WIKIPEDIA

# 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.

## Contents



## General format

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

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 (http://www.iana.org/assignments/message-headers/message-headers.xml#perm-headers) and repository of provisional registrations (http://www.iana.org/assignments/message-headers/message-headers.xml#prov-headers) are maintained by the IANA. Additional field names and permissible values may be defined by each application.

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

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

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

### Standard request fields

| Name | Description | Example | Status | Standard |
|---|---|---|---|---|
| A-IM | Acceptable instance-manipulations for the request.[10] | `A-IM: feed` | Permanent | RFC 3229 (https://tools.ietf.org/html/rfc32 |
| Accept | Media type(s) that is/are acceptable for the response. See Content negotiation. | `Accept: text/html` | Permanent | RFC 2616 (https://tools.ietf.org/html/rfc26 7231 (https://tools.ietf.org/html/rfc72 |
| Accept-Charset | Character sets that are acceptable. | `Accept-Charset: utf-8` | Permanent | RFC 2616 (https://tools.ietf.org/html/rfc26 |
| Accept-Datetime | Acceptable version in time. | `Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT` | Provisional | RFC 7089 (https://tools.ietf.org/html/rfc70 |
| Accept-Encoding | List of acceptable encodings. See HTTP compression. | `Accept-Encoding: gzip, deflate` | Permanent | RFC 2616 (https://tools.ietf.org/html/rfc26 7231 (https://tools.ietf.org/html/rfc72 |
| Accept-Language | List of acceptable human languages for response. See Content negotiation. | `Accept-Language: en-US` | Permanent | RFC 2616 (https://tools.ietf.org/html/rfc26 7231 (https://tools.ietf.org/html/rfc72 |
| 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 *must* be obeyed by all caching mechanisms along the request-response chain. | `Cache-Control: no-cache` | Permanent | |
| 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 (http://tools.ietf.org/html/rfc7231#section-7.1.1.1)). | `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 | |
| 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 *304 Not Modified* to be returned if content is unchanged. | `If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT` | Permanent | |
| If-None-Match | Allows a *304 Not Modified* 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 | | `Max-Forwards: 10` | Permanent | |

| Name | Description | Example | Status | Standard |
|---|---|---|---|---|
| | Limit the number of times the message can be forwarded through proxies or gateways. | | | |
| 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 [*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` | Permanent | |
| 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.<br>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.<br>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

| 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`<br>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)<br>`DNT: 0` (Do Not Track Disabled) |
| X-Forwarded-For[25] | 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. Superseded by *Forwarded* header. | `X-Forwarded-For: client1, proxy1, proxy2`<br>`X-Forwarded-For: 129.78.138.66, 129.78.64.103` |
| X-Forwarded-Host[26] | 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. Superseded by *Forwarded* header. | `X-Forwarded-Host: en.wikipedia.org:8080`<br>`X-Forwarded-Host: en.wikipedia.org` |
| X-Forwarded-Proto[27] | 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. Superseded by *Forwarded* 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.<br>Must not be used with HTTP/2.[13] | `Proxy-Connection: keep-alive` |
| X-UIDH[33][34][35] | 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[36] | Used to prevent cross-site request forgery. Alternative header names are: `X-CSRFToken`[37] and `X-XSRF-TOKEN`[38] | `X-Csrf-Token: i8XNjC4b8KVok4uw5RftR38Wgp2BFwql` |
| X-Request-ID[39][40], X-Correlation-ID[41][42] | Correlates HTTP requests between a client and server. | `X-Request-ID: f058ebd6-02f7-4d3f-942e-904344e8cde5` |
| 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. | `Save-Data: on` |

## Response fields

### Standard response fields

| Field name | Description | Example | Status | Standard |
|---|---|---|---|---|
| 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 | `Access-Control-Allow-Origin: *` | Permanent: standard | |
| Accept-Patch[43] | Specifies which patch document formats this server supports | `Accept-Patch: text/example;charset=utf-8` | Permanent | |
| Accept-Ranges | What partial content range types this server supports via byte serving | `Accept-Ranges: bytes` | Permanent | |
| Age | The age the object has been in a proxy cache in seconds | `Age: 12` | Permanent | |
| Allow | Valid methods for a specified resource. To be used for a *405 Method not allowed* | `Allow: GET, HEAD` | Permanent | |
| 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<br>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]<br>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: Q2hlY2sgSW50ZWdyaXR5IQ==` | Obsolete[14] | |
| 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. | ▪ Example 1: `Location: http://www.w3.org/pub/WWW/People.html`<br>▪ Example 2: `Location: /pub/WWW/People.html` | Permanent | |
| P3P | This field is supposed to set P3P policy, in the form of `P3P:CP="your_compact_policy"`. However, P3P did not take off,[50] most browsers have never fully | `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 | Standard |
|---|---|---|---|---|
| | 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. | | | |
| 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] | ▪ Example 1: `Retry-After: 120`<br>▪ 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 (http://www.iana.org/assignments/http-parameters) are: chunked, compress, deflate, gzip, identity.<br>Must not be used with HTTP/2.[13] | `Transfer-Encoding: chunked` | Permanent | |
| Tk | Tracking Status header, value suggested to be sent in response to a DNT(do-not-track), possible values:<br>`"!" – under construction`<br>`"?" – dynamic`<br>`"G" – gateway to multiple parties`<br>`"N" – not tracking`<br>`"T" – tracking`<br>`"C" – tracking with consent`<br>`"P" – tracking only if consented`<br>`"D" – disregarding DNT`<br>`"U" – updated` | `Tk: ?` | Permanent | |
| Upgrade | Ask the client to upgrade to another protocol.<br>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. | ▪ Example 1: `Vary: *`<br>▪ 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 | |
| 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**

| 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 `Timing-Allow-Origin` response header specifies origins that are allowed to see values of attributes retrieved via features of the Resource Timing API (https://developer.mozilla.org/en-US/docs/Web/API/Resource_Timing_API), which would otherwise be reported as zero due to cross-origin restrictions.[57] | `Timing-Allow-Origin: *`<br>`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 `X-Runtime`, `X-Version`, or `X-AspNet-Version`) | `X-Powered-By: PHP/5.4.0` |
| 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`<br>`X-UA-Compatible: IE=edge`<br>`X-UA-Compatible: Chrome=1` |
| X-XSS-Protection[64] | Cross-site scripting (XSS) filter | `X-XSS-Protection: 1; mode=block` |

## Effects of selected fields

### 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 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

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

## References


1. "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" (http://tools.ietf.org/html/rfc7230#section-3.2.4). ietf.org. Retrieved July 23, 2014.
2. RFC-7230 section 3.2 (https://tools.ietf.org/html/rfc7230#section-3.2)
3. RFC-7210 section 3.1.1 (https://tools.ietf.org/html/rfc7230#section-3.1.1)
4. RFC-7231 section 4.1 (https://tools.ietf.org/html/rfc7231#section-4.1)
5. Internet Engineering Task Force (June 1, 2012). "RFC 6648" (http://tools.ietf.org/html/rfc6648). Retrieved November 12, 2012.
6. "Message Headers" (http://www.iana.org/assignments/message-headers/message-headers.xml). Iana.org. June 11, 2014. Retrieved June 12, 2014.
7. "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" (http://tools.ietf.org/html/rfc7230#section-3.2.6). itef.org. Retrieved July 24, 2014.
8. "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content" (http://tools.ietf.org/html/rfc7231#section-5.3.1). ietf.org. Retrieved July 24, 2014.
9. "core - Apache HTTP Server" (https://web.archive.org/web/20120509104709/https://httpd.apache.org/docs/2.3/mod/core Httpd.apache.org. Archived from the original (http://httpd.apache.org/docs/2.3/mod/core.html#limitrequestfieldsize) on May 9, 2012. Retrieved March 13, 2012.
10. *RFC 3229* (https://tools.ietf.org/html/rfc3229). doi:10.17487/RFC3229 (https://doi.org/10.17487%2FRFC3229).
11. "Cross-Origin Resource Sharing" (https://www.w3.org/TR/cors/). Retrieved July 24, 2017.
12. "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" (http://tools.ietf.org/html/rfc7230#section-6.1). IETF. June 2014. Retrieved December 19, 2014.
13. "Hypertext Transfer Protocol Version 2 (HTTP/2)" (https://tools.ietf.org/html/rfc7540#section-8.1.2.2). IETF. May 2015. Retrieved June 6, 2017.
14. "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content" (http://tools.ietf.org/html/rfc7231#appendix-B). Retrieved June 3, 2015.
15. "Forwarded HTTP Extension: Introduction" (http://tools.ietf.org/html/rfc7239#section-1). IETF. June 2014. Retrieved January 7, 2016.

16. "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" (http://tools.ietf.org/html/rfc7230#section-5.4). IETF. June 2014. Retrieved July 24, 2014.
17. "Hypertext Transfer Protocol Version 2 (HTTP/2)" (https://tools.ietf.org/html/rfc7540#section-8.1.2.3). IETF. May 2015. Retrieved June 6, 2017.
18. "Message Headers" (https://www.iana.org/assignments/message-headers/message-headers.xml). *www.iana.org*. Retrieved November 26, 2018.
19. "Hypertext Transfer Protocol Version 2 (HTTP/2)" (https://httpwg.org/specs/rfc7540.html#Http2SettingsHeader). *httpwg.org*. May 30, 2015. Retrieved February 22, 2019.
20. "Upgrade Insecure Requests - W3C Candidate Recommendation" (https://www.w3.org/TR/upgrade-insecure-requests/#preference). *W3C*. October 8, 2015. Retrieved January 14, 2016.
21. "Try out the "Do Not Track" HTTP header" (http://blog.sidstamm.com/2011/01/try-out-do-not-track-http-header.html). Retrieved January 31, 2011.
22. "Web Tracking Protection: Minimum Standards and Opportunities to Innovate" (http://blogs.msdn.com/b/ie/archive/2011/03/14/web-tracking-protection-minimum-standards-and-opportunities-to-innovate.aspx). Retrieved March 24, 2011.
23. IETF Do Not Track: A Universal Third-Party Web Tracking Opt Out (http://tools.ietf.org/html/draft-mayer-do-not-track-00) March 7, 2011
24. W3C Tracking Preference Expression (DNT) (http://www.w3.org/2011/tracking-protection/drafts/tracking-dnt.html), January 26, 2012
25. Amos Jeffries (July 2, 2010). "SquidFaq/ConfiguringSquid - Squid Web Proxy Wiki" (http://wiki.squid-cache.org/SquidFaq/ConfiguringSquid#head-3518b69c63e221cc3cd7885415e365ffaf3dd27f). Retrieved September 10, 2009.
26. The Apache Software Foundation. "mod_proxy - Apache HTTP Server Version 2.2" (http://httpd.apache.org/docs/2.2/mod/mod_proxy.html#x-headers). Retrieved November 12, 2014.
27. Dave Steinberg (April 10, 2007). "How do I adjust my SSL site to work with GeekISP's loadbalancer?" (http://www.geekisp.com/faq/6_65_en.html). Retrieved September 30, 2010.
28. "Helping to Secure Communication: Client to Front-End Server" (https://technet.microsoft.com/en-us/library/aa997519(v=exchg.65).aspx). July 27, 2006. Retrieved April 23, 2012.
29. "OpenSocial Core API Server Specification 2.5.1" (https://opensocial.github.io/spec/2.5.1/Core-API-Server.xml#rfc.section.2.1.1.1). Retrieved October 8, 2014.
30. "ATT Device ID" (http://developer.att.com/developer/forward.jsp?passedItemId=5300270). Retrieved January 14, 2012.
31. "WAP Profile" (http://www.developershome.com/wap/detection/detection.asp?page=profileHeader). Retrieved January 14, 2012.
32. de Boyne Pollard, Jonathan (2007). "The Proxy-Connection: header is a mistake in how some web browsers use HTTP" (https://jdebp.eu/FGA/web-proxy-connection-header.html). Retrieved January 16, 2018.
33. "Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls" (https://www.eff.org/deeplinks/2014/11/verizon-x-uidh). Electronic Frontier Foundation. Retrieved January 19, 2014.
34. "Checking known AT&T, Verizon, Sprint, Bell Canada & Vodacom Unique Identifier beacons" (http://lessonslearned.org/sniff). Retrieved January 19, 2014.
35. Craig Timberg. "Verizon, AT&T tracking their users with 'supercookies' " (https://www.washingtonpost.com/business/technology/verizon-atandt-tracking-their-users-with-super-cookies/2014/11/03/7bbbf382-6395-11e4-bb14-4cfea1e742d5_story.html). The Washington Post. Retrieved January 19, 2014.
36. "SAP Cross-Site Request Forgery Protection" (https://help.sap.com/saphelp_nw74/helpdata/en/b3/5c22518bc72214e10000000a4417 SAP SE. Retrieved January 20, 2015.
37. "Django Cross Site Request Forgery protection" (https://docs.djangoproject.com/en/1.7/ref/contrib/csrf/). Django (web framework). Retrieved January 20, 2015.
38. "Angular Cross Site Request Forgery (XSRF) Protection" (https://docs.angularjs.org/api/ng/service/$http#cross-site-request-forgery-xsrf-protection). AngularJS. Retrieved January 20, 2015.
39. "What is the X-REQUEST-ID http header?" (https://stackoverflow.com/questions/25433258/what-is-the-x-request-id-http-header). *stackoverflow.com*. Retrieved May 19, 2016.
40. "HTTP Request IDs" (https://devcenter.heroku.com/articles/http-request-id). *devcenter.heroku.com*. Retrieved February 6, 2018.
41. "The Value of Correlation IDs" (https://blog.rapid7.com/2016/12/23/the-value-of-correlation-ids/). *Rapid7 Blog*. December 23, 2016. Retrieved April 13, 2018.
42. Hilton, Peter. "Correlation IDs for microservices architectures - Peter Hilton" (http://hilton.org.uk/blog/microservices-correlation-id). *hilton.org.uk*. Retrieved April 13, 2018.
43. "RFC 5789" (http://tools.ietf.org/html/rfc5789#section-3.1). Retrieved December 24, 2014.
44. "HTTP Alternative Services" (https://tools.ietf.org/html/rfc7838). IETF. April 2016. Retrieved April 19, 2016.
45. "HTTP Alternative Services, section 3" (https://tools.ietf.org/html/rfc7838#section-3). IETF. April 2016. Retrieved June 8, 2017.
46. "RFC 6266" (http://tools.ietf.org/html/rfc6266). Retrieved March 13, 2015.
47. "RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content" (https://tools.ietf.org/html/rfc7231#section-3.1.3.2). Tools.ietf.org. Retrieved December 11, 2017.
48. "RFC7231 Compliant HTTP Date Headers" (http://ronwilliams.io/blog/post/rfc7231-compliant-http-date-headers).
49. Indicate the canonical version of a URL by responding with the Link rel="canonical" HTTP header (https://support.google.com/webmasters/bin/answer.py?hl=en&answer=139394) Retrieved: 2012-02-09
50. W3C P3P Work Suspended (http://www.w3.org/P3P)
51. "Public Key Pinning Extension for HTTP" (http://www.rfc-editor.org/rfc/rfc7469.txt). IETF. Retrieved April 17, 2015.
52. "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content" (http://tools.ietf.org/html/rfc7231#section-7.1.3). Retrieved July 24, 2014.
53. "HTTP Header Field X-Frame-Options" (https://tools.ietf.org/html/rfc7034). IETF. 2013. Retrieved June 12, 2014.
54. "Content Security Policy Level 2" (http://www.w3.org/TR/CSP11/#frame-ancestors-and-frame-options). Retrieved August 2, 2014.
55. "Content Security Policy" (http://www.w3.org/TR/CSP/). W3C. 2012. Retrieved April 28, 2017.
56. "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" (http://tools.ietf.org/html/rfc7230#section-3.1.2). Retrieved July 24, 2014.
57. "Timing-Allow-Origin" (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Timing-Allow-Origin). *Mozilla Developer Network*. Retrieved January 25, 2018.
58. "Configuring servers for Ogg media" (https://developer.mozilla.org/en-US/docs/Web/HTTP/Configuring_servers_for_Ogg_media#Serve_X-Content-Duration_headers). May 26, 2014. Retrieved January 3, 2015.
59. Eric Lawrence (September 3, 2008). "IE8 Security Part VI: Beta 2 Update" (http://blogs.msdn.com/b/ie/archive/2008/09/02/ie8-security-part-vi-beta-2-update.aspx). Retrieved September 28, 2010.
60. "Hosting - Google Chrome Extensions - Google Code" (https://code.google.com/chrome/extensions/hosting.html). Retrieved June 14, 2012.
61. van Kesteren, Anne (August 26, 2016). "Fetch standard" (https://fetch.spec.whatwg.org/#x-content-type-options-header). *WHATWG*. Archived (https://web.archive.org/web/20160826132911/https://fetch.spec.whatwg.org/#x-content-type-options-header) from the original on August 26, 2016. Retrieved August 26, 2016.
62. "Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow" (https://stackoverflow.com/questions/1288338/why-does-asp-net-framework-add-the-x-powered-byasp-net-http-header-in-response). Retrieved September 30, 2010.
63. "Defining Document Compatibility: Specifying Document Compatibility Modes" (http://msdn.microsoft.com/en-us/library/ie/cc288325%28v=vs.85%29.aspx#SetMode). April 1, 2011. Retrieved January 24, 2012.
64. Eric Lawrence (July 2, 2008). "IE8 Security Part IV: The XSS Filter" (http://blogs.msdn.com/b/ie/archive/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx). Retrieved September 30, 2010.
65. "Hypertext Transfer Protocol (HTTP/1.1): Caching" (http://tools.ietf.org/html/rfc7234#section-5.4). ietf.org. Retrieved July 24, 2014.
66. "How to prevent caching in Internet Explorer" (https://support.microsoft.com/en-us/kb/234067/). Microsoft. September 22, 2011. Retrieved April 15, 2015.


## External links

- Headers: Permanent Message Header Field Names (http://www.iana.org/assignments/message-headers/message-headers.xml#perm-headers%7CMessage)
- 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 (http://www.and.org/texts/server-http)
- Internet Explorer and Custom HTTP Headers - EricLaw's IEInternals - Site Home - MSDN Blogs (http://blogs.msdn.com/b/ieinternals/archive/2009/06/30/internet-explorer-custom-http-headers.aspx)