



[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](#) 
[العربية](#)
[Deutsch](#)
[Español](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)



List of HTTP header fields

From Wikipedia, the free encyclopedia

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](#)
- [Field names](#)
- [Field values](#)
- [Size limits](#)
- [Request fields](#)
 - [Standard request fields](#)
 - [Common non-standard request fields](#)
- [Response fields](#)
 - [Standard response fields](#)
 - [Common non-standard response fields](#)
- [Effects of selected fields](#)
 - [Avoiding caching](#)
- [See also](#)
- [References](#)
- [External links](#)

General format [\[edit\]](#)

The header fields are transmitted after the request line (in case of a request HTTP message) or the response line (in case of a response HTTP message), which is the first line of a message. Header fields are colon-separated key-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(line), resulting in the transmission of two consecutive CR-LF pairs. In the past, 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.

HTTP

[Persistence](#) · [Compression](#) · [HTTPS](#) · [QUIC](#)

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](#)

Security access control methods

[Basic access authentication](#) · [Digest access authentication](#)

[V](#) · [T](#) · [E](#)

Header field names are case-insensitive^[2]. This is in contrast to HTTP method names (GET, POST, etc.), which are case-sensitive^{[3][4]}.

HTTP/2 makes some restrictions on specific header fields (see below).

Non-standard header fields were conventionally marked by prefixing the field name with `X-` but this convention was deprecated in June 2012 because of the inconveniences it caused when non-standard fields became standard.^[5] An earlier restriction on use of `Downgraded-` was lifted in March 2013.^[6]

Field values [\[edit \]](#)

A few fields can contain comments (i.e. in User-Agent, Server, Via fields), which can be ignored by software.^[7]

Many field values may contain a quality (*q*) key-value pair separated by [equals sign](#), specifying a weight to use in [content negotiation](#).^[8]

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 8,190 bytes, and there can be at most 100 header fields in a single request.^[9]

Request fields [\[edit \]](#)

Standard request fields [\[edit \]](#)

Header field name ↕	Description	Example	Status ↕
A-IM	Acceptable instance-manipulations for the request ^[10] .	A-IM: feed	Permanent
Accept	Media type(s) that is(/are) acceptable for the response. See Content negotiation .	Accept: text/html	Permanent
Accept-Charset	Character sets that are acceptable.	Accept-Charset: utf-8	Permanent
Accept-Encoding	List of acceptable encodings. See HTTP compression .	Accept-Encoding: gzip, deflate	Permanent
Accept-Language	List of acceptable human languages for response. See Content negotiation .	Accept-Language: en-US	Permanent
Accept-Datetime	Acceptable version in time.	Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT	Provisional
Access-Control-Request-Method, Access-Control-Request-Headers ^[11]	Initiates a request for cross-origin resource sharing with Origin (below).	Access-Control-Request-Method: GET	Permanent: standard
Authorization	Authentication credentials for HTTP authentication .	Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==	Permanent
Cache-Control	Used to specify directives that <i>must</i> be obeyed by all caching mechanisms along the request-response chain.	Cache-Control: no-cache	Permanent

Header field name	Description	Example	Status
Connection	Control options for the current connection and list of hop-by-hop request fields. ^[12] Must not be used with HTTP/2. ^[13]	Connection: keep-alive Connection: Upgrade	Permanent
Content-Length	The length of the request body in octets (8-bit bytes).	Content-Length: 348	Permanent
Content-MD5	A Base64 -encoded binary MD5 sum of the content of the request body.	Content-MD5: Q2hLY2sgSW50ZWdyaXR5IQ==	Obsolete ^[14]
Content-Type	The Media type of the body of the request (used with POST and PUT requests).	Content-Type: application/x-www-form-urlencoded	Permanent
Cookie	An HTTP cookie previously sent by the server with Set-Cookie (below).	Cookie: \$Version=1; Skin=new;	Permanent: standard
Date	The date and time at which the message was originated (in "HTTP-date" format as defined by RFC 7231 Date/Time Formats [Ⓔ]).	Date: Tue, 15 Nov 1994 08:12:31 GMT	Permanent
Expect	Indicates that particular server behaviors are required by the client.	Expect: 100-continue	Permanent
Forwarded	Disclose original information of a client connecting to a web server through an HTTP proxy. ^[15]	Forwarded: for=192.0.2.60;proto=http;by=203.0.113.43 Forwarded: for=192.0.2.43, for=198.51.100.17	Permanent
From	The email address of the user making the request.	From: user@example.com	Permanent
Host	The domain name of the server (for virtual hosting), and the TCP port number on which the server is listening. The port number may be omitted if the port is the standard port for the service requested. Mandatory since HTTP/1.1. ^[16] If the request is generated directly in HTTP/2, it should not be used. ^[17]	Host: en.wikipedia.org:8080 Host: en.wikipedia.org	Permanent
HTTP2-Settings	A request that upgrades from HTTP/1.1 to HTTP/2 MUST include exactly one HTTP2-Setting header field. The HTTP2-Settings header field is a connection-specific header field that includes parameters that govern the HTTP/2 connection, provided in anticipation of the server accepting the request to upgrade. ^{[18][19]}	HTTP2-Settings: token64	Permanent: standard

Header field name	Description	Example	Status
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"	Permanent
If-Modified-Since	Allows a <i>304 Not Modified</i> to be returned if content is unchanged.	If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT	Permanent
If-None-Match	Allows a <i>304 Not Modified</i> to be returned if content is unchanged, see HTTP ETag .	If-None-Match: "737060cd8c284d8af7ad3082f209582d"	Permanent
If-Range	If the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity.	If-Range: "737060cd8c284d8af7ad3082f209582d"	Permanent
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	Permanent
Max-Forwards	Limit the number of times the message can be forwarded through proxies or gateways.	Max-Forwards: 10	Permanent
Origin ^[11]	Initiates a request for cross-origin resource sharing (asks server for Access-Control-* response fields).	Origin: http://www.example-social-network.com	Permanent: standard
Pragma	Implementation-specific fields that may have various effects anywhere along the request-response chain.	Pragma: no-cache	Permanent
Proxy-Authorization	Authorization credentials for connecting to a proxy.	Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==	Permanent
Range	Request only part of an entity. Bytes are numbered from 0. See Byte serving .	Range: bytes=500-999	Permanent
Referer <i>[sic]</i>	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	Permanent

Header field name	Description	Example	Status
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 additional fields in the trailer after the last, zero-sized, chunk. Only trailers is supported in HTTP/2. ^[13]	TE: trailers, deflate	Permanent
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/12.0	Permanent
Upgrade	Ask the server to upgrade to another protocol. Must not be used in HTTP/2. ^[13]	Upgrade: h2c, HTTPS/1.3, IRC/6.9, RTA/x11, websocket	Permanent
Via	Informs the server of proxies through which the request was sent.	Via: 1.0 fred, 1.1 example.com (Apache/1.1)	Permanent
Warning	A general warning about possible problems with the entity body.	Warning: 199 Miscellaneous warning	Permanent

Common non-standard request fields [\[edit \]](#)

Field name	Description	Example
Upgrade-Insecure-Requests ^[20]	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 Must not be used with HTTP/2 ^[13]	Upgrade-Insecure-Requests: 1
X-Requested-With	Mainly used to identify Ajax requests. Most JavaScript frameworks send this field with value of XMLHttpRequest	X-Requested-With: XMLHttpRequest
DNT ^[21]	Requests a web application to disable their tracking of a user. This is Mozilla's version of the X-Do-Not-Track header field (since Firefox 4.0 Beta 11). Safari and IE9 also have support for this field. ^[22] On March 7, 2011, a draft proposal was submitted to IETF. ^[23] The W3C Tracking Protection Working Group is producing a specification. ^[24]	DNT: 1 (Do Not Track Enabled) DNT: 0 (Do Not Track Disabled)

Field name	Description	Example
X-Forwarded-For ^[25]	A <i>de facto</i> standard for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer. Superseded by <i>Forwarded</i> header.	X-Forwarded-For: client1, proxy1, proxy2 X-Forwarded-For: 129.78.138.66, 129.78.64.103
X-Forwarded-Host ^[26]	A <i>de facto</i> 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. Superseded by <i>Forwarded</i> header.	X-Forwarded-Host: en.wikipedia.org:8080 X-Forwarded-Host: en.wikipedia.org
X-Forwarded-Proto ^[27]	A <i>de facto</i> 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. Superseded by <i>Forwarded</i> header.	X-Forwarded-Proto: https
Front-End-Https ^[28]	Non-standard header field used by Microsoft applications and load-balancers	Front-End-Https: on
X-Http-Method-Override ^[29]	Requests a web application to override the method specified in the request (typically POST) with the method given in the header field (typically PUT or DELETE). This can be used when a user agent or firewall prevents PUT or DELETE methods from being sent directly (note that this is 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-HTTP-Method-Override: DELETE
X-ATT-DeviceId ^[30]	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 ^[31]	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 ^[32]	Implemented as a misunderstanding of the HTTP specifications. Common because of mistakes in implementations of early HTTP versions. Has exactly the same functionality as standard Connection field. Must not be used with HTTP/2. ^[13]	Proxy-Connection: keep-alive
X-UIDH ^{[33][34][35]}	Server-side <i>deep packet insertion</i> of a unique ID identifying customers of Verizon Wireless; also known as "perma-cookie" or "supercookie"	X-UIDH: ...

Field name	Description	Example
X-Csrf-Token ^[36]	Used to prevent cross-site request forgery . Alternative header names are: <code>X-CSRFToken</code> ^[37] and <code>X-XSRF-TOKEN</code> ^[38]	<code>X-Csrf-Token: i8XNjC4b8KVok4uw5RftR38Wgp2BFwql</code>
X-Request-ID ^{[39][40]} , X-Correlation-ID ^{[41][42]}	Correlates HTTP requests between a client and server.	<code>X-Request-ID: f058ebd6-02f7-4d3f-942e-904344e8cde5</code>
Save-Data	The Save-Data client hint request header available in Chrome, Opera, and Yandex browsers lets developers deliver lighter, faster applications to users who opt-in to data saving mode in their browser.	<code>Save-Data: on</code>

Response fields [\[edit \]](#)

Standard response fields [\[edit \]](#)

Field name	Description	Example	Status
Access-Control-Allow-Origin, Access-Control-Allow-Credentials, Access-Control-Expose-Headers, Access-Control-Max-Age, Access-Control-Allow-Methods, Access-Control-Allow-Headers ^[11]	Specifying which web sites can participate in cross-origin resource sharing	<code>Access-Control-Allow-Origin: *</code>	Permanent: standard
Accept-Patch ^[43]	Specifies which patch document formats this server supports	<code>Accept-Patch: text/example; charset=utf-8</code>	Permanent
Accept-Ranges	What partial content range types this server supports via byte serving	<code>Accept-Ranges: bytes</code>	Permanent
Age	The age the object has been in a proxy cache in seconds	<code>Age: 12</code>	Permanent
Allow	Valid methods for a specified resource. To be used for a <i>405 Method not allowed</i>	<code>Allow: GET, HEAD</code>	Permanent

Field name	Description	Example	Status
Alt-Svc ^[44]	A server uses "Alt-Svc" header (meaning Alternative Services) to indicate that its resources can also be accessed at a different network location (host or port) or using a different protocol When using HTTP/2, servers should instead send an ALTSVC frame. ^[45]	Alt-Svc: http/1.1="http2.example.com:8001"; ma=7200	Permanent
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	Permanent
Connection	Control options for the current connection and list of hop-by-hop response fields. ^[12] Must not be used with HTTP/2. ^[13]	Connection: close	Permanent
Content-Disposition ^[46]	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: attachment; filename="fname.ext"	Permanent
Content-Encoding	The type of encoding used on the data. See HTTP compression .	Content-Encoding: gzip	Permanent
Content-Language	The natural language or languages of the intended audience for the enclosed content ^[47]	Content-Language: da	Permanent
Content-Length	The length of the response body in octets (8-bit bytes)	Content-Length: 348	Permanent
Content-Location	An alternate location for the returned data	Content-Location: /index.htm	Permanent
Content-MD5	A Base64 -encoded binary MD5 sum of the content of the response	Content-MD5: Q2h1Y2sgSW50ZWdyaXR5IQ==	Obsolete ^[14]

Field name	Description	Example	Status
Content-Range	Where in a full body message this partial message belongs	Content-Range: bytes 21010-47021/47022	Permanent
Content-Type	The MIME type of this content	Content-Type: text/html; charset=utf-8	Permanent
Date	The date and time that the message was sent (in "HTTP-date" format as defined by RFC 7231) ^[48]	Date: Tue, 15 Nov 1994 08:12:31 GMT	Permanent
Delta-Base	Specifies the delta-encoding entity tag of the response ^[10] .	Delta-Base: "abc"	Permanent
ETag	An identifier for a specific version of a resource, often a message digest	ETag: "737060cd8c284d8af7ad3082f209582d"	Permanent
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 16:00:00 GMT	Permanent: standard
IM	Instance-manipulations applied to the response ^[10] .	IM: feed	Permanent
Last-Modified	The last modified date for the requested object (in "HTTP-date" format as defined by RFC 7231)	Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT	Permanent
Link	Used to express a typed relationship with another resource, where the relation type is defined by RFC 5988	Link: </feed>; rel="alternate" ^[49]	Permanent
Location	Used in redirection , or when a new resource has been created.	<ul style="list-style-type: none"> Example 1: Location: <code>http://www.w3.org/pub/WWW/People.html</code> Example 2: Location: <code>/pub/WWW/People.html</code> 	Permanent
P3P	This field is supposed to set P3P policy, in the form of <code>P3P:CP="your_compact_policy"</code> . However, P3P did not take off, ^[50] 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 policy! See https://en.wikipedia.org/wiki/Special:CentralAutoLogin/P3P for more info."	Permanent

Field name	Description	Example	Status
Pragma	Implementation-specific fields that may have various effects anywhere along the request-response chain.	Pragma: no-cache	Permanent
Proxy-Authenticate	Request authentication to access the proxy.	Proxy-Authenticate: Basic	Permanent
Public-Key-Pins ^[51]	HTTP Public Key Pinning , announces hash of website's authentic TLS certificate	Public-Key-Pins: max-age=2592000; pin-sha256="E9CZ9INDbd+2eRQozYqqbQ2yXLVKB9+xcprMF+44U1g=";	Permanent
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. ^[52]	<ul style="list-style-type: none"> Example 1: Retry-After: 120 Example 2: Retry-After: Fri, 07 Nov 2014 23:59:59 GMT 	Permanent
Server	A name for the server	Server: Apache/2.4.1 (Unix)	Permanent
Set-Cookie	An HTTP cookie	Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1	Permanent: standard
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: max-age=16070400; includeSubDomains	Permanent: standard
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	Permanent
Transfer-Encoding	The form of encoding used to safely transfer the entity to the user. Currently defined methods ^[5] are: chunked , compress , deflate , gzip , identity . Must not be used with HTTP/2. ^[13]	Transfer-Encoding: chunked	Permanent

Field name	Description	Example	Status
Tk	Tracking Status header, value suggested to be sent in response to a DNT(do-not-track), possible values: <div> "! " – 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 </div>	Tk: ?	Permanent
Upgrade	Ask the client to upgrade to another protocol. Must not be used in HTTP/2 ^[13]	Upgrade: h2c, HTTPS/1.3, IRC/6.9, RTA/x11, websocket	Permanent
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.	<ul style="list-style-type: none"> Example 1: Vary: * Example 2: Vary: Accept-Language 	Permanent
Via	Informs the client of proxies through which the response was sent.	Via: 1.0 fred, 1.1 example.com (Apache/1.1)	Permanent
Warning	A general warning about possible problems with the entity body.	Warning: 199 Miscellaneous warning	Permanent
WWW-Authenticate	Indicates the authentication scheme that should be used to access the requested entity.	WWW-Authenticate: Basic	Permanent

Field name ↕	Description	Example	Status ↕
X-Frame-Options ^[53]	Clickjacking protection: deny - no rendering within a frame, sameorigin - no rendering if origin mismatch, allow-from - allow from specified location, allowall - non-standard, allow from any location	X-Frame-Options: deny	Obsolete ^[54]

Common non-standard response fields [\[edit \]](#)

Field name ↕	Description	Example
Content-Security-Policy, X-Content-Security-Policy, X-WebKit-CSP ^[55]	Content Security Policy definition.	X-WebKit-CSP: default-src 'self'
Refresh	Used in redirection, or when a new resource has been created. This refresh redirects after 5 seconds. Header extension introduced by Netscape and supported by most web browsers.	Refresh: 5; url=http://www.w3.org/pub/WWW/People.html
Status	CGI header field specifying the status of the HTTP response. Normal HTTP responses use a separate "Status-Line" instead, defined by RFC 7230 . ^[56]	Status: 200 OK
Timing-Allow-Origin	The <code>Timing-Allow-Origin</code> response header specifies origins that are allowed to see values of attributes retrieved via features of the Resource Timing API , which would otherwise be reported as zero due to cross-origin restrictions. ^[57]	Timing-Allow-Origin: * Timing-Allow-Origin: <origin>[, <origin>]*
X-Content-Duration ^[58]	Provide the duration of the audio or video in seconds; only supported by Gecko browsers	X-Content-Duration: 42.666
X-Content-Type-Options ^[59]	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. ^[60]	X-Content-Type-Options: nosniff ^[61]
X-Powered-By ^[62]	Specifies the technology (e.g. ASP.NET, PHP, JBoss) supporting the web application (version details are often in <code>X-Runtime</code> , <code>X-Version</code> , or <code>X-AspNet-Version</code>)	X-Powered-By: PHP/5.4.0

Field name ↕	Description	Example
X-Request-ID, X-Correlation-ID ^[39]	Correlates HTTP requests between a client and server.	X-Request-ID: f058ebd6-02f7-4d3f-942e-904344e8cde5
X-UA-Compatible ^[63]	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-XSS-Protection ^[64]	Cross-site scripting (XSS) filter	X-XSS-Protection: 1; mode=block

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 requests 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.^[65] The behavior of `Pragma: no-cache` in a response is implementation specific. While some user agents do pay attention to this field in responses,^[66] 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. ^a ["Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing"](#). ietf.org. Retrieved 2014-07-23.
2. ^a [RFC-7230 section 3.2](#)
3. ^a [RFC-7210 section 3.1.1](#)
4. ^a [RFC-7231 section 4.1](#)
5. ^a Internet Engineering Task Force (2012-06-01). ["RFC 6648"](#). Retrieved 2012-11-12.
6. ^a ["Message Headers"](#). iana.org. 2014-06-11. Retrieved 2014-06-12.
7. ^a ["Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing"](#). itef.org. Retrieved 2014-07-24.
8. ^a ["Hypertext Transfer Protocol \(HTTP/1.1\): Semantics and Content"](#). ietf.org. Retrieved 2014-07-24.
9. ^a ["core - Apache HTTP Server"](#). Httpd.apache.org. Archived from [the original](#) on 2012-05-09. Retrieved 2012-03-13.
10. ^{a b c} [RFC 3229](#). doi:10.17487/RFC3229.
11. ^{a b c} ["Cross-Origin Resource Sharing"](#). Retrieved 2017-07-24.
12. ^{a b} ["Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing"](#). IETF. June 2014. Retrieved 2014-12-19.
13. ^{a b c d e f g h} ["Hypertext Transfer Protocol Version 2 \(HTTP/2\)"](#). IETF. May 2015. Retrieved 2017-06-06.
14. ^{a b} ["Hypertext Transfer Protocol \(HTTP/1.1\): Semantics and Content"](#). Retrieved 2015-06-03.
15. ^a ["Forwarded HTTP Extension: Introduction"](#). IETF. June 2014. Retrieved 2016-01-07.
16. ^a ["Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing"](#). IETF. June 2014. Retrieved 2014-07-24.
17. ^a ["Hypertext Transfer Protocol Version 2 \(HTTP/2\)"](#). IETF. May 2015. Retrieved 2017-06-06.
18. ^a ["Message Headers"](#). www.iana.org. Retrieved 2018-11-26.
19. ^a ["Hypertext Transfer Protocol Version 2 \(HTTP/2\)"](#). httpwg.org. 2015-05-30. Retrieved 2019-02-22.
20. ^a ["Upgrade Insecure Requests - W3C Candidate Recommendation"](#). W3C. 8 October 2015. Retrieved 14 January 2016.
21. ^a ["Try out the "Do Not Track" HTTP header"](#). Retrieved 2011-01-31.
22. ^a ["Web Tracking Protection: Minimum Standards and Opportunities to Innovate"](#). Retrieved 2011-03-24.
23. ^a IETF [Do Not Track: A Universal Third-Party Web Tracking Opt Out](#) March 7, 2011
24. ^a [W3C Tracking Preference Expression \(DNT\)](#), January 26, 2012
25. ^a Amos Jeffries (2010-07-02). ["SquidFaq/ConfiguringSquid - Squid Web Proxy Wiki"](#). Retrieved 2009-09-10.
26. ^a The Apache Software Foundation. ["mod_proxy - Apache HTTP Server Version 2.2"](#). Retrieved 2014-11-12.
27. ^a Dave Steinberg (2007-04-10). ["How do I adjust my SSL site to work with GeekISP's loadbalancer?"](#). Retrieved 2010-09-30.
28. ^a ["Helping to Secure Communication: Client to Front-End Server"](#). 2006-07-27. Retrieved 2012-04-23.
29. ^a ["OpenSocial Core API Server Specification 2.5.1"](#). Retrieved 2014-10-08.
30. ^a ["ATT Device ID"](#). Retrieved 2012-01-14.
31. ^a ["WAP Profile"](#). Retrieved 2012-01-14.
32. ^a de Boyne Pollard, Jonathan (2007). ["The Proxy-Connection: header is a mistake in how some web browsers use HTTP"](#). Retrieved 2018-01-16.
33. ^a ["Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls"](#). Electronic Frontier Foundation. Retrieved 2014-01-19.
34. ^a ["Checking known AT&T, Verizon, Sprint, Bell Canada & Vodacom Unique Identifier beacons"](#). Retrieved 2014-01-19.
35. ^a Craig Timberg. ["Verizon, AT&T tracking their users with 'supercookies'"](#). The Washington Post. Retrieved 2014-01-19.
36. ^a ["SAP Cross-Site Request Forgery Protection"](#). SAP SE. Retrieved 2015-01-20.
37. ^a ["Django Cross Site Request Forgery protection"](#). Django (web framework). Retrieved 2015-01-20.
38. ^a ["Angular Cross Site Request Forgery \(XSRF\) Protection"](#). AngularJS. Retrieved 2015-01-20.
39. ^{a b} ["What is the X-REQUEST-ID http header?"](#). stackoverflow.com. Retrieved 2016-05-19.
40. ^a ["HTTP Request IDs"](#). devcenter.heroku.com. Retrieved 2018-02-06.
41. ^a ["The Value of Correlation IDs"](#). Rapid7 Blog. 2016-12-23. Retrieved 2018-04-13.
42. ^a Hilton, Peter. ["Correlation IDs for microservices architectures - Peter Hilton"](#). hilton.org.uk. Retrieved 2018-04-13.
43. ^a ["RFC 5789"](#). Retrieved 2014-12-24.
44. ^a ["HTTP Alternative Services"](#). IETF. April 2016. Retrieved 2016-04-19.
45. ^a ["HTTP Alternative Services, section 3"](#). IETF. April 2016. Retrieved 2017-06-08.
46. ^a ["RFC 6266"](#). Retrieved 2015-03-13.
47. ^a ["RFC 7231 - Hypertext Transfer Protocol \(HTTP/1.1\): Semantics and Content"](#). Tools.ietf.org. Retrieved 2017-12-11.
48. ^a ["RFC7231 Compliant HTTP Date Headers"](#).
49. ^a [Indicate the canonical version of a URL by responding with the Link rel="canonical" HTTP header](#) Retrieved: 2012-02-09
50. ^a [W3C P3P Work Suspended](#)
51. ^a ["Public Key Pinning Extension for HTTP"](#). IETF. Retrieved 17 April 2015.
52. ^a ["Hypertext Transfer Protocol \(HTTP/1.1\): Semantics and Content"](#). Retrieved 2014-07-24.
53. ^a ["HTTP Header Field X-Frame-Options"](#). IETF. 2013. Retrieved 2014-06-12.

54. [^] ["Content Security Policy Level 2"](#)[↗]. Retrieved 2014-08-02.
55. [^] ["Content Security Policy"](#)[↗]. W3C. 2012. Retrieved 28 April 2017.
56. [^] ["Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing"](#)[↗]. Retrieved 2014-07-24.
57. [^] ["Timing-Allow-Origin"](#)[↗]. *Mozilla Developer Network*. Retrieved 2018-01-25.
58. [^] ["Configuring servers for Ogg media"](#)[↗]. 2014-05-26. Retrieved 2015-01-03.
59. [^] Eric Lawrence (2008-09-03). ["IE8 Security Part VI: Beta 2 Update"](#)[↗]. Retrieved 2010-09-28.
60. [^] ["Hosting - Google Chrome Extensions - Google Code"](#)[↗]. Retrieved 2012-06-14.
61. [^] van Kesteren, Anne (2016-08-26). ["Fetch standard"](#)[↗]. *WHATWG*. [Archived](#)[↗] from the original on 2016-08-26. Retrieved 2016-08-26.
62. [^] ["Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow"](#)[↗]. Retrieved 2010-09-30.
63. [^] ["Defining Document Compatibility: Specifying Document Compatibility Modes"](#)[↗]. 2011-04-01. Retrieved 2012-01-24.
64. [^] Eric Lawrence (2008-07-02). ["IE8 Security Part IV: The XSS Filter"](#)[↗]. Retrieved 2010-09-30.
65. [^] ["Hypertext Transfer Protocol \(HTTP/1.1\): Caching"](#)[↗]. ietf.org. Retrieved 2014-07-24.
66. [^] ["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 7239](#)[↗]: Forwarded HTTP Extension
- [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 headers](#) | [Internet-related lists](#)

This page was last edited on 22 February 2019, at 05:30 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#) [Mobile view](#)

