
Описание протокола HTTP

[На главную страницу](#) | [Интернет](#) | [Хостинг](#) | [Сервисы](#)

HTTP (HyperText Transfer Protocol - протокол передачи гипертекста) был разработан как основа World Wide Web.

Работа по протоколу HTTP происходит следующим образом: программа-клиент устанавливает TCP-соединение с сервером (стандартный номер порта-80) и выдает ему HTTP-запрос. Сервер обрабатывает этот запрос и выдает HTTP-ответ клиенту.

Структура HTTP-запроса

HTTP-запрос состоит из заголовка запроса и тела запроса, разделенных пустой строкой. Тело запроса может отсутствовать.

Заголовок запроса состоит из главной (первой) строки запроса и последующих строк, уточняющих запрос в главной строке. Последующие строки также могут отсутствовать.

Запрос в главной строке состоит из трех частей, разделенных пробелами:

Метод (иначе говоря, команда HTTP):

GET - запрос документа. Наиболее часто употребляемый метод; в HTTP/0.9, говорят, он был единственным.

HEAD - запрос заголовка документа. Отличается от GET тем, что выдается только заголовок запроса с информацией о документе. Сам документ не выдается.

POST - этот метод применяется для передачи данных CGI-скриптам. Сами данные следуют в последующих строках запроса в виде параметров.

PUT - разместить документ на сервере. Насколько я знаю, используется редко. Запрос с этим методом имеет тело, в котором передается сам документ.

Ресурс - это путь к определенному файлу на сервере, который клиент хочет получить (или разместить - для метода PUT). Если ресурс - просто какой-либо файл для считывания, сервер должен по этому запросу выдать его в теле ответа. Если же это путь к какому-либо CGI-скрипту, то сервер запускает скрипт и возвращает результат его выполнения. Кстати, благодаря такой унификации ресурсов для клиента практически безразлично, что он представляет собой на сервере.

Версия протокола-версия протокола HTTP, с которой работает клиентская программа.

Таким образом, простейший HTTP-запрос может выглядеть следующим образом:

GET / HTTP/1.0

Этот запрос запрашивает версию файла по номеру файла, указанному web-сервером

здесь запрашивается корневой файл из корневой директории web-сервера.

Строки после главной строки запроса имеют следующий формат:

Параметр: значение.

Таким образом задаются параметры запроса. Это является необязательным, все строки после главной строки запроса могут отсутствовать; в этом случае сервер принимает их значение по умолчанию или по результатам предыдущего запроса (при работе в режиме Keep-Alive).

Перечислю некоторые наиболее употребительные параметры HTTP-запроса:

Connection (соединение)- может принимать значения Keep-Alive и close. Keep-Alive ("оставить в живых") означает, что после выдачи данного документа соединение с сервером не разрывается, и можно выдавать еще запросы. Большинство браузеров работают именно в режиме Keep-Alive, так как он позволяет за одно соединение с сервером "скачать" html-страницу и рисунки к ней. Будучи однажды установленным, режим Keep-Alive сохраняется до первой ошибки или до явного указания в очередном запросе Connection: close.

close ("закреть") - соединение закрывается после ответа на данный запрос.

User-Agent - значением является "кодовое обозначение" браузера, например:

Mozilla/4.0 (compatible; MSIE 5.0; Windows 95; DigExt)

Accept - список поддерживаемых браузером типов содержимого в порядке их предпочтения данным браузером, например для моего IE5:

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */*

Это, очевидно, нужно для случая, когда сервер может выдавать один и тот же документ в разных форматах.

Значение этого параметра используется в основном CGI-скриптами для формирования ответа, адаптированного для данного браузера.

Referer - URL, с которого перешли на этот ресурс.

Host - имя хоста, с которого запрашивается ресурс. Полезно, если на сервере имеется несколько виртуальных серверов под одним IP-адресом. В этом случае имя виртуального сервера определяется по этому полю.

Accept-Language - поддерживаемый язык. Имеет значение для сервера, который может выдавать один и тот же документ в разных языковых версиях.

Формат HTTP-ответа

Формат ответа очень похож на формат запроса: он также имеет заголовок и тело, разделенное пустой строкой.

Заголовок также состоит из основной строки и строк параметров, но формат основной строки отличается от таковой в заголовке запроса.

Основная строка запроса состоит из 3-х полей, разделенных пробелами:

Версия протокола - аналогичен соответствующему параметру запроса.

Код ошибки - кодовое обозначение "успешности" выполнения запроса. Код 200 означает "все нормально" (OK).

Словесное описание ошибки - "расшифровка" предыдущего кода. Например для 200 это OK, для 500 - Internal Server Error.

Наиболее употребительные параметры http-ответа:

Connection - аналогичен соответствующему параметру запроса.

Если сервер не поддерживает Keep-Alive (есть и такие), то значение Connection в ответе всегда close.

Поэтому, на мой взгляд, правильной тактикой браузера является следующая:

- 1. выдать в запросе Connection: Keep-Alive;*
- 2. о состоянии соединения судить по полю Connection в ответе.*

Content-Type ("тип содержимого") - содержит обозначение типа содержимого ответа.

В зависимости от значения Content-Type браузер воспринимает ответ как HTML-страницу, картинку gif или jpeg, как файл, который надо сохранить на диске, или как что-либо еще и предпринимает соответствующие действия. Значение Content-Type для браузера аналогично значению расширения файла для Windows.

Некоторые типы содержимого:

text/html - текст в формате HTML (веб-страница);

text/plain - простой текст (аналогичен "блокнотовскому");

image/jpeg - картинка в формате JPEG;

image/gif - то же, в формате GIF;

application/octet-stream - поток "октетов" (т.е. просто байт) для записи на диск.

На самом деле типов содержимого гораздо больше.

Content-Length ("длина содержимого") - длина содержимого ответа в байтах.

Last-Modified ("Модифицирован в последний раз") - дата последнего изменения документа.

(С) Андрей Черный

angel07@inbox.ru

[На главную страницу](#) | [Интернет](#) | [Хостинг](#) | [Сервисы](#)

Readme.txt

