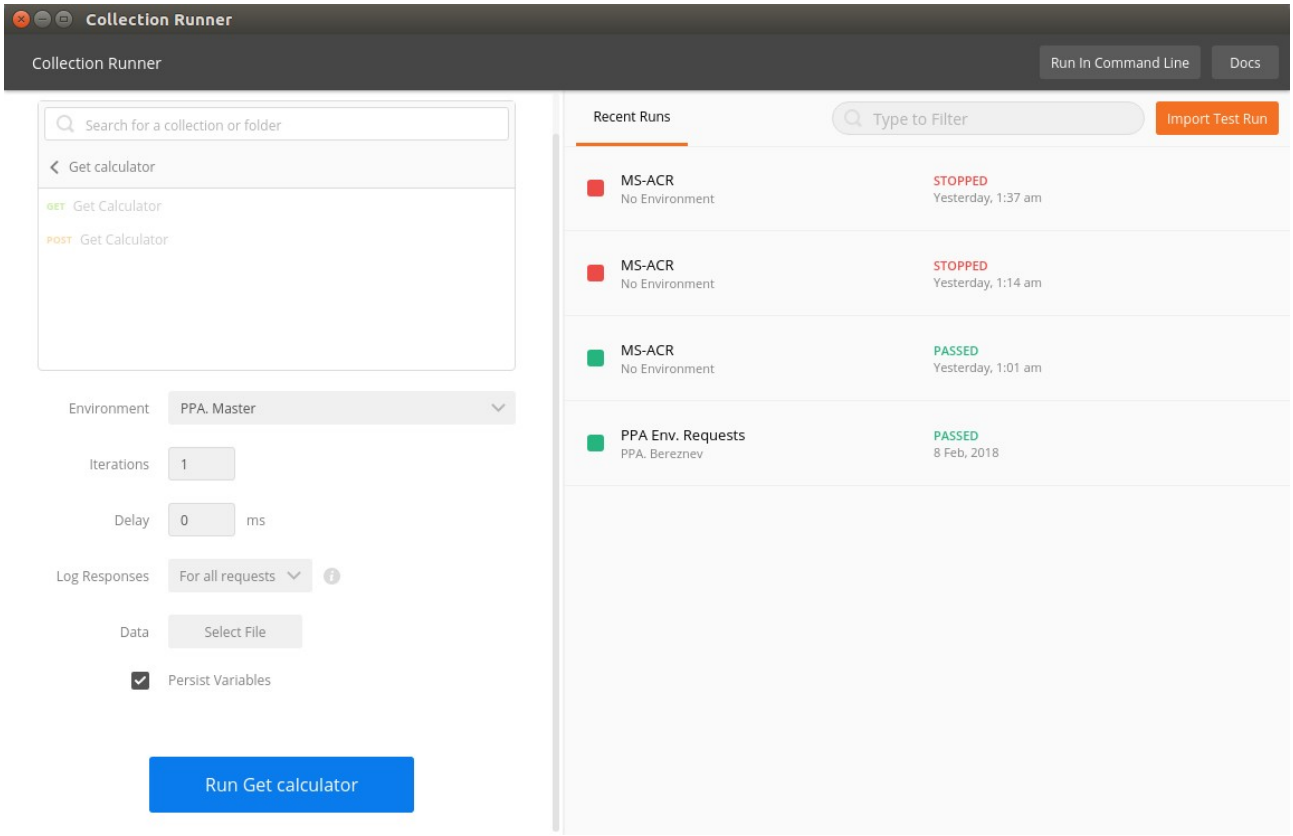
```
// получить глобальную переменную
pm.globals.get("variable_key");
// получить переменную из окружения
pm.environment.get("variable_key");
// получить переменную из любого пространства согласно приоритету
pm.variables.get("variable_key");
```

Collection Runner

Предназначен для тестирования и выполнения всех запросов из коллекции или папки, на ваш выбор. При запуске можно указать количество итераций, сколько раз будет запущена папка или коллекция, окружение, а также дополнительный файл с переменными. Стоит упомянуть, что запросы выполняются последовательно, согласно расположению в коллекции и папках. Порядок выполнения можно изменить используя встроенную команду:

```
// Следующим выполнится запрос с названием "Create order",
postman.setNextRequest('Create order');
```

После выполнения всех запросов формируется отчет, который покажет количество успешных и провальных проверок из скриптов "Tests".



Console

Пользуйтесь консолью для отладки ваших скриптов, и просмотра дополнительной информации по запросам. Консоль работает, как во время запуска одного запроса, так и во время запуска пакета запросов через Runner. Чтобы её открыть, найдите иконку консоли в нижнем левом углу основного экрана приложения.

Заключение

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

- 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)

•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.