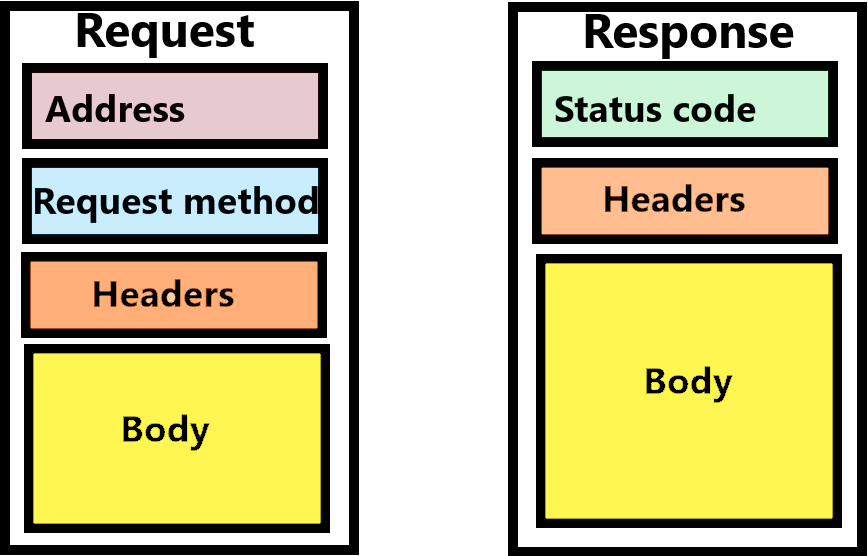
HTTP - к примеру postman отсылает запрос (риквест) на сервер на котором находится запрашиваемая инф., сервер получает, обрабатывает и отсылает ответ (респонс). Для того чтобы такой вид камуникации был доступен используется HTTP. Другими словами **это протокол, (набор правил) которым необходимо следовать и серверу и клиенту** (в нашем случае postman). в каждом риквесте и риспонсе обязательно должны быть указанные на картинке элементы. HTTP протокол (набор правил) в какойто мере можно сравнить с отправкой письма на почте, эта процедура также проводится по определённым правилам (можно сказать по протоколу). Каждое письмо обязательно имеет:

1)- адресс (что соответствует address у HTTP)

2)- почтовые марки (что соответствует Request method)

3)- служебные пометки (Headers)

4)- само письмо (body).

И так же как и с настоящим письмом, если что-то не заполнено письмо не дойдет до получателя, тоже самое и с HTTP.

**From the perspective of how we interact with APIs in Postman, there is no difference between HTTP and HTTPS. HTTP and HTTPS (HTTP Secure) are both protocols for transferring data over the internet. The main difference between the two is that HTTPS uses an encrypted connection to secure the transmission of data, while HTTP does not. When you make a request to an API in Postman, URL specifies the protocol to use (such as HTTP or HTTPS). Regardless of whether you use HTTP or HTTPS in the URL, the way you use Postman to make the request and inspect the response will be essentially the same.**

**HTTP**

Протокол прикладного уровня передачи данных. Говоря о клиент сервеной архитектуре, о запросах и ответах сервера, дак вот эти запросы и ответы это HTTP-запросы и HTTP-ответы.

Благодаря протоколу HTTP Ваш браузер, получая данные с сервера, знает, как их требуется обработать, и успешно обрабатывает их, показывая Вам запрашиваемую страничку.

**Из чего состоят HTTP-запросы:**

1. Стартовая строка (request line).

а) Методы HTTP протокола .

get- получить данные из сервера

you can call get method multiple time with no side effect, (не несёт никаких изменений в API)

doesn't have request body

parameters in headers and URL

put- внесение изменений в существующую сущность,

post- создание какой то новой сущности,

each time you call post new data will be created (это влияет на API)

has a request body

parameters in body, headers, url

delete- удаление существующей сущности…)

б)Цель / путь запроса (как правило URL - Uniform Recorse Locator) пример пути запроса:



Resource - это например товары в интернет магазине или видео на YouTube.

Path parameter - это детализация нашего resource

Query Parameters - задают необходимую фильтрацию, указываются (перечисляются) после вопросительного знака

в) версия HTTP

1. Заголовки (Headers) умное определение это метаданные. Другимим словами это информация о информации т.е. для корректного ответа нам надо сказать в каком формате мы ожидаем информацию(content type) + идёт доп инф. про клиента, т.е. те же кукис и пр.. т.е. чисто техническая инф между клиентом и сервером. Hedears Бывают:

| a)- General Headers | b)- Request headers, | c)- Response headers. |
| --- | --- | --- |
|  |  |  |

1. Тело (Body), обязательно присутствует когда мы используем методы REQUST-а put и post потому, что так как данные методы Request-а что-то создают или меняют(обновляют) нам же надо где то эту информацию “принести” на сервер чтоб он понял не только что мы от него хотим чтоб он сделал но и дали ему материал для этой работы. Нельзя же сказать создай нового пользователя и на этом точка. Надо дать ему имя пользователя, пол, возраст и т. д., т.е. данные. Для этого и существует body в request-е. И в ответе к стати тоже, я думаю.

**Из чего состоят HTTP- RESPONSE:**

1. Строка статуса (Status line)
2. Версия протокола (обычно HTTP /1.1)
3. Код статуса (Status code)

\*1XX - информационные

\*2ХХ - Успешная операция. In HTTP, a status code of 200 OK indicates that the request was successful and the server was able to fulfill the request. This is the most common status code that is returned in response to a successful request.

\*3ХХ - Перенапрвление

\*4ХХ - Клиентские ошибки

\*5ХХ - Серверные ошибки

с) Пояснения (Status text)

2) Заголовки (Headears)

3) Тело (Body)

**ТИПЫ ДАННЫХ:**

Имеется ввиду типы данных которые могут передаваться по HTTP: XML; JSON (javascript object notation)

