List of HTTP header fields

HTTP header fields are components of the header section of request and response messages in the Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction.

1 General format

The header fields are transmitted after the request or response line, which is the first line of a message. Header fields are colon-separated name-value pairs in clear-text string format, terminated by a carriage return (CR) and line feed (LF) character sequence. The end of the header section is indicated by an empty field, resulting in the transmission of two consecutive CR-LF pairs. Historically, long lines could be folded into multiple lines; continuation lines are indicated by the presence of a space (SP) or horizontal tab (HT) as the first character on the next line. This folding is now deprecated.[1]

2 Field names

A core set of fields is standardized by the Internet Engineering Task Force (IETF) in RFC 7231. An official registry of these fields as well as those of supplementary specifications is maintained by the IANA. Additional field names and permissible values may be defined by each application.

The permanent registry of header fields and repository of provisional registrations are maintained by the IANA.

Non-standard header fields were conventionally marked by prefixing the field name with X- .[2] However, this convention became deprecated in June 2012 due to the inconveniences it caused when non-standard fields became standard.[3]

A prior restriction on use of Downgraded- has also since been lifted.[4]

3 Field values

A few fields can contain comments (i.e. in User-Agent, Server, Via fields), which can be ignored by software.[5]

Many field values may contain a quality (q) key-value pair, specifying a weight to use in content negotiation.[6]

4 Size limits

The standard imposes no limits to the size of each header field name or value, or to the number of fields. However, most servers, clients, and proxy software impose some limits for practical and security reasons. For example, the Apache 2.3 server by default limits the size of each field to 8190 bytes, and there can be at most 100 header fields in a single request.[7]

5 Request fields

5.1 Common non-standard request fields

6 Response fields

6.1 Common non-standard response fields

7 Effects of selected fields

7.1 Avoiding caching

If a web server responds with Cache-Control: no-cache then a web browser or other caching system (intermediate proxies) must not use the response to satisfy subsequent responses without first checking with the originating server (this process is called validation). This header field is part of HTTP version 1.1, and is ignored by some caches and browsers. It may be simulated by setting the Expires HTTP version 1.0 header field value to a time earlier than the response time. Notice that no-cache is not instructing the browser or proxies about whether or not to cache the content. It just tells the browser and proxies to validate the cache content with the server before using it (this is done by using if-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match attributes mentioned above). Sending a no-cache value thus instructs a browser or proxy to not use the cache contents merely based on “freshness criteria” of the cache content. Another common way to prevent old content from being shown to the user without validation is Cache-Control: max-age=0. This instructs the user agent that the content is stale and should be validated before use.

The header field Cache-Control: no-store is intended to instruct a browser application to make a best effort not to write it to disk (i.e not to cache it).

The request that a resource should not be cached is no guarantee that it will not be written to disk. In particular, the HTTP/1.1 definition draws a distinction between history stores and caches. If the user navigates back to a previous page a browser may still show you a page that has been stored on disk in the history store. This is correct behavior according to the specification. Many user agents show different behavior in loading pages from the history store or cache depending on whether the protocol is HTTP or HTTPS.

The Cache-Control: no-cache HTTP/1.1 header field is also intended for use in requests made by the client. It is a means for the browser to tell the server and any intermediate caches that it wants a fresh version of the resource. The Pragma: no-cache header field, defined in the HTTP/1.0 spec, has the same purpose. It, however, is only defined for the request header. Its meaning in a response header is not specified.[46] The behavior of Pragma: no-cache in a response is implementation specific. While some user agents do pay attention to this field in responses, the HTTP/1.1 RFC specifically warns against relying on this behavior.

8 See also

- HTTP header injection
- HTTP ETag
- List of HTTP status codes

9 References

[1] "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". ietf.org. Retrieved 2014-07-23.

[2] Simtec Limited. "2. HTTP Headers". Retrieved 2010-09-10.

[3] Internet Engineering Task Force (2012-06-01). "RFC 6648". Retrieved 2012-11-12.

[4] "Message Headers". Iana.org. 2014-06-11. Retrieved 2014-06-12.

[5] "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". itef.org. Retrieved 2014-07-24.

[6] "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". ietf.org. Retrieved 2014-07-24.

[7] "core - Apache HTTP Server". Httpd.apache.org. Archived from the original on 2012-05-09. Retrieved 2012-03-13.

[8] "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". IETF. June 2014. Retrieved 2014-12-19.

[9] "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". IETF. June 2014. Retrieved 2014-07-24.

[10] "Try out the "Do Not Track" HTTP header". Retrieved 2011-01-31.

[11] "Web Tracking Protection: Minimum Standards and Opportunities to Innovate". Retrieved 2011-03-24.

[12] IETF Do Not Track: A Universal Third-Party Web Tracking Opt Out March 7, 2011

[13] W3C Tracking Preference Expression (DNT), January 26, 2012

[14] Amos Jeffries (2010-07-02). "SquidFaq/ConfiguringSquid - Squid Web Proxy Wiki". Retrieved 2009-09-10.

[15] The Apache Software Foundation. "mod_proxy - Apache HTTP Server Version 2.2". Retrieved 2014-11-12.

[16] Dave Steinberg (2007-04-10). "How do I adjust my SSL site to work with GeekISP's loadbalancer?". Retrieved 2010-09-30.

[17] "Helping to Secure Communication: Client to Front-End Server". 2006-07-27. Retrieved 2012-04-23.

[18] "OpenSocial Core API Server Specification 2.5.1". Retrieved 2014-10-08.

[19] "ATT Device ID". Retrieved 2012-01-14.

[20] "WAP Profile". Retrieved 2012-01-14.

[21] "The Proxy-Connection: header is a mistake in how some web browsers use HTTP.". Retrieved 2014-01-02.

[22] "Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls". Electronic Frontier Foundation. Retrieved 2014-01-19.

[23] "Checking known AT&T, Verizon, Sprint, Bell Canada & Vodacom Unique Identifier beacons". Retrieved 2014-01-19.

[24] Craig Timberg. "Verizon, AT&T tracking their users with 'supercookies'". The Washington Post. Retrieved 2014-01-19.

[25] "SAP Cross-Site Request Forgery Protection". SAP SE. Retrieved 2015-01-20.

[26] "Django Cross Site Request Forgery protection". Django (web framework). Retrieved 2015-01-20.

[27] "Angular Cross Site Request Forgery (XSRF) Protection". AngularJS. Retrieved 2015-01-20.

[28] "RFC 5789". Retrieved 2014-12-24.

[29] "RFC 2183". Retrieved 2012-05-17.

[30] Indicate the canonical version of a URL by responding with the Link rel="canonical" HTTP header Retrieved: 2012-02-09

[31] W3C P3P Work Suspended

[32] "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". Retrieved 2014-07-24.

[33] "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". Retrieved 2014-07-24.

[34] "HTTP Header Field X-Frame-Options". IETF. 2013. Retrieved 2014-06-12.

[35] "The mess with X-Frame-Options: ALLOWALL". IPSec.pl. Retrieved 1 March 2013.

[36] "Content Security Policy Level 2". Retrieved 2014-08-02.

[37] "Public Key Pinning Extension for HTTP". IETF. Retrieved 8 January 2015.

[38] "Public key pins, a new safeguard for HTTPS websites". IPSec.pl. 2014. Retrieved 2014-06-12.

[39] Eric Lawrence (2008-07-02). "IE8 Security Part IV: The XSS Filter". Retrieved 2010-09-30.

[40] "Content Security Policy". W3C. 2012. Retrieved 2013.

[41] Eric Lawrence (2008-09-03). "IE8 Security Part VI: Beta 2 Update". Retrieved 2010-09-28.

[42] "Hosting - Google Chrome Extensions - Google Code". Retrieved 2012-06-14.

[43] "Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow". Retrieved 2010-09-30.

[44] "Defining Document Compatibility: Specifying Document Compatibility Modes". 2011-04-01. Retrieved 2012-01-24.

[45] "Configuring servers for Ogg media". 2014-05-26. Retrieved 2015-01-03.

[46] "Hypertext Transfer Protocol (HTTP/1.1): Caching". ietf.org. Retrieved 2014-07-24.

10 External links

- Headers: Permanent Message Header Field Names
- RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing
- RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content
- RFC 7232: Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests
- RFC 7233: Hypertext Transfer Protocol (HTTP/1.1): Range Requests
- RFC 7234: Hypertext Transfer Protocol (HTTP/1.1): Caching
- RFC 7235: Hypertext Transfer Protocol (HTTP/1.1): Authentication
- RFC 2965: IETF HTTP State Management Mechanism RFC
- HTTP/1.1 headers from a web server point of view
- HTTP Request Header Viewer
- HTTP Response Header Viewer - Retrieves the HTTP response headers of any domain.
- Internet Explorer and Custom HTTP Headers - EricLaw's IEInternals - Site Home - MSDN Blogs
- crwlr.net - HTTP Header index
- HTTP Header with Privacyinfo - Display your HTTP request and response headers

11 Text and image sources, contributors, and licenses

11.1 Text

- **List of HTTP header fields** *Source:* http://en.wikipedia.org/wiki/List%20of%20HTTP%20header%20fields?oldid=643360853 *Contributors:* Edward, Ixfd64, Mac, Brettz9, Timwi, Mcenedella, HaeB, Eric Harris-Braun, Kravietz, Adam McMaster, Vadmium, Abdull, Reschke, Rich Farmbrough, Pmsyyz, Cygnosis, Mnot, Foobaz, Towel401, Nevyn, Justinc, Timmywimmy, Psz, Mbloore, Dan East, Mindmatrix, Wbeek, Husky, Pmc, Gurch, Sstrader, Bgwhite, Rawlogic, RadioFan2 (usurped), FuzzyBSc, Smithkennedy, Bluezy, Notanumber, SmackBot, Gilliam, KiloByte, OrangeDog, Will Thompson, Octahedron80, Cbuckley, GBuilder, SeL, CmdrObot, Walling, Voltaic, TheJosh, Widefox, Guy Macon, Brianary, Antwan911, Wseltzer, Kgfleischmann, Connor Behan, RockMFR, Ianbicking, Tweisbach, Philip Trueman, EvanCarroll, Kbrose, Jesdisciple, Jpala, TheseusXav, Oxymoron83, Sliwers, Cowo, ArchiSchmedes, RobinHood70, Sfan00 IMG, Vladkornea, The Thing That Should Not Be, Spandrawn, AWiersch, GrelorBlood, SunnySideOfStreet, Koyama, Bert.Roos, Oore, DumZiBoT, Zachofalltrades, Mabdul, FrankAndProust, Scientus, Fluffernutter, AndersBot, Didz93, Jarble, JakobVoss, Yobot, Povlhp, Mmxx, AnomieBOT, Theoprakt, PoweredByLolcats, Ammubhave, Apoorv020, SmoothPorcupine, SD5, Annihilator SP, FrescoBot, Tellatime-corp, BlaF, Kenticus, Poke64, Holiverh, Impala2009, Edsu, Lotje, Ccutrer, Ronixus, Checat, EmausBot, GoingBatty, Xmm0, Hughesey, Bombshop, Gynvael, HenryHayes, Juergeen, Flaxious, Pleasancoder, BG19bot, Sylvainlasnier, Timotheus1, Chmarkine, BattyBot, Cembon, Aaronmhamilton, Blevintron, Mpmel, Dpant, ArmbrustBot, Ginsuloft, Btorcaso, Meteor sandwich yum, Skr15081997, WikiVoluntario, Byunghyun Bang, Uplex slink and Anonymous: 139

11.2 Images

11.3 Content license

- Creative Commons Attribution-Share Alike 3.0