

Протокол HTTP

Веб-программирование

Назначение

HTTP (HyperText Transfer Protocol) - «протокол передачи гипертекста»)

прикладной протокол стека TCP/IP;

→ символьный, клиент-серверный

основной протокол WWW;

«транспорт» для других прикладных протоколов (SOAP, XML-RPC, WebDAV ...);

→ HTTP/1.0 - RFC 1945, **HTTP/1.1 - RFC 2616**

Участники обмена

Веб-серверы — поставщики услуг хранения и обработки информации

Apache, IIS, nginx ...

Клиенты — конечные потребители услуг сервера

Браузеры: Mozilla, IE, Opera, Safari, Chrome ...

Роботы: пауки, кроулеры, дятлы & so on...

Прокси-серверы — транспортные посредники
(кэширование данных)

Squid, Naviscope, Kerio, UserGate ...



Процедура обмена



in nuce:

1. Клиент запрашивает ресурс (по URI)
2. Прокси-сервер проверяет запрос на наличие ресурса в кэше (hit || miss)
3. Веб-сервер возвращает запрошенный ресурс

Структура протокола

Стартовая строка

Методы

Заголовки

Коды состояния

Тело сообщения

No.	Source	Destination	Protocol	Info
31	192.168.3.1	194.188.210.1	HTTP	HTTP/1.0 302 Moved Temporarily
37	192.168.3.1	194.188.210.1	HTTP	HTTP/1.0 200 OK (text/html)
54	192.168.3.1	194.188.210.1	HTTP	HTTP/1.0 200 OK (text/css)

HTTP/1.0 200 OK\r\n	Server: Apache/2.2.3 (CentOS)\r\n	Last-Modified: Wed, 09 Feb 2011 17:13:15 GMT\r\n	Content-Type: text/html; charset=UTF-8\r\n	Accept-Ranges: bytes\r\n	Date: Thu, 03 Mar 2011 04:04:36 GMT\r\n	Content-Length: 2945\r\n	Age: 13165\r\n	X-Cache: HIT from proxy.omgtu\r\n	via: 1.0 proxy.omgtu (squid/3.1.8)\r\n	Connection: keep-alive\r\n	\r\n
Line-based text data: text/html											
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/T											
<html xmlns="http://www.w3.org/1999/xhtml">\n											
<head>\n											
\t<title>IANA — Example domains</title>\n											
\t<!-- start common-head -->\n											
\t<meta http-equiv="Content-type" content="text/html; charset=utf-8" />\n											
\t<link rel="stylesheet" type="text/css" href="/_css/reset-fonts-grids.css" />\n											
\t<link rel="stylesheet" type="text/css" media="screen" href="/_css/screen.css" />\n											
\t<link rel="stylesheet" type="text/css" media="print" href="/_css/print.css" />\n											
\t<link rel="shortcut icon" type="image/ico" href="/favicon.ico" />\n											

0000	48 54 54 50 2f 31 2e 30 20 32 30 30 20 4f 4b 0d	HTTP/1.0 200 OK.
0010	0a 53 65 72 76 65 72 3a 20 41 70 61 63 68 65 2f	Server: Apache/

Стартовая строка

Запрос клиента

Метод Ресурс Протокол/Версия
GET /index.html HTTP/1.1

Ответ сервера

Протокол/Версия КодСостояния Пояснение
HTTP/1.0 200 ОК

Методы протокола

GET - запрос содержимого

GET /path/resource?param1=value1 HTTP/1.1

Данные передаются в адресной строке

Условные GET содержат заголовки If-Modified-Since, If-Match и подобные (кэширование)

Частичные GET содержат в запросе Range (докачка, многопоточное скачивание)

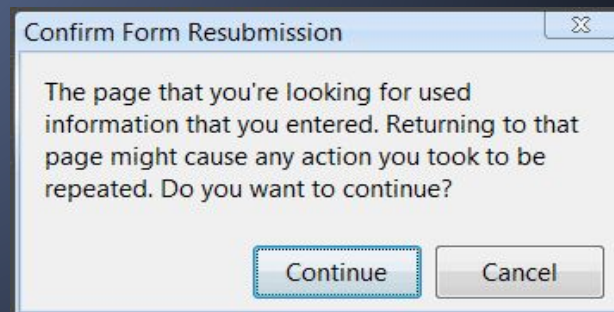
Еще методы...

POST - применяется для передачи пользовательских данных (из веб-форм) и загрузки файлов на сервер.

Данные передаются в *теле запроса*

Повторные запросы могут изменять возвращаемые результаты

Ответы сервера не кэшируются



Опять методы...

OPTIONS - определение возможностей веб-сервера и/или параметров соединения

HEAD - применяется для извлечения метаданных и валидации ресурса (link check)

TRACE - Возвращает полученный запрос так, что клиент может увидеть, что прокси-сервера добавляют или изменяют в запросе



... а АйТишники дятлами называют специальные программы-линкчекеры...

Снова методы...

PUT - Применяется для загрузки содержимого запроса по указанному адресу.

PATCH - Аналогично PUT, но применяется только к фрагменту ресурса.

DELETE - Удаляет указанный ресурс.

LINK - Устанавливает связь указанного ресурса с другими.

CONNECT - используется для туннелирования трафика

Заголовки

General Headers — должны включаться в любое сообщение клиента и сервера

Request Headers — только в запросах клиента

Response Headers — только в ответах сервера

Entity Headers — сопровождают каждую сущность сообщения, применяются при передаче *множественного содержимого*

Пример обмена

Запрос:

```
GET /index.html HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; U; Linux
i686; ru; rv:1.9b5) Gecko/2008050509
Firefox/3.0b5
Accept: text/html
Connection: close
```

Ответ:

```
HTTP/1.0 200 OK
Date: Wed, 11 Feb 2009 11:20:59 GMT
Server: Apache
X-Powered-By: PHP/5.2.4-2ubuntu5wm1
Last-Modified: Wed, 11 Feb 2009 11:20:59
GMT
Content-Language: ru
Content-Type: text/html; charset=utf-8
Content-Length: 1234
Connection: close
тут пустая строка (обязательно!)
<html><head>...
```

Коды состояния

1xx Informational — Состояние процесса передачи

2xx Success — Успешное выполнение

3xx Redirection — Перенаправление

4xx Client Error — Ошибки клиента

5xx Server Error — Ошибки сервера



Тело сообщения

Непосредственно запрошенные данные

гипертекст, двоичные данные

Тип определяется значением заголовка

Content-Type

`Content-Type: text/html;`

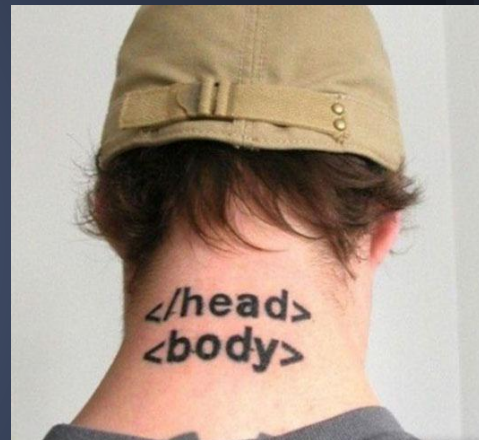
MIME types

Multipurpose Internet Mail Extensions

`text/html, text/plain, multipart/mixed, text/xml, application/xml...`

(подробно:

`http://www.iana.org/assignments/media-types/media-types.xhtml)`



Pro et Contra

Достоинства - простота, расширяемость, распространённость

Недостатки - большой размер сообщений, отсутствие «навигации» (клиент не может явным образом запросить список доступных файлов, как в протоколе FTP), нет поддержки распределённых вычислений

В 1998 году W3C предложил альтернативный протокол HTTP-NG: до сих пор на стадии разработки.

more: RFC 2616, RTFM