

HTTP – Hyper Text Transfer Protocol – RFC 2616

HTTP Methods

Get:

- requests resource whose URL is given as an argument.
- Example www.somesite.com/animalsearch?monkeys&bananas.
- resource could be a program or data.
- Resources that are implemented as data are supplied as MIME type i.e. Multipurpose Internet Mail Extensions. Data is prefixed with its MIME type so that browser knows how to handle it. A MIME type specifies a type & sub-type. For example text/plain, text/html, image/gif, image/jpeg etc. Clients can also specify the MIME type that they are willing to accept.
- If URL is a program, the web server runs it and returns the results to the client.
- Arguments may be added to the URL i.e. contents of a form could be sent as parameters to a CGI program using “?” and “&” operators etc.
- GET operation can be made conditional upon the date the resource was modified last.
- If GET is a request for data then there is no request body.

Post:

- specifies the URL of a resource that can deal with the data supplied by the request.
- The entries in the form are added to the message body
- This method is designed to deal with
 - Providing a block of data to a server or a CGI program
 - Posting a message to a bulletin board mailing list or newsgroup
 - Extending a database with an append operation.

Head:

- Identical to GET, but does not return any data, just all the information about data.

Put:

- Server stores the supplied data at the specified URL.

Delete:

- Server deletes the resource identified by the given URL.
- This may not be always allowed & thus will result in a failure.

Options:

- The server supplies the client with a list of methods it allows to be applied to the given URL e.g. GET, HEAD, PUT etc & its special requirements.

TRACE:

- The server sends back the request message.
- It is used for diagnostics purposes.

The requests described above may be intercepted by a proxy server. The responses to GET and HEAD may be cached by proxy services.

HTTP Messages:

- Status code is 3 digit integer; reason is a textured phrase providing report on the success or failure of Request.
- Header fields contain request modifiers & client information such as conditions on the latest date of modification of resource or acceptable content type.
- Message body contains the data.
- Message body has its own headers specifying information about data such as its length, MIME type, content encoding and last date it was modified

HTTP 1.0 only supports GET, POST & HEAD. HTTP 1.1 allows for additional methods including PUT & DELETE.

HTTP 1.1 uses persistent connections i.e. multiple request responses possible with one TCP connection. In HTTP 1.0 each request response is a separate connection. If HTTP 1.1 server receives http 1.0 request then the connection will be closed.

HTTP Request Message Format

method	sp	URL	sp	version	cr lf
Header Field Name	sp	value	cr lf		

~

Header Field Name	sp	value	cr lf
cr lf			

entire body	
-------------	--

~

Example:

GET / somedir/page.html HTTP/1.1

HOST: www.sameschool.edu

Connection: close

User-agent: Mozilla/4.0

Accept Language: fr

HTTP Response Message Format

version	sp	Status Code	sp	Phrase	cr lf
Header Field Name	sp	value	cr lf		

~

Header Field Name	sp	value	cr lf
cr lf			

entire body	
-------------	--

~

Example:
HTTP/1.1 200 ok
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/ 1.3.0 (unix)
Last Modified: Mon, 22 Jun 1998 09:23:29 GMT
Content-length: 6821
Content -Type: text/html
(data data...)

Status Examples:

200 OK
301 moved permanently (new URL in identified in Location Header)
400 bad request
404 not found
505 http version not supported

try this

telnet www.somewebstide.com 80
GET/~dir/inder.thml HTTP/1.0

Authorization:

- Server specifies authorization extension in WWW-Authenticate-Header.
- Client responds by putting the required information in the "Authorization:" header line.

Cookies:

- Used to keep track of user's activity.
- Browser maintains a cookie file at the client.
- Server maintains a database that stores the user's activity.
- Server assigns a cookie number to the client and informs it to the client in "Set-Cookie" Header. For example "Set-Cookie: 1678453".
- Client browser copies this number to the cookie file and for all future correspondence to this server, specifies the cookie number using "cookie: 1678453"
- Server thus can keep track of which pages the user has visited. If the user revisits the same web site after some period. The browser then adds the same cookie. The server then uses the cookie number to identify this user's profile in it's database

The Conditional Get:

Request

GET /fruit/kiw.gif HTTP/1.0
User_Agent: Mozilla/4.0
If Modified Since: Mon, 22 Jun 1998 09:23:24

Response

HTTP/1.0 304 not modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/ 1.3.0 (unix)