List of HTTP header fields

From Wikipedia, the free encyclopedia
Jump to: navigation, search

HTTP

- Persistence
- Compression
- HTTPS

Request methods

- OPTIONS
- GET
- HEAD
- POST
- PUT
- DELETE
- TRACE
- CONNECT
- PATCH

Header fields

- Cookie
- ETag
- Location
- HTTP referer
- DNT
- X-Forwarded-For

Status codes

- 301 Moved Permanently
- 302 Found
- 303 See Other
- 403 Forbidden
- 404 Not Found
- 451 Unavailable For Legal Reasons

- v

- t
- e

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.

Contents

[hide]

General format[edit]

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]

Field names[edit]

A core set of fields is standardized by the Internet Engineering Task Force (IETF) in RFCs 7230, 7231, 7232, 7233, 7234, and 7235. The

permanent registry of header fields and repository of provisional registrations are maintained by the IANA. Additional field names and permissible values may be defined by each application.

Non-standard header fields were conventionally marked by prefixing the field name with X-[2] but this convention was deprecated in June 2012 because of the inconveniences it caused when non-standard fields became standard.[3] An earlier restriction on use of Downgraded- was lifted in March 2013.[4]

Field values[edit]

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]

Size limits[edit]

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]

Request fields[edit]

Header field name	Description	Example	Statu
Accept	Content-Types that are acceptable for the response. See Content negotiation.	Accept: text/plain	Permaner
Accept-Charset	Character sets that are acceptable	Accept-Charset: utf-8	Permaner
Accept-Encoding	List of acceptable encodings. See	Accept-Encoding: gzip, deflate	Permaner

Header field name	Description	Example	Statu
	HTTP compression.		
Accept-Language	List of acceptable human languages for response. See Content negotiation.	Accept-Language: en-US	Permaner
Accept-Datetime	Acceptable version in time	Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT	Provisio
Authorization	Authentication credentials for HTTP authentication	Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==	Permaner
Cache-Control	Used to specify directives that *must* be obeyed by all caching mechanisms along the request-response chain	Cache-Control: no-cache	Permaner
Connection	Control options for the current connection and list of hop-by-hop request fields[8]	Connection: keep-alive Connection: Upgrade	Permaner
Cookie	An HTTP cookie previously sent by the server with Set-Cookie (below)	Cookie: $Version=1; Skin=new;	Permaner standard
Content-Length	The length of the request body in octets (8-bit bytes)	Content-Length: 348	Permaner
Content-MD5	A Base64-encoded binary MD5 sum of the	Content-MD5: Q2hlY2sgSW5OZWdyaXR5IQ==	Obsolete

Header field name	Description	Example	Statu
	content of the request body		
Content-Type	The MIME type of the body of the request (used with POST and PUT requests)	Content-Type: application/x-www-form-urlencoded	Permaner
Date	The date and time that the message was sent (in "HTTP-date" format as defined by RFC 7231 Date/Time Formats)	Date: Tue, 15 Nov 1994 08:12:31 GMT	Permaner
Expect	Indicates that particular server behaviors are required by the client	Expect: 100-continue	Permaner
Forwarded	Disclose original information of a client connecting to a web server through an HTTP proxy [10]	Forwarded: for=192.0.2.60;proto=http;by=203.0.113.43 Forwarded: for=192.0.2.43, for=198.51.100.17	Permaner
From	The email address of the user making the request	From: user@example.com	Permaner
Host	The domain name of the server (for virtual hosting), and the TCP port number on which the server is listening. The	Host: en.wikipedia.org:80 Host: en.wikipedia.org	Permaner

Header field name	Description	Example	Statu
	port number may be omitted if the port is the standard port for the service requested. [11] Mandatory since HTTP/1.1.		
If-Match	Only perform the action if the client supplied entity matches the same entity on the server. This is mainly for methods like PUT to only update a resource if it has not been modified since the user last updated it.	If-Match: "737060cd8c284d8af7ad3082f209582d"	Permaner
If-Modified-Since	Allows a *304 Not Modified* to be returned if content is unchanged	If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT	Permaner
If-None-Match	Allows a *304 Not Modified* to be returned if content is unchanged, see HTTP ETag	If-None-Match: "737060cd8c284d8af7ad3082f209582d"	Permaner
If-Range	If the entity is unchanged, send me the part(s) that I am missing; otherwise, send	If-Range: "737060cd8c284d8af7ad3082f209582d"	Permaner

Header field name	Description	Example	Statu
	me the entire new entity		
If-Unmodified-Since	Only send the response if the entity has not been modified since a specific time.	If-Unmodified-Since: Sat, 29 Oct 1994 19:43:31 GMT	Permaner
Max-Forwards	Limit the number of times the message can be forwarded through proxies or gateways.	Max-Forwards: 10	Permaner
Origin	Initiates a request for cross-origin resource sharing (asks server for an 'Access-Control-Allow-Origin' response field) .	Origin: http://www.example-social-network.com	Permaner standard
Pragma	Implementation-specific fields that may have various effects anywhere along the request-response chain.	Pragma: no-cache	Permaner
Proxy-Authorization	Authorization credentials for connecting to a proxy.	Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==	Permaner
Range	Request only part of an entity. Bytes are numbered from 0. See Byte serving.	Range: bytes=500-999	Permaner

Header field name	Description	Example	Statu
Referer [*sic*]	This is the address of the previous web page from which a link to the currently requested page was followed. (The word “referrer” has been misspelled in the RFC as well as in most implementations to the point that it has become standard usage and is considered correct terminology)	Referer: http://en.wikipedia.org/wiki/Main_Page	Permaner
TE	The transfer encodings the user agent is willing to accept: the same values as for the response header field Transfer-Encoding can be used, plus the "trailers" value (related to the "chunked" transfer method) to notify the server it expects to receive	TE: trailers, deflate	Permaner

Header field name	Description	Example	Statu
	additional fields in the trailer after the last, zero-sized, chunk.		
User-Agent	The user agent string of the user agent	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101 Firefox/21.0	Permaner
Upgrade	Ask the server to upgrade to another protocol.	Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11	Permaner
Via	Informs the server of proxies through which the request was sent.	Via: 1.0 fred, 1.1 example.com (Apache/1.1)	Permaner
Warning	A general warning about possible problems with the entity body.	Warning: 199 Miscellaneous warning	Permaner

Common non-standard request fields[edit]

Field name	Description	Example
X-Requested-With	mainly used to identify Ajax requests. Most JavaScript frameworks send this field with value of XMLHttpRequest	X-Requested-With: XMLHttpRequest
DNT[12]	Requests a web application to disable their tracking of a user. This is Mozilla's version	DNT: 1 (Do Not Track Enabled) DNT: 0 (Do Not Track Disabled)

Field name	Description	Example
	of the X-Do-Not-Track header field (since Firefox 4.0 Beta 11). Safari and IE9 also have support for this field.[13] On March 7, 2011, a draft proposal was submitted to IETF.[14] The W3C Tracking Protection Working Group is producing a specification.[15]	
X-Forwarded-For[16]	a *de facto* standard for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer	X-Forwarded-For: client1, proxy1, proxy2 X-Forwarded-For: 129.78.138.66, 129.78.64.103
X-Forwarded-Host[17]	a *de facto* standard for identifying the original host requested by the client in the Host HTTP request header, since the host name and/or port of the reverse proxy (load balancer) may differ from the origin server handling the request.	X-Forwarded-Host: en.wikipedia.org:80 X-Forwarded-Host: en.wikipedia.org

Field name	Description	Example
X-Forwarded-Proto[18]	a *de facto* standard for identifying the originating protocol of an HTTP request, since a reverse proxy (or a load balancer) may communicate with a web server using HTTP even if the request to the reverse proxy is HTTPS. An alternative form of the header (X-ProxyUser-Ip) is used by Google clients talking to Google servers.	X-Forwarded-Proto: https
Front-End-Https[19]	Non-standard header field used by Microsoft applications and load-balancers	Front-End-Https: on
X-Http-Method-Override[20]	Requests a web application override the method specified in the request (typically POST) with the method given in the header field (typically PUT or DELETE). Can be used when a user agent or firewall prevents PUT or DELETE methods from being sent directly (note	X-HTTP-Method-Override: DELETE

Field name	Description	Example
	that this either a bug in the software component, which ought to be fixed, or an intentional configuration, in which case bypassing it may be the wrong thing to do).	
X-ATT-DeviceId[21]	Allows easier parsing of the MakeModel/Firmware that is usually found in the User-Agent String of AT&T Devices	X-Att-Deviceid: GT-P7320/P7320XXLPG
X-Wap-Profile[22]	Links to an XML file on the Internet with a full description and details about the device currently connecting. In the example to the right is an XML file for an AT&T Samsung Galaxy S2.	x-wap-profile: http://wap.samsungmobile.com/uaprof/SGH-I777.xml
Proxy-Connection[23]	Implemented as a misunderstanding of the HTTP specifications. Common because of mistakes in implementations of early HTTP versions. Has exactly the same functionality as	Proxy-Connection: keep-alive

Field name	Description	Example
	standard Connection field.	
X-UIDH[24][25][26]	Server-side deep packet insertion of a unique ID identifying customers of Verizon Wireless; also known as "perma-cookie" or "supercookie"	X-UIDH: ...
X-Csrf-Token[27]	Used to prevent cross-site request forgery. Alternative header names are: X-CSRFToken[28] and X-XSRF-TOKEN[29]	X-Csrf-Token: i8XNjC4b8KVok4uw5RftR38Wgp2BFwql

Response fields[edit]

Field name	Description	
Access-Control-Allow-Origin	Specifying which web sites can participate in cross-origin resource sharing	Access-Control-Allow-Origin:
Accept-Patch[30]	Specifies which patch document formats this server supports	Accept-Patch: text/example;ch
Accept-Ranges	What partial content range types this server supports via byte serving	Accept-Ranges: bytes
Age	The age the object has been in a proxy cache in seconds	Age: 12
Allow	Valid actions for a specified resource. To be used for a *405 Method not allowed*	Allow: GET, HEAD
Alt-Svc[31]	A server uses "Alt-Svc" header (meaning Alternative Services) to indicate that its resources can also be accessed at a different	Alt-Svc: h2="http2.example.co

Field name	Description	
	network location (host or port) or using a different protocol	
Cache-Control	Tells all caching mechanisms from server to client whether they may cache this object. It is measured in seconds	Cache-Control: max-age=3600
Connection	Control options for the current connection and list of hop-by-hop response fields[8]	Connection: close
Content-Disposition[32]	An opportunity to raise a "File Download" dialogue box for a known MIME type with binary format or suggest a filename for dynamic content. Quotes are necessary with special characters.	Content-Disposition: attachme
Content-Encoding	The type of encoding used on the data. See HTTP compression.	Content-Encoding: gzip
Content-Language	The natural language or languages of the intended audience for the enclosed content[33]	Content-Language: da
Content-Length	The length of the response body in octets (8-bit bytes)	Content-Length: 348
Content-Location	An alternate location for the returned data	Content-Location: /index.htm
Content-MD5	A Base64-encoded binary MD5 sum of the content of the response	Content-MD5: Q2hlY2sgSW5OZWdy
Content-Range	Where in a full body message this partial message belongs	Content-Range: bytes 21010-47
Content-Type	The MIME type of this content	Content-Type: text/html; char
Date	The date and time that the message was sent (in "HTTP-date" format as defined by RFC 7231)	Date: Tue, 15 Nov 1994 08:12:

Field name	Description	
ETag	An identifier for a specific version of a resource, often a message digest	ETag: "737060cd8c284d8af7ad30
Expires	Gives the date/time after which the response is considered stale (in "HTTP-date" format as defined by RFC 7231)	Expires: Thu, 01 Dec 1994 16:
Last-Modified	The last modified date for the requested object (in "HTTP-date" format as defined by RFC 7231)	Last-Modified: Tue, 15 Nov 19
Link	Used to express a typed relationship with another resource, where the relation type is defined by RFC 5988	Link: </feed>; rel="alternate
Location	Used in redirection, or when a new resource has been created.	Location: http://www.w3.org/p
P3P	This field is supposed to set P3P policy, in the form of P3P:CP="your_compact_policy". However, P3P did not take off,[36] most browsers have never fully implemented it, a lot of websites set this field with fake policy text, that was enough to fool browsers the existence of P3P policy and grant permissions for third party cookies.	P3P: CP="This is not a P3P po http://www.google.com/support for more info."
Pragma	Implementation-specific fields that may have various effects anywhere along the request-response chain.	Pragma: no-cache
Proxy-Authenticate	Request authentication to access the proxy.	Proxy-Authenticate: Basic
Public-Key-Pins[37]	HTTP Public Key Pinning, announces hash of website's authentic TLS certificate	Public-Key-Pins: max-age=2592 sha256="E9CZ9INDbd+2eRQozYqqb

Field name	Description	
Refresh	Used in redirection, or when a new resource has been created. This refresh redirects after 5 seconds.	Refresh: 5; url=http://www.w3
Retry-After	If an entity is temporarily unavailable, this instructs the client to try again later. Value could be a specified period of time (in seconds) or a HTTP-date.[38]	• Example 1: Retry-After: • Example 2: Retry-After:
Server	A name for the server	Server: Apache/2.4.1 (Unix)
Set-Cookie	An HTTP cookie	Set-Cookie: UserID=JohnDoe; M
Status	CGI header field specifying the status of the HTTP response. Normal HTTP responses use a separate "Status-Line" instead, defined by RFC 7230.[39]	Status: 200 OK
Strict-Transport-Security	A HSTS Policy informing the HTTP client how long to cache the HTTPS only policy and whether this applies to subdomains.	Strict-Transport-Security: ma
Trailer	The Trailer general field value indicates that the given set of header fields is present in the trailer of a message encoded with chunked transfer coding.	Trailer: Max-Forwards
Transfer-Encoding	The form of encoding used to safely transfer the entity to the user. Currently defined methods are: chunked,	Transfer-Encoding: chunked

Field name	Description	
	compress, deflate, gzip, identity.	
TSV	Tracking Status Value, value suggested to be sent in response to a DNT(do-not-track), possible values: "!" — under construction "?" — dynamic "G" — gateway to multiple parties "N" — not tracking "T" — tracking "C" — tracking with consent "P" — tracking only if consented "D" — disregarding DNT "U" — updated	TSV: ?
Upgrade	Ask the client to upgrade to another protocol.	Upgrade: HTTP/2.0, SHTTP/1.3,
Vary	Tells downstream proxies how to match future request headers to decide whether the cached response can be used rather than requesting a fresh one from the origin server.	• Example 1: Vary: * • Example 2: Vary: Accept
Via	Informs the client of proxies through which the response was sent.	Via: 1.0 fred, 1.1 example.co
Warning	A general warning about possible problems with the entity body.	Warning: 199 Miscellaneous wa
WWW-Authenticate	Indicates the authentication scheme that should be used to access the requested entity.	WWW-Authenticate: Basic
X-Frame-Options[40]	Clickjacking protection: deny - no rendering within a frame, sameorigin - no rendering if origin mismatch, allow-from - allow from specified location, allowall	X-Frame-Options: deny

Field name	Description	
	- non-standard, allow from any location	

Common non-standard response fields[edit]

Field name	Description	Example
X-XSS-Protection[42]	Cross-site scripting (XSS) filter	X-XSS-Protection: 1; mode=block
Content-Security-Policy, *X-Content-Security-Policy*, *X-WebKit-CSP*[43]	Content Security Policy definition.	X-WebKit-CSP: default-src 'self'
X-Content-Type-Options[44]	The only defined value, "nosniff", prevents Internet Explorer from MIME-sniffing a response away from the declared content-type. This also applies to Google Chrome, when downloading extensions.[45]	X-Content-Type-Options: nosniff
X-Powered-By[46]	specifies the technology (e.g. ASP.NET, PHP, JBoss) supporting the web application (version details are often in X-Runtime, X-Version, or X-AspNet-Version)	X-Powered-By: PHP/5.4.0
X-UA-Compatible[47]	Recommends the preferred rendering engine (often a backward-compatibility mode) to use to display the content. Also used to activate Chrome Frame in Internet Explorer.	X-UA-Compatible: IE=EmulateIE7 X-UA-Compatible: IE=edge X-UA-Compatible: Chrome=1
X-Content-Duration[48]	Provide the duration of the audio or video in seconds; only supported by Gecko browsers	X-Content-Duration: 42.666

Field name	Description	Example
Upgrade-Insecure-Requests[49]	Tells a server which (presumably in the middle of a HTTP -> HTTPS migration) hosts mixed content that the client would prefer redirection to HTTPS and can handle Content-Security-Policy: upgrade-insecure-requests	Upgrade-Insecure-Requests: 1

Effects of selected fields[edit]

Avoiding caching[edit]

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.[50] The behavior of Pragma: no-cache in a response is implementation specific. While some user agents do pay attention to this field in responses,[51] the HTTP/1.1 RFC specifically warns against relying on this behavior.

See also[edit]

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

References[edit]

1. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing"*. *ietf.org. Retrieved 2014-07-23.*
2. Jump up ^ *Simtec Limited. "2. HTTP Headers". Retrieved 2010-09-10.*
3. Jump up ^ *Internet Engineering Task Force (2012-06-01). "RFC 6648". Retrieved 2012-11-12.*
4. Jump up ^ *"Message Headers". Iana.org. 2014-06-11. Retrieved 2014-06-12.*
5. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". itef.org. Retrieved 2014-07-24.*
6. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". ietf.org. Retrieved 2014-07-24.*
7. Jump up ^ *"core - Apache HTTP Server". Httpd.apache.org. Archived from the original on 2012-05-09. Retrieved 2012-03-13.*
8. ^ Jump up to: a b *"Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". IETF. June 2014. Retrieved 2014-12-19.*
9. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". Retrieved 2015-06-03.*
10. Jump up ^ *"Forwarded HTTP Extension: Introduction". IETF. June 2014. Retrieved 2016-01-07.*
11. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". IETF. June 2014. Retrieved 2014-07-24.*
12. Jump up ^ *"Try out the "Do Not Track" HTTP header". Retrieved 2011-01-31.*
13. Jump up ^ *"Web Tracking Protection: Minimum Standards and Opportunities to Innovate". Retrieved 2011-03-24.*
14. Jump up ^ IETF Do Not Track: A Universal Third-Party Web Tracking Opt Out March 7, 2011
15. Jump up ^ W3C Tracking Preference Expression (DNT), January 26, 2012

16. Jump up ^ *Amos Jeffries (2010-07-02). "SquidFaq/ConfiguringSquid - Squid Web Proxy Wiki". Retrieved 2009-09-10.*
17. Jump up ^ *The Apache Software Foundation. "mod_proxy - Apache HTTP Server Version 2.2". Retrieved 2014-11-12.*
18. Jump up ^ *Dave Steinberg (2007-04-10). "How do I adjust my SSL site to work with GeekISP's loadbalancer?". Retrieved 2010-09-30.*
19. Jump up ^ *"Helping to Secure Communication: Client to Front-End Server". 2006-07-27. Retrieved 2012-04-23.*
20. Jump up ^ *"OpenSocial Core API Server Specification 2.5.1". Retrieved 2014-10-08.*
21. Jump up ^ *"ATT Device ID". Retrieved 2012-01-14.*
22. Jump up ^ *"WAP Profile". Retrieved 2012-01-14.*
23. Jump up ^ *"The Proxy-Connection: header is a mistake in how some web browsers use HTTP.". Retrieved 2014-01-02.*
24. Jump up ^ *"Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls". Electronic Frontier Foundation. Retrieved 2014-01-19.*
25. Jump up ^ *"Checking known AT&T, Verizon, Sprint, Bell Canada & Vodacom Unique Identifier beacons". Retrieved 2014-01-19.*
26. Jump up ^ *Craig Timberg. "Verizon, AT&T tracking their users with 'supercookies'". The Washington Post. Retrieved 2014-01-19.*
27. Jump up ^ *"SAP Cross-Site Request Forgery Protection". SAP SE. Retrieved 2015-01-20.*
28. Jump up ^ *"Django Cross Site Request Forgery protection". Django (web framework). Retrieved 2015-01-20.*
29. Jump up ^ *"Angular Cross Site Request Forgery (XSRF) Protection". AngularJS. Retrieved 2015-01-20.*
30. Jump up ^ *"RFC 5789". Retrieved 2014-12-24.*
31. Jump up ^ *"HTTP Alternative Services". IETF. April 2016. Retrieved 2016-04-19.*
32. Jump up ^ *"RFC 6266". Retrieved 2015-03-13.*
33. Jump up ^ https://tools.ietf.org/html/rfc7231#section-3.1.3.2
34. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". Retrieved 2015-06-03.*
35. Jump up ^ Indicate the canonical version of a URL by responding with the Link rel="canonical" HTTP header Retrieved: 2012-02-09
36. Jump up ^ W3C P3P Work Suspended
37. Jump up ^ *"Public Key Pinning Extension for HTTP". IETF. Retrieved 17 April 2015.*
38. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". Retrieved 2014-07-24.*
39. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing". Retrieved 2014-07-24.*
40. Jump up ^ *"HTTP Header Field X-Frame-Options". IETF. 2013. Retrieved 2014-06-12.*
41. Jump up ^ *"Content Security Policy Level 2". Retrieved 2014-08-02.*
42. Jump up ^ *Eric Lawrence (2008-07-02). "IE8 Security Part IV: The XSS Filter". Retrieved 2010-09-30.*
43. Jump up ^ *"Content Security Policy". W3C. 2012. Retrieved 2013.*
44. Jump up ^ *Eric Lawrence (2008-09-03). "IE8 Security Part VI: Beta 2 Update". Retrieved 2010-09-28.*
45. Jump up ^ *"Hosting - Google Chrome Extensions - Google Code". Retrieved 2012-06-14.*

46. Jump up ^ *"Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow"*. *Retrieved 2010-09-30.*
47. Jump up ^ *"Defining Document Compatibility: Specifying Document Compatibility Modes"*. *2011-04-01. Retrieved 2012-01-24.*
48. Jump up ^ *"Configuring servers for Ogg media"*. *2014-05-26. Retrieved 2015-01-03.*
49. Jump up ^ *"Upgrade Insecure Requests - W3C Candidate Recommendation"*. *W3C. 8 October 2015. Retrieved 14 January 2016.*
50. Jump up ^ *"Hypertext Transfer Protocol (HTTP/1.1): Caching"*. *ietf.org. Retrieved 2014-07-24.*
51. Jump up ^ *"How to prevent caching in Internet Explorer"*. *Microsoft. 2011-09-22. Retrieved 2015-04-15.*

External links[edit]

- 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
- Internet Explorer and Custom HTTP Headers - EricLaw's IEInternals - Site Home - MSDN Blogs

Categories:

- Hypertext Transfer Protocol
- Hypertext Transfer Protocol headers
- Internet-related lists

Navigation menu

Personal tools

- Not logged in

- Talk
- Contributions
- Create account
- Log in

Namespaces

- Article
- Talk

Variants

Views

- Read
- Edit
- View history

More

Search

Go

Navigation

- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wikipedia
- Wikipedia store

Interaction

- Help
- About Wikipedia
- Community portal
- Recent changes
- Contact page

Tools

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information
- Wikidata item
- Cite this page

Print/export

- Create a book
- Download as PDF
- Printable version

Languages

- Deutsch
- Español
- فارسی
- Polski
- Português
- Русский
- Српски / srpski
- ไทย
- 中文

Edit links

- 
a
WIKIMEDIA
project
- 
Powered By
MediaWiki