**Протокол передачи гипертекста (Hypertext Transfer Protocol - HTTP) -** это прикладной протокол для передачи гипертекстовых документов, таких как HTML. В настоящий момент используется для передачи произвольных данных. Основан на TCP/IP (TLS – защищенный TCP), использует его для пересылки своих сообщений. Может использоваться для передачи содержимого серврам (например с помощью HTML-форм), а также для получения частей документа, с целью обновления веб-страницы по запросу (AJAX).

Гипертекст - термин, обозначающий систему из текстовых страниц, имеющих перекрёстные ссылки. (Текст с навигацией)

Протокол следует **клиент-серверной** модели. Основынми реализациями клиентов являются браузеры. Основными реализациями серверов – Apache, nginx, Google Web Server.

Клиенты и серверы взаимодействуют обмениваясь одиночными сообщениями, а не потоком данных. Сообщения, отправленные клиентом, называются **запросами**, а сообщения, отправленные сервером – **ответами**.

Для взаимодействия с сервером клиент выполняет следующие шаги:

1. Открытие TCP соединения, которое будет использоваться для отправки запроса и получения ответа. Клиент может открыть новое соединение, переиспользовать существующее или открыть несколько TCP-соединений.
2. Отправка запроса в виде HTTP-сообщения.
3. Чтение ответа от сервера.
4. Закрывает или переиспользует соединение дальнейших запросов.

Сервер не обязательно расположен на одной машине, и наоборот - несколько серверов могут быть расположены (поститься) на одной и той же машине. В соответствии с версией HTTP/1.1 и имея [Host](https://developer.mozilla.org/ru/docs/Web/HTTP/Headers/Host) заголовок, они даже могут делить тот же самый IP-адрес.

HTTP – это протокол **без сохранения состояния.**Не существует связи между двумя запросами, которые последовательно выполняются по одному соединению. Сервер не сохраняет никаких данных о промежуточном состоянии между парами “запрос ответ”.

HTTP **расширяемый.**  Заголовки (headers) сделали этот протокол легким для расширения. Новая функциональность может быть даже введена простым соглашением между клиентом и сервером о семантике нового заголовка.

**HTTP-сообщения**

Каждое HTTP-сообщение состоит из следующих частей:

1. Стартовая строка, определяющая тип сообщения
2. HTTP заголовки
3. Пустая строка, отделяющая заголовки от тела
4. Тело сообщения – данные запроса (ответа)

Заголовки и тело могут отсутствовать, стартовая строка является обязательным элементом.

**Стартовая строка запроса** состоит из следующих элементов

1. Метод HTTP, указывающий какое действие выполнить для данного ресурса.
2. URI – унифицированный идентификатор ресурса
3. Версия HTTP.

**Стартовая строка ответа (строка статуса)** содержит следующую информацию:

1. Версию протокола, обычно HTTP/1.1
2. Код состояния (status code), сообщающий об успешности запроса или причине неудачи.
3. Пояснение (status text). Краткое текстовое описание кода состояния.

HTTP/1.1 404 Not Found.

Заголовки HTTP – пара имя – значение. Позволяют клиенту и серверу отправлять дополнительную информацию с HTTP запросом или ответом.

Они могут содержать описание данных и информацию, необходимую для взаимодействия между клиентом и сервером. Имя заголовка не чувствительно к регистру, пробелы перед значением игнорируются.

**Виды заголовков**

***Основные заголовки (General headers*)** – должны включаться в любое сообщение клиента и сервера.

**Cache-Control** используется для задания инструкций кеширования как для запросов, так и для ответов.

Заголовок **Connection** определяет, остаётся ли сетевое соединение активным после завершения текущей транзакции (запроса).

keep-alive – не закрывать TCP соединение.

**Close –** указывает, что клиент или сервер не хотели бы закрывать соединение.

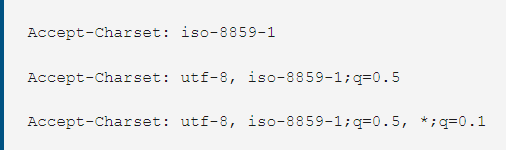
**Date –** основной HTTP заголовок содержащий дату и время, в которое сообщение было создано.

***Заголовки запроса (Request headers*)** – используются в запросах клиента.

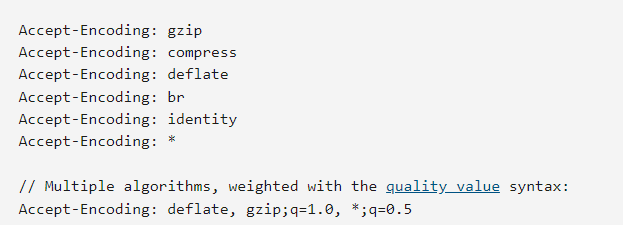
HTTP заголовок запроса **Accept** указывает, какие типы контента, выраженные как MIME типы, клиент может понять.



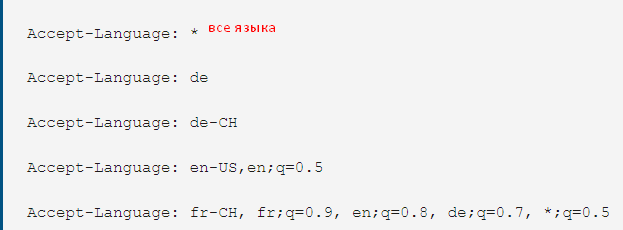
Заголовок **Accept-Charset** запроса HTTP сообщает какую кодировку клиент может понять.



HTTP-заголовок запроса **Accept-Encoding** указывает кодировку содержимого (обычно алгоритм сжатия), которую может понять клиент.

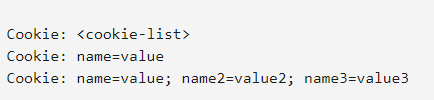


**Accept-Language** сообщает серверу, какие языки клиент понимает и какая локаль предпочтительнее



заголовок **Cookie** содержит сохраненные HTTP-куки, связанные с сервером (т.е. ранее отправленные сервером с помощью заголовка Set-Cookie или установленные в Javascript с помощью Document.cookie).

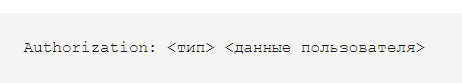
Заголовок Cookie является необязательным и может быть опущен, если, например, настройки конфиденциальности браузера блокируют cookies.



Заголовок **Host** содержит имя домена, для которого предназначен запрос и, опционально, номер порта.

HTTP-запрос отправляется на определенные IP-адреса. Но так как большинство серверов способны размещать несколько сайтов под одним IP, они должны знать, какое доменное имя ищет браузер.

Заголовок HTTP запроса **Authorization** включает в себя данные пользователя для проверки подлинности пользовательского агента.



**<данные пользователя>**

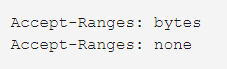
Если используется схема авторизации «Базовая», данные пользователя формируются следующим образом:

* Логин и пароль, разделённые двоеточием (aladdin:opensesame).
* Результирующая строка, закодированная в [base64](https://developer.mozilla.org/en-US/docs/Glossary/Base64) (YWxhZGRpbjpvcGVuc2VzYW1l).

**User-Agent**

***Заголовки ответа (Response headers*)** – используются в ответах серверов.

Заголовок ответа **Accept-Ranges** -- это маркер, который использует сервер, чтобы уведомить клиента о поддержке "запросов по кускам".



Заголовок **Allow** перечисляет набор методов, поддерживаемых ресурсом.

Этот заголовок должен быть отправлен, если сервер отвечает кодом состояния 405 Method Not Allowed, чтобы указать, какие методы запроса могут быть использованы.



**Last-Modified –** дата и время последнего изменения ресурса.

Заголовок ответа **Location** указывает URL, на который следует перенаправить страницу. Он имеет значение только тогда, когда обслуживается с ответом статуса 3xx (перенаправление) или 201 (создано).

В случае создания ресурса он указывает URL-адрес вновь созданного ресурса.

Если код 303, то для перенаправления совершается еще один запрос, используя при этом метод GET.

При других кодах состояния, обычное используется тот же метод, что и при первом запросе.

**Set-Cookie** Когда веб-сервер хочет установить или обновить файл cookie в вашем браузере, он будет использовать этот header.

Каждый файл cookie отправляется как отдельный header. Обратите внимание, что файлы cookie, установленные с помощью JavaScript, не проходят через HTTP headers.

Если дата истечения срока действия не указана, cookie удаляется, когда окно браузера закрыто.

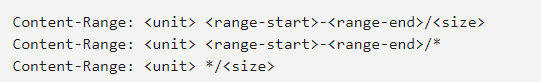
***Заголовки сущности (Entity Headers)***содержут информацию, относящуюся к телу сообщения.

**Content-Encoding –** заголовок сущности, указывающий кодировки примененные для сжатия тела сообщения.

**Content-Language –** указывает язык контента (тела, отправленного получателю).

Заголовок **Content-Length** указывает размер отправленного получателю тела объекта в байтах.

HTTP-заголовок **Content-Range** ответа указывает, к какому месту в полном тексте сообщения относится частичное сообщение (когда получаем ресурс по частям).

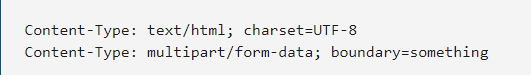




**Content-Type** используется для указания типа ресурса.

В ответах заголовок Content-Type предоставляет клиенту фактический тип содержимого возвращаемого содержимого.

В запросах (таких как POST или PUT) клиент сообщает серверу, какой тип данных будет отправлен.



**Тело**

Тело содержит ресурс, отправляемый на сервер, либо получаемый с сервера.

Включается тело в сообщение или нет зависит как от меода запроса, так и от кода состояния ответа.

**Разница между версиями HTTP**

HTTP/1.0 открывал TCP-соединение для каждого обмена запросом/ответом.