HTTP headers

Jump to section

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (`:`), then by its value. Whitespace before the value is ignored.

Custom proprietary headers have historically been used with an `X-` prefix, but this convention was deprecated in June 2012 because of the inconveniences it caused when nonstandard fields became standard in RFC 6648; others are listed in an IANA registry, whose original content was defined in RFC 4229. IANA also maintains a registry of proposed new HTTP headers.

Headers can be grouped according to their contexts:

- General headers apply to both requests and responses, but with no relation to the data transmitted in the body.
- Request headers contain more information about the resource to be fetched, or about the client requesting the resource.
- Response headers hold additional information about the response, like its location or about the server providing it.
- Entity headers contain information about the body of the resource, like its content length or MIME type.

Headers can also be grouped according to how proxies handle them:

- `Connection`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `TE`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade` (see also Protocol upgrade mechanism).

End-to-end headers

These headers *must* be transmitted to the final recipient of the message: the server for a request, or the client for a response. Intermediate proxies must retransmit these headers unmodified and caches must store them.

Hop-by-hop headers

These headers are meaningful only for a single transport-level connection, and *must not* be retransmitted by proxies or cached. Note that only hop-by-hop headers may be set using the `Connection` general header.

Authentication

`WWW-Authenticate`

Defines the authentication method that should be used to access a resource.

`Authorization`

Contains the credentials to authenticate a user-agent with a server.

`Proxy-Authenticate`

Defines the authentication method that should be used to access a resource behind a proxy server.

`Proxy-Authorization`

Contains the credentials to authenticate a user agent with a proxy server.

Caching

`Age`

The time, in seconds, that the object has been in a proxy cache.

`Cache-Control`

Directives for caching mechanisms in both requests and responses.

`Clear-Site-Data`

Clears browsing data (e.g. cookies, storage, cache) associated with the requesting website.

`Expires`

The date/time after which the response is considered stale.

`Pragma`

Implementation-specific header that may have various effects anywhere along the request-response chain. Used for backwards compatibility with HTTP/1.0 caches where the `Cache-Control` header is not yet present.

`Warning`

General warning information about possible problems.

Client hints

HTTP Client hints are a work in progress. Actual documentation can be found on the website of the HTTP working group.

`Accept-CH`

Servers can advertise support for Client Hints using the `Accept-CH` header field or an equivalent HTML `<meta>` element with `http-equiv` attribute ([HTML5]).

`Accept-CH-Lifetime`

Servers can ask the client to remember the set of Client Hints that the server supports for a specified period of time, to enable delivery of Client Hints on subsequent requests to the server's origin ([RFC6454]).

`Early-Data`

Indicates that the request has been conveyed in early data.

`Content-DPR`

A number that indicates the ratio between physical pixels over CSS pixels of the selected image response.

`DPR`

A number that indicates the client's current Device Pixel Ratio (DPR), which is the ratio of

A number that indicates the client's current Device Pixel Ratio (DPR), which is the ratio of physical pixels over CSS pixels (Section 5.2 of [CSSVAL]) of the layout viewport (Section 9.1.1 of [CSS2]) on the device.

`Device-Memory`

Technically a part of Device Memory API, this header represents an approximate amount of RAM client has.

`Save-Data`

A boolean that indicates the user agent's preference for reduced data usage.

`Viewport-Width`

A number that indicates the layout viewport width in CSS pixels. The provided pixel value is a number rounded to the smallest following integer (i.e. ceiling value).

If `Viewport-Width` occurs in a message more than once, the last value overrides all previous occurrences.

`Width`

The `Width` request header field is a number that indicates the desired resource width in physical pixels (i.e. intrinsic size of an image). The provided pixel value is a number rounded to the smallest following integer (i.e. ceiling value).

If the desired resource width is not known at the time of the request or the resource does not have a display width, the `Width` header field can be omitted. If `Width` occurs in a message more than once, the last value overrides all previous occurrences

Conditionals

`Last-Modified`

The last modification date of the resource, used to compare several versions of the same resource. It is less accurate than `ETag`, but easier to calculate in some environments. Conditional requests using `If-Modified-Since` and `If-Unmodified-Since` use this value to change the behavior of the request.

`ETag`

A unique string identifying the version of the resource. Conditional requests using `If-Match` and `If-None-Match` use this value to change the behavior of the request.

`If-Match`

Makes the request conditional, and applies the method only if the stored resource matches one of the given ETags.

`If-None-Match`

Makes the request conditional, and applies the method only if the stored resource *doesn't* match any of the given ETags. This is used to update caches (for safe requests), or to prevent to upload a new resource when one already exists.

`If-Modified-Since`

Makes the request conditional, and expects the entity to be transmitted only if it has been modified after the given date. This is used to transmit data only when the cache is out of date.

`If-Unmodified-Since`

Makes the request conditional, and expects the entity to be transmitted only if it has not been modified after the given date. This ensures the coherence of a new fragment of a specific range with previous ones, or to implement an optimistic concurrency control system when modifying existing documents.

`Vary`

Determines how to match request headers to decide whether a cached response can be used rather than requesting a fresh one from the origin server.

Connection management

`Connection`

Controls whether the network connection stays open after the current transaction finishes.

`Keep-Alive`

Controls how long a persistent connection should stay open.

Content negotiation

`Accept`

Informs the server about the types of data that can be sent back.

`Accept-Charset`

Which character encodings the client understands.

`Accept-Encoding`

The encoding algorithm, usually a compression algorithm, that can be used on the resource sent back.

`Accept-Language`

Informs the server about the human language the server is expected to send back. This is a hint and is not necessarily under the full control of the user: the server should always pay attention not to override an explicit user choice (like selecting a language from a dropdown).

Controls

`Expect`

Indicates expectations that need to be fulfilled by the server to properly handle the request.

`Max-Forwards`

Cookies

`Cookie`

Contains stored HTTP cookies previously sent by the server with the `Set-Cookie` header.

`Set-Cookie`

Send cookies from the server to the user-agent.

`Cookie2`

Contains an HTTP cookie previously sent by the server with the `Set-Cookie2` header, but has been **obsoleted**. Use `Cookie` instead.

`Set-Cookie2`

Sends cookies from the server to the user-agent, but has been **obsoleted**. Use `Set-Cookie` instead.

CORS

Learn more about CORS here.

`Access-Control-Allow-Origin`

Indicates whether the response can be shared.

`Access-Control-Allow-Credentials`

Indicates whether the response to the request can be exposed when the credentials flag is true.

`Access-Control-Allow-Headers`

Used in response to a preflight request to indicate which HTTP headers can be used when making the actual request.

`Access-Control-Allow-Methods`

Specifies the methods allowed when accessing the resource in response to a preflight request.

`Access-Control-Expose-Headers`

Indicates which headers can be exposed as part of the response by listing their names.

`Access-Control-Max-Age`

`Access-Control-Max-Age`

Indicates how long the results of a preflight request can be cached.

`Access-Control-Request-Headers`

Used when issuing a preflight request to let the server know which HTTP headers will be used when the actual request is made.

`Access-Control-Request-Method`

Used when issuing a preflight request to let the server know which HTTP method will be used when the actual request is made.

`Origin`

Indicates where a fetch originates from.

`Timing-Allow-Origin`

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.

Do Not Track

`DNT`

Expresses the user's tracking preference.

`Tk`

Indicates the tracking status of the corresponding response.

Downloads

`Content-Disposition`

Indicates if the resource transmitted should be displayed inline (default behavior without the header), or if it should be handled like a download and the browser should present a “Save As” dialog.

Message body information

`Content-Length`

The size of the resource, in decimal number of bytes.

`Content-Type`

Indicates the media type of the resource.

`Content-Encoding`

Used to specify the compression algorithm.

`Content-Language`

Describes the human language(s) intended for the audience, so that it allows a user to differentiate according to the users' own preferred language.

`Content-Location`

Indicates an alternate location for the returned data.

Proxies

`Forwarded`

Contains information from the client-facing side of proxy servers that is altered or lost when a proxy is involved in the path of the request.

`X-Forwarded-For` ⚠

Identifies the originating IP addresses of a client connecting to a web server through an HTTP proxy or a load balancer.

`X-Forwarded-Host` ⚠

Identifies the original host requested that a client used to connect to your proxy or load balancer.

`X-Forwarded-Proto` ⚠

Identifies the protocol (HTTP or HTTPS) that a client used to connect to your proxy or load balancer.

`Via`

Added by proxies, both forward and reverse proxies, and can appear in the request headers and the response headers.

Redirects

`Location`

Indicates the URL to redirect a page to.

Request context

`From`

Contains an Internet email address for a human user who controls the requesting user agent.

`Host`

Specifies the domain name of the server (for virtual hosting), and (optionally) the TCP port number on which the server is listening.

number on which the server is listening.

`Referer`
: The address of the previous web page from which a link to the currently requested page was followed.

`Referrer-Policy`
: Governs which referrer information sent in the `Referer` header should be included with requests made.

`User-Agent`
: Contains a characteristic string that allows the network protocol peers to identify the application type, operating system, software vendor or software version of the requesting software user agent. See also the Firefox user agent string reference.

Response context

`Allow`
: Lists the set of HTTP request methods supported by a resource.

`Server`
: Contains information about the software used by the origin server to handle the request.

Range requests

`Accept-Ranges`
: Indicates if the server supports range requests, and if so in which unit the range can be expressed.

`Range`
: Indicates the part of a document that the server should return.

`If-Range`
: Creates a conditional range request that is only fulfilled if the given etag or date matches the remote resource. Used to prevent downloading two ranges from incompatible version of the resource.

`Content-Range`
: Indicates where in a full body message a partial message belongs.

Security

`Cross-Origin-Embedder-Policy` (**COEP**)

Allows a server to declare an embedder policy for a given document.

`Cross-Origin-Opener-Policy` (**COOP**)

Prevents other domains from opening/controlling a window.

`Cross-Origin-Resource-Policy` (**CORP**)

Prevents other domains from reading the response of the resources to which this header is applied.

`Content-Security-Policy` (**CSP**)

Controls resources the user agent is allowed to load for a given page.

`Content-Security-Policy-Report-Only`

Allows web developers to experiment with policies by monitoring, but not enforcing, their effects. These violation reports consist of JSON documents sent via an HTTP `POST` request to the specified URI.

`Expect-CT`

Allows sites to opt in to reporting and/or enforcement of Certificate Transparency requirements, which prevents the use of misissued certificates for that site from going unnoticed. When a site enables the Expect-CT header, they are requesting that Chrome check that any certificate for that site appears in public CT logs.

`Feature-Policy`

Provides a mechanism to allow and deny the use of browser features in its own frame, and

in iframes that it embeds.

`Strict-Transport-Security` (**HSTS**)

Force communication using HTTPS instead of HTTP.

`Upgrade-Insecure-Requests`

Sends a signal to the server expressing the client's preference for an encrypted and authenticated response, and that it can successfully handle the `upgrade-insecure-requests` directive.

`X-Content-Type-Options`

Disables MIME sniffing and forces browser to use the type given in `Content-Type`.

`X-Download-Options`

The `X-Download-Options` HTTP header indicates that the browser (Internet Explorer) should not display the option to "Open" a file that has been downloaded from an application, to prevent phishing attacks as the file otherwise would gain access to execute in the context of the application. (Note: related MS Edge bug).

`X-Frame-Options` (**XFO**)

Indicates whether a browser should be allowed to render a page in a `<frame>`, `<iframe>`, `<embed>` or `<object>`.

`X-Permitted-Cross-Domain-Policies`

Specifies if a cross-domain policy file (`crossdomain.xml`) is allowed. The file may define a policy to grant clients, such as Adobe's Flash Player, Adobe Acrobat, Microsoft Silverlight,

or Apache Flex, permission to handle data across domains that would otherwise be restricted due to the Same-Origin Policy. See the Cross-domain Policy File Specification for more information.

`X-Powered-By`

May be set by hosting environments or other frameworks and contains information about them while not providing any usefulness to the application or its visitors. Unset this header to avoid exposing potential vulnerabilities.

`X-XSS-Protection`

Enables cross-site scripting filtering.

HTTP Public Key Pinning (HPKP)

HTTP Public Key Pinning has been deprecated and removed in favor of Certificate Transparency and `Expect-CT`.

`Public-Key-Pins`

Associates a specific cryptographic public key with a certain web server to decrease the risk of MITM attacks with forged certificates.

`Public-Key-Pins-Report-Only`

Sends reports to the report-uri specified in the header and does still allow clients to connect to the server even if the pinning is violated.

Fetch metadata request headers

`Sec-Fetch-Site`

It is a request header that indicates the relationship between a request initiator's origin and its target's origin. It is a Structured Header whose value is a token with possible values `cross-site`, `same-origin`, `same-site`, and `none`.

`Sec-Fetch-Mode`

It is a request header that indicates the request's mode to a server. It is a Structured Header whose value is a token with possible values `cors`, `navigate`, `nested-navigate`, `no-cors`, `same-origin`, and `websocket`.

`Sec-Fetch-User`

It is a request header that indicates whether or not a navigation request was triggered by user activation. It is a Structured Header whose value is a boolean so possible values are `?0` for false and `?1` for true.

`Sec-Fetch-Dest`

It is a request header that indicates the request's destination to a server. It is a Structured Header whose value is a token with possible values `audio`, `audioworklet`, `document`, `embed`, `empty`, `font`, `image`, `manifest`, `object`, `paintworklet`, `report`, `script`, `serviceworker`, `sharedworker`, `style`, `track`, `video`, `worker`, `xslt`, and `nested-document`.

Server-sent events

`Last-Event-ID`

...

`NEL` 

Defines a mechanism that enables developers to declare a network error reporting policy.

`Ping-From`

...

`Ping-To`

...

`Report-To`

Used to specify a server endpoint for the browser to send warning and error reports to.

Transfer coding

`Transfer-Encoding`

Specifies the form of encoding used to safely transfer the entity to the user.

`TE`

Specifies the transfer encodings the user agent is willing to accept.

`Trailer`

Allows the sender to include additional fields at the end of chunked message.

WebSockets

`Sec-WebSocket-Key`

...

`Sec-WebSocket-Extensions`

...

`Sec-WebSocket-Accept`

...

`Sec-WebSocket-Protocol`

...

`Sec-WebSocket-Version`

...

Other

`Accept-Push-Policy`

A client can express the desired push policy for a request by sending an `Accept-Push-Policy` header field in the request.

`Accept-Signature`

A client can send the `Accept-Signature` header field to indicate intention to take advantage of any available signatures and to indicate what kinds of signatures it supports.

`Alt-Svc`

Used to list alternate ways to reach this service.

`Date`

Contains the date and time at which the message was originated.

`Large-Allocation`

Tells the browser that the page being loaded is going to want to perform a large allocation.

`Link`

The `Link` entity-header field provides a means for serialising one or more links in HTTP headers. It is semantically equivalent to the HTML `<link>` element.

`Push-Policy` 🧪

A `Push-Policy` defines the server behaviour regarding push when processing a request.

`Retry-After`

Indicates how long the user agent should wait before making a follow-up request.

`Signature` 🧪

The `Signature` header field conveys a list of signatures for an exchange, each one accompanied by information about how to determine the authority of and refresh that signature.

`Signed-Headers` 🧪

The `Signed-Headers` header field identifies an ordered list of response header fields to include in a signature.

`Server-Timing`

Communicates one or more metrics and descriptions for the given request-response cycle.

`Service-Worker-Allowed`

Used to remove the path restriction by including this header in the response of the Service Worker script.

`SourceMap`

Links generated code to a source map.

`Upgrade`

The relevant RFC document for the Upgrade header field is RFC 7230, section 6.7. The standard establishes rules for upgrading or changing to a different protocol on the current client, server, transport protocol connection. For example, this header standard allows a client to change from HTTP 1.1 to HTTP 2.0, assuming the server decides to acknowledge and implement the Upgrade header field. Neither party is required to accept the terms specified in the Upgrade header field. It can be used in both client and server headers. If the Upgrade header field is specified, then the sender MUST also send the Connection header field with the upgrade option specified. For details on the Connection header field please see section 6.1 of the aforementioned RFC.

`X-DNS-Prefetch-Control`

Controls DNS prefetching, a feature by which browsers proactively perform domain name resolution on both links that the user may choose to follow as well as URLs for items referenced by the document, including images, CSS, JavaScript, and so forth.

`X-Firefox-Spdy` 👎 ⚠

...

`X-Pingback` ⚠

...

`X-Requested-With`

...

`X-Robots-Tag` ⚠

The `X-Robots-Tag` HTTP header is used to indicate how a web page is to be indexed within public search engine results. The header is effectively equivalent to `<meta name="robots" content="...">`.

`X-UA-Compatible` ⚠

Used by Internet Explorer to signal which document mode to use.

Contributing

You can help by writing new entries or improving the existing ones.

See also

- Wikipedia page on List of HTTP headers
- IANA registry
- HTTP Working Group

Last modified: Apr 27, 2020, by MDN contributors

Related Topics

HTTP

Guides:

- Resources and URIs
- HTTP guide
- HTTP security

HTTP access control (CORS)

HTTP authentication

HTTP caching

HTTP compression

HTTP conditional requests

HTTP content negotiation

HTTP cookies

HTTP range requests

HTTP redirects

HTTP specifications

Feature policy

References:

- HTTP headers

  - `Accept`
  - `Accept-CH`
  - `Accept-CH-Lifetime`
  - `Accept-Charset`
  - `Accept-Encoding`
  - `Accept-Language`
  - `Accept-Patch`
  - `Accept-Ranges`
  - `Access-Control-Allow-Credentials`
  - `Access-Control-Allow-Headers`
  - `Access-Control-Allow-Methods`
  - `Access-Control-Allow-Origin`
  - `Access-Control-Expose-Headers`
  - `Access-Control-Max-Age`

Access-Control-Max-Age

Access-Control-Request-Headers

Access-Control-Request-Method

Age

Allow

Alt-Svc

Authorization

Cache-Control

Clear-Site-Data

Connection

Content-Disposition

Content-Encoding

Content-Language

Content-Length

Content-Location

Content-Range

Content-Security-Policy

Content-Security-Policy-Report-Only

Content-Type

Cookie

Cookie2

Cross-Origin-Embedder-Policy

Cross-Origin-Opener-Policy

Cross-Origin-Resource-Policy

DNT

DPR

Date

Device-Memory

Digest

ETag

Early-Data

Expect

Expect-CT

Expires

Feature-Policy

Forwarded

From

Host

If-Match

If-Modified-Since

If-None-Match

If-Range

If-Unmodified-Since

Index

Keep-Alive

⚠ Large-Allocation

Last-Modified

Link

Location

NEL

Origin

👎 Pragma

Proxy-Authenticate

Proxy-Authorization

👎 🗑 Public-Key-Pins

👎 🗑 Public-Key-Pins-Report-Only

Range

Referer

Referrer-Policy

Retry-After

Save-Data

Sec-Fetch-Dest

Sec-Fetch-Mode

Sec-Fetch-Site

Sec-Fetch-User

Sec-WebSocket-Accept

Server

Server-Timing

Set-Cookie

Set-Cookie2

SourceMap

Strict-Transport-Security

TE

Timing-Allow-Origin

Tk

Trailer

Transfer-Encoding

Upgrade-Insecure-Requests

User-Agent

Vary

Via

WWW-Authenticate

Want-Digest

Warning

X-Content-Type-Options

- `X-DNS-Prefetch-Control`
- ⚠ `X-Forwarded-For`
- ⚠ `X-Forwarded-Host`
- ⚠ `X-Forwarded-Proto`
- `X-Frame-Options`
- `X-XSS-Protection`

- ▶ HTTP request methods
- ▶ HTTP response status codes
- ▶ CSP directives
- ▶ CORS errors
- ▶ Feature-Policy directives

Get the latest and greatest from MDN delivered straight to your inbox.

you@example.com

Sign up now

MDN web docs
moz://a

Web Technologies

Learn Web Development

About MDN

Feedback

MDN

About

MDN Web Docs Store

Contact Us

Firefox

Mozilla