Liste der HTTP-Headerfelder

HTTP-Header-Felder (oft ungenau **HTTP-Header**) sind Bestandteile des Hypertext Transfer Protocol (HTTP)-Protokollheaders und übermitteln die für die Übertragung von Dateien über HTTP wichtigen Parameter und Argumente, z. B. gewünschte Sprache oder Zeichensatz sowie oft Informationen über den Client. Oft wird „HTTP-Header“ synonym genutzt, besitzt allerdings die Mehrdeutigkeit zwischen einem einzelnen Feld des Headerblocks und dem ganzen Headerblock. Hier wird für die Gesamtheit der Headerfelder der Begriff „Header“ und für eine einzelne Zeile im Header der Begriff „Headerfeld“ entsprechend RFC 2616 genutzt.

Die einzelnen Felder im Header werden immer nach der Anfrage-(Request)-Zeile (z. B. `GET /index.html HTTP/1.1`) bzw. der Antwort-(Response)-Zeile (bei Erfolg `HTTP/1.1 200 OK`) übermittelt. Die Zeilen des Headers selbst sind Schlüssel-Wert-Paare, getrennt durch Doppelpunkte (z. B. `Content-type: text/html`). Die Namen sind durch verschiedene Standards fest spezifiziert. Die Zeilenenden werden durch die Zeichenkombination `<CR><LF>` (carriage return, line feed) markiert, das Ende des Headers wird durch eine Leerzeile signalisiert, was der Übermittlung von `<CR><LF><CR><LF>` gleicht.

Die meisten Headerfelder werden durch RFCs der IETF standardisiert, z. B. der „Kern“ in RFC 2616 und Erweiterungen in RFC 4229. Die in diesen Spezifikationen getroffenen Standards müssen in allen HTTP-Implementierungen vorhanden sein. Zusätzlich können Hersteller oder Projekte zusätzliche Erweiterungen in ihre Software einbauen (für die dann allerdings keine Garantie besteht, dass sie von allen Implementierungen korrekt „verstanden“ werden). Je nach Produkt kann auch der einzelne Anwender oder Administrator eigene Headerfelder definieren.[1][2]

Da im HTTP-Antwort-Header auch unter Umständen sicherheitskritische Informationen wie beispielsweise der verwendete Webserver inklusive Version ersichtlich sind (z. B. `Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)`), wird empfohlen, diese zu verbergen.[3][4]

Einsehen lassen sich die dauerhaften und provisorischen Headerfelder bei der Internet Assigned Numbers Authority (IANA).

Inhaltsverzeichnis

1 **Anfrage-Headerfelder**
2 **Antwort-Headerfelder**
3 **Allgemeine, nicht-standardisierte Felder**
  3.1 Anfrage-Felder
  3.2 Antwort-Felder
4 **Siehe auch**
5 **Einzelnachweise**

Anfrage-Headerfelder

Die Anfrage-Felder kommen im Header der Anfrage eines HTTP-Clients (z. B. Browsers) an einen Webserver vor. Sie beinhalten z. B. Informationen über die angeforderte Ressource und die vom Client angenommenen MIME-Typen.

Für exakte Nachforschungen sei die Lektüre von RFC 2616, Kapitel 14 (S. 62 ff, PDF; 551 kB) empfohlen (Kapitelnummer in der zweiten Spalte der Tabelle).

Name des Felds	Kapitel in RFC 2616	Beschreibung	Beispiel
Accept	14.1	Welche Inhaltstypen der Client verarbeiten kann. Ist es dem Server nicht möglich, einen Inhaltstyp bereitzustellen, der vom Client akzeptiert wird, kann er entweder den HTTP-Statuscode `406 Not acceptable` senden oder einen beliebigen Inhaltstyp zum Kodieren der angeforderten Informationen verwenden.[5] Fehlt das Accept-Feld, so bedeutet dies, dass der Client alle Inhaltstypen akzeptiert. Kann der Server in diesem Beispiel den Inhalt der angeforderten Ressource sowohl als HTML als auch als Bild im GIF-Format an den Client senden, führt der Accept-Header der Anfrage dazu, dass als Inhaltstyp der Antwort HTML gewählt wird.	`Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8`
Accept-Charset	14.2	Welche Zeichensätze der Client anzeigen kann und somit empfangen möchte. Die passende Datei wird über Content Negotiation (z. B. bei Apache `mod_negotiation`) herausgesucht.	`Accept-Charset: utf-8`
Accept-Encoding	14.3	Welche komprimierten Formate der Client unterstützt. Über Content Negotiation wird eine passend komprimierte Datei ausgeliefert.	`Accept-Encoding: gzip,deflate`
Accept-Language	14.4	Welche Sprachen der Client akzeptiert. Falls der Server passend eingerichtet ist und die Sprachversionen vorhanden sind, wird über Content Negotiation die passende Datei ausgeliefert.	`Accept-Language: en-US`
Authorization	14.8	Authentifizierungsdaten für HTTP-Authentifizierungsverfahren	`Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==`
Cache-Control	14.9	Wird genutzt, um Optionen festzulegen, denen durch alle Caching-Mechanismen entlang der Anfrage-/Antwort-Kette Folge geleistet werden muss.	`Cache-Control: no-cache`
Connection	14.10	Welchen Typ von Verbindung der Client bevorzugt.	`Connection: close`
Cookie		ein HTTP-Cookie, das zuvor vom Server mit `Set-Cookie` gesetzt wurde	`Cookie: $Version=1; Skin=new;`
Content-Length	14.13	Länge des Bodys in Bytes	`Content-Length: 348`
Content-MD5	14.15	Eine Base64-codierte MD5-Checksumme des Bodys	`Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ==`
Content-Type	14.17	MIME-Typ des Bodys (hier genutzt für POST- und PUT-Operationen)	`Content-Type: application/x-www-form-urlencoded`
Date	14.18	Datum und Zeit zum Sendezeitpunkt der Anfrage	`Date: Tue, 15 Nov 1994 08:12:31 GMT`
Expect	14.20	Zeigt, welches Verhalten der Client vom Server erwartet. Falls der Server diesen Header nicht versteht oder das Verhalten nicht erfüllen kann, muss er den Code `417 Expectation Failed` senden. Der Client sendet ein `Expect: 100-continue`, wenn er nur den Header, aber nicht den Body einer (sehr großen) Anfrage sendet und daraufhin den HTTP-Statuscode `100 Continue` als Bestätigung erwartet, um eine evtl. sehr große Anfrage schicken zu können. Zweck ist hierbei sicherzugehen, dass der Server die (sehr große) Anfrage annehmen wird.	`Expect: 100-continue`

From	14.22	E-Mail-Adresse des Nutzers, der die Anfrage stellte (heute unüblich). RFC 2616 sagt hierzu, dass der `From:` *nicht* ohne ausdrückliche Genehmigung des Nutzers gesendet werden darf.	`From: user@example.com`
Host	14.23	Domain-Name des Servers, zwingend vorgeschrieben seit HTTP/1.1 und nötig für namensbasierte Hosts Bei Fehlen dieses Headers muss der Server nach Definition mit `400 Bad Request` antworten.	`Host: en.wikipedia.org`
If-Match	14.24	Aktion nur durchführen, falls der gesendete Code mit dem auf dem Server vorhandenen Code übereinstimmt.	`If-Match: "737060cd8c284d8af7ad3082f209582d"`
If-Modified-Since	14.25	Erlaubt dem Server, den Statuscode `304 Not Modified` zu senden, falls sich seit dem angegebenen Zeitpunkt nichts verändert hat.	`If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT`
If-None-Match	14.26	Erlaubt dem Server bei unverändertem Inhalt (verifiziert durch ETags) den Statuscode `304 Not Modified` als Antwort, siehe HTTP ETag	`If-None-Match: "737060cd8c284d8af7ad3082f209582d"`
If-Range	14.27	Falls der Client einen Teil einer Datei vom Server im Cache liegen hat, die sich auf dem Server nicht verändert hat, nur den fehlenden Rest senden; ansonsten ganze Datei schicken.	`If-Range: "737060cd8c284d8af7ad3082f209582d"`
If-Unmodified-Since	14.28	Nur dann die Seite senden, falls diese seit dem angegebenen Zeitpunkt nicht geändert wurde. Wurde die Seite geändert, so sendet der Server den Statuscode `412 Precondition Failed`, bei unveränderter Seite unterscheidet sich die Antwort nicht von einer normalen Antwort und der Client erhält einen 2xx-Statuscode (`Success`).	`If-Unmodified-Since: Sat, 29 Oct 1994 19:43:31 GMT`
Max-Forwards	14.31	Begrenzt die Anzahl der möglichen Weiterleitungen durch Proxys oder Gateways. Das Feld enthält die verbleibende Anzahl an Weiterleitungen, somit muss jeder Proxy diese Zahl aktualisieren (dekrementieren)	`Max-Forwards: 10`
Pragma	14.32	Das Feld Pragma enthält Optionen, die möglicherweise nur von einigen Implementationen verstanden werden und sich an alle Glieder in der Frage-Antwort-Kette richten.	`Pragma: no-cache`
Proxy-Authorization	14.34	Im Feld `Proxy-Authorization` können Autorisierungsdaten für Proxys mit Autorisierungszwang eingebettet werden.	`Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==`
Range	14.35	Enthält eine Bereichsangabe für den Bereich, den der Client vom Server anfordert (in diesem Beispiel nur die Bytes 500-999)	`Range: bytes=500-999`
Referer[sic]	14.36	Im Feld Referer ist der URI der verweisenden Seite enthalten. Klickt man also auf der Hauptseite der deutschsprachigen Wikipedia einen Link an, so sendet der Browser dem Server der aufgerufenen Seite ein Headerfeld wie im Beispiel. (Das Wort „Referer“ ist, sowohl im RFC als auch in den meisten Implementationen falsch geschrieben; richtig wäre „Referrer“ (von *to refer, referred, referred*))	`Referer: http://de.wikipedia.org/wiki/Wikipedia:Hauptseite`
TE	14.39	Welche Formate der Client annehmen kann, möglich sind hier z. B. gzip oder deflate. „trailers“ gibt hier an, dass der Client das Feld	`TE: trailers, deflate`

		„Trailer“ in den einzelnen Stücken beim Encoding-Modus „Chunked“ akzeptiert und auswertet. (Siehe hierzu Kapitel 3.6, 3.6.1, 14.39, 14.40 in RFC 2616)	
Transfer-Encoding	14.41	Die Transformationen, die angewendet wurden, um den Inhalt sicher zum Server zu transportieren. Zurzeit sind folgende Methoden definiert: chunked (aufgeteilt), compress (komprimiert), deflate (komprimiert), gzip (komprimiert), identity.	`Transfer-Encoding: chunked`
Upgrade	14.42	Vorschlag an den Server, ein anderes Protokoll zu nutzen	`Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11`
User-Agent	14.43	Der User-Agent-String des Clients. In ihm stehen Informationen über den Client, sodass z. B. ein serverseitiges Skript an verschiedene Browser angepasste Inhalte ausliefern kann (z. B. bei Downloadseiten, bei denen für Mac OS andere Links angeboten werden sollen als für Microsoft Windows)	`User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)`
Via	14.45	Gibt dem Server Informationen über Proxys im Übertragungsweg.	`Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)`
Warning	14.46	Allgemeine Warnungen über den Umgang mit dem Body oder den Body selbst.	`Warning: 199 Miscellaneous warning`

Antwort-Headerfelder

Headerfeld	Kapitel in RFC 2616	Beschreibung	Beispiel
Accept-Ranges	14.5	Welche Einheiten für Range-Angaben der Server akzeptiert.	`Accept-Ranges: bytes`
Age	14.6	Wie lange das Objekt im Proxy-Cache gelegen hat (in sec).	`Age: 12`
Allow	14.7	Erlaubte Aktionen für eine bestimmte Ressource. Muss u. a. mit einem `405 Method Not Allowed` gesendet werden.	`Allow: GET, HEAD`
Cache-Control	14.9	Teilt allen Caching-Mechanismen entlang der Abrufkette (z. B. Proxys) mit, ob und wie lange das Objekt gespeichert werden darf (in sec).	`Cache-Control: max-age=3600`
Connection	14.10	bevorzugte Verbindungsarten	`Connection: close`
Content-Encoding	14.11	Codierung des Inhalts	`Content-Encoding: gzip`
Content-Language	14.12	Die Sprache, in der die Datei vorliegt (nur sinnvoll bei Content-Negotiation). Wird gesendet, falls der Server mittels Content Negotiation entweder eine Sprache erkannt und ausliefert oder wenn der Server anhand der Endung eine Sprache erkennt.	`Content-Language: de`
Content-Length	14.13	Länge des Body in Bytes	`Content-Length: 348`
Content-Location	14.14	Alternativer Name/Speicherplatz für das angeforderte Element. Wird mittels CN beispielsweise „foo.html“ angefordert, und der Server schickt aufgrund des `Accept-language`-Felds die deutsche Version, die eigentlich unter foo.html.de liegt, zurück, so wird in `Content-Location` der Name der Originaldatei geschrieben.	`Content-Location: /foo.html.de`
Content-MD5	14.15	Die Base64-codierte MD5-Checksumme des Body	`Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ==`
Content-Disposition	19.5.1[1)]	Mit diesem nicht standardisierten und als gefährlich eingestuften Feld kann der Server für bestimmte MIME-Typen Downloadfenster erzeugen und einen Dateinamen vorschlagen.	`Content-Disposition: attachment; filename=fname.ext` `Content-Disposition: inline; filename="picture name.png"`
Content-Range	14.16	Welchen Bereich des Gesamtbodys der gesendete Inhalt abdeckt.	`Content-Range: bytes 21010-47021/47022`
Content-Security-Policy	W3C CSP 1.0	Sicherheitskonzept, um Cross-Site-Scripting (XSS) und ähnliche Angriffe abzuwehren.	`Content-Security-Policy: default-src https://cdn.example.net; frame-src 'none'; object-src 'none'`
Content-Type	14.17	Der MIME-Typ der angeforderten Datei. Er kann nicht mit einer Charset Angabe im HTML header überschrieben werden.	`Content-Type: text/html; charset=utf-8`
Date	14.18	Zeitpunkt des Absendens	`Date: Tue, 15 Nov 1994 08:12:31 GMT`
ETag	14.19	Eine bestimmte Version einer Datei, oft als Message Digest realisiert.	`ETag: "737060cd8c284d8af7ad3082f209582d"`
Expires	14.21	Ab wann die Datei als veraltet angesehen werden kann.	`Expires: Thu, 01 Dec 1994 16:00:00 GMT`
Last-Modified	14.29	Zeitpunkt der letzten Änderung an der Datei (als RFC 2822).	`Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT`
Link	RFC 5988 Abschn. 5	Wird benutzt, um dem Client „verwandte“ Dateien oder Ressourcen mitzuteilen, z. B. einen RSS-Feed, einen Favicon, Copyright-Lizenzen etc. Dieses Header-Feld ist äquivalent zum `<link />`-Feld in (X)HTML.[6]	`Link: </feed>; rel="alternate"`
Location	14.30	Oft genutzt, um Clients weiterzuleiten (mit einem 3xx-Code).	`Location: http://www.w3.org/pub/WWW/People.html`
P3P	–	Dieses Feld wird genutzt, um eine P3P-Datenschutz-Policy wie folgt mitzuteilen: `P3P:CP="your_compact_policy"`. P3P setzte sich nicht richtig durch,[7] wird jedoch von einigen Browsern und Webseiten genutzt, um z. B. Cookie-Richtlinien durchzusetzen oder zu überprüfen.	`P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=151657 for more info."`
Pragma	14.32	Implementierungs-spezifische Optionen, die mehrere Stationen in der Request-Response-Kette beeinflussen können.	`Pragma: no-cache`

Proxy-Authenticate	14.33	Anweisung, ob und wie der Client sich beim Proxy zu authentifizieren hat.	`Proxy-Authenticate: Basic`
Refresh	Proprietär	Refresh wird genutzt, um nach einer bestimmten Zahl von Sekunden weiterzuleiten oder die Seite zu aktualisieren. Dieses Headerfeld ist proprietär und kommt von Netscape, wird aber von den meisten Browsern unterstützt.	`Refresh: 5; url=http://www.w3.org/pub/WWW/People.html`
Retry-After	14.37	Falls eine Ressource zeitweise nicht verfügbar ist, so teilt der Server dem Client mit diesem Feld mit, wann sich ein neuer Versuch lohnt.	`Retry-After: 120`
Server	14.38	Serverkennung (so wie `User-Agent` für den Client ist, ist `Server` für die Serversoftware).	`Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)`
Set-Cookie	–	Ein Cookie	`Set-Cookie: UserID=FooBar; Max-Age=3600; Version=1`
Trailer	14.40	Das Trailer-Feld enthält die Namen der Headerfelder, die im Trailer der Antwort (bei *Chunked*-Encoding) enthalten sind. Eine Nachricht in *Chunked*-Encoding ist aufgeteilt in den Header (Kopf), den Rumpf (Body) und den Trailer, wobei der Rumpf aus Effizienzgründen in Teile (Chunks) aufgeteilt sein kann. Der Trailer kann dann (je nach Wert des TE-Felders der Anfrage) Header-Informationen beinhalten, deren Vorabberechnung der Effizienzsteigerung zuwiderläuft.	`Trailer: Max-Forwards`
Transfer-Encoding	14.41	Die Methode, die genutzt wird, den Inhalt sicher zum Nutzer zu bringen. Zurzeit sind folgende Methoden definiert: chunked (aufgeteilt), compress (komprimiert), deflate (komprimiert), gzip (komprimiert), identity	`Transfer-Encoding: chunked`
Vary	14.44	Zeigt Downstream-Proxys, wie sie anhand der Headerfelder zukünftige Anfragen behandeln sollen, also ob die gecachte Antwort genutzt werden kann oder eine neue Anfrage gestellt werden soll.	`Vary: *`
Via	14.45	Informiert den Client, über welche Proxys die Antwort gesendet wurde.	`Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)`
Warning	14.46	Eine allgemeine Warnung vor Problemen mit dem Body.	`Warning: 199 Miscellaneous warning`
WWW-Authenticate	14.47	Definiert die Authentifikationsmethode, die genutzt werden soll, um eine bestimmte Datei herunterzuladen (Genauer definiert in RFC 2617).	`WWW-Authenticate: Basic`

[1] Nicht im offiziellen HTTP/1.1-Standard, da (Kapitel 15.5) eine Reihe von Sicherheitsbedenken geäußert wurden. `Content-disposition` ist in RFC 2183 genauer beschrieben.

Allgemeine, nicht-standardisierte Felder

Anfrage-Felder

Nicht-standardisierte Header tragen oft ein `X-` am Anfang. Mit RFC 6648 gilt das Präfix `X-` als veraltet.

Feldname	Beschreibung	Beispiel
X-Requested-With[8]	Oft genutzt bei Ajax.	`X-Requested-With: XMLHttpRequest`
X-Do-Not-Track[9]	Befiehlt einer Website, das Verfolgen (*Tracken*) des Nutzers zu deaktivieren. Bis jetzt wird dieses Feld von den allermeisten Servern ignoriert. Dies könnte sich jedoch in der Zukunft noch ändern. Siehe auch das Feld 'DNT'.	`X-Do-Not-Track: 1`
DNT[10] oder Dnt	Befiehlt einer Website, den Nutzer nicht zu tracken. Dieses Feld ist Mozillas Version des `X-Do-Not-Track`-Feldes, wird aber auch von Safari 5, Internet Explorer 9 und Google Chrome (letzterer verwendet die Variante `Dnt`) unterstützt.[11] Am 7. März 2011 wurde ein Entwurf bei der IETF eingereicht.[12]	`DNT: 1`
X-Forwarded-For[13]	ein De-facto-Standard zur Identifizierung der ursprünglichen IP-Adresse eines Clients, der sich mit einem Webserver über einen HTTP-Proxy oder Lastverteiler verbindet.	`X-Forwarded-For: client1, proxy1, proxy2`
X-Forwarded-Proto[14]	ein De-facto-Standard zur Identifizierung des ursprünglichen Protokolls einer HTTP-Anforderung, da ein Reverse-Proxy (Lastverteiler) mit einem Webserver über HTTP kommuniziert.	`X-Forwarded-Proto: https`

Antwort-Felder

Feldname	Beschreibung	Beispiel
X-Frame-Options[15]	Clickjacking-Schutz: „DENY“ – kein Rendering in einem Frame; „SAMEORIGIN“ – Nur dann kein Rendering, falls die Herkunft falsch ist.	`X-Frame-Options: DENY`
X-XSS-Protection[16]	Filter für Cross-Site-Scripting (XSS)	`X-XSS-Protection: 1; mode=block`
X-Content-Type-Options[17]	Der einzige definierte Wert „nosniff“ untersagt dem Internet Explorer durch MIME-Sniffing einen anderen als den deklarierten Inhaltstyp zu bestimmen und anzuwenden.	`X-Content-Type-Options: nosniff`
X-Powered-By[18]	Gibt an, auf welcher Technologie (ASP.NET, PHP, JBoss, u.a.) die Webapplikation basiert (Details zur Version finden sich oft in `X-Runtime`, `X-Version`, oder `X-AspNet-Version`)	`X-Powered-By: PHP/5.3.8`

X-UA-Compatible[19]	Empfiehlt die empfohlene Render-Engine (oft ein abwärtskompatibler Modus) um den Inhalt anzuzeigen. Auch genutzt um den Chrome Frame im Internet Explorer zu aktivieren.	`X-UA-Compatible: IE=EmulateIE7`
		`X-UA-Compatible: IE=edge`
		`X-UA-Compatible: Chrome=1`

X-Robots-Tag[20]	Legt für Webcrawler fest, welche Inhalte indexiert werden dürfen.	`X-Robots-Tag: noarchive`
		`X-Robots-Tag: unavailable_after: 25 Jun 2010 15:00:00 PST`
		`X-Robots-Tag: googlebot: nofollow`

Siehe auch

- Hypertext Transfer Protocol (HTTP)
- HTTP-Statuscode
- HTTP-Cookie
- HTTP ETag

Einzelnachweise

1. Anleitung, um mit Apache2 eigene Headerfelder zu definieren (http://lbo.spheniscida.de/doc/apache2/defining-own-headers.txt)
2. Dokumentation der Direktive `header` (http://httpd.apache.org/docs/current/mod/mod_headers.html#header)
3. *Ubuntu: Apache-Informationen „verstecken“* (http://johann.gr/ubuntu-apache-informationen-verstecken/). In: *Johann.gr.* 23. Dezember 2015, abgerufen am 20. Juni 2016.
4. *How to hide Nginx version | Nginx Tips.* (http://www.scalescale.com/tips/nginx/how-to-hide-nginx-version/) In: *ScaleScale.com.* Abgerufen am 20. Juni 2016 (amerikanisches englisch).
5. RFC 2616, Abschnitt 10.4.7 (https://tools.ietf.org/html/rfc2616#section-10.4.7)
6. RFC 5988 Abschn. 5 (https://www.ietf.org/rfc/rfc5988.txt.pdf) (PDF; 36 kB)
7. W3C P3P Work Suspended (http://www.w3.org/P3P)
8. docs.djangoproject.com (http://docs.djangoproject.com/en/1.2/ref/contrib/csrf/)
9. hackademix.net (http://hackademix.net/2010/12/28/x-do-not-track-support-in-noscript/)
10. blog.sidstamm.com (http://blog.sidstamm.com/2011/01/try-out-do-not-track-http-header.html)
11. blogs.msdn.com (http://blogs.msdn.com/b/ie/archive/2011/03/14/web-tracking-protection-minimum-standards-and-opportunities-to-innovate.aspx)
12. IETF Do Not Track: A Universal Third-Party Web Tracking Opt Out (http://tools.ietf.org/html/draft-mayer-do-not-track-00)
13. Amos Jeffries: *SquidFaq/ConfiguringSquid – Squid Web Proxy Wiki (http://wiki.squid-cache.org/SquidFaq/ConfiguringSquid#head-3518b69c63e221cc3cd7885415e365ffaf3dd27f)* 2. Juli 2010. Abgerufen am 10. September 2009.
14. Dave Steinberg: *How do I adjust my SSL site to work with GeekISP's loadbalancer? (http://www.geekisp.com/faq/6_65_en.html)* 10. April 2007. Abgerufen am 30. September 2010.
15. blogs.msdn.com (http://blogs.msdn.com/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx)
16. Eric Lawrence: *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)*. 2. Juli 2008. Abgerufen am 30. September 2010.
17. Eric Lawrence: *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)* 3. September 2008. Abgerufen am 28. September 2010.
18. *Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow (http://stackoverflow.com/questions/1288338/why-does-asp-net-framework-add-the-x-powered-by-asp-net-http-header-in-response)*. Abgerufen am 30. September 2010.
19. *Definiere Dokument Kompatibilität: Spezifiziere Dokument Kompatibilität Modus (http://msdn.microsoft.com/en-us/library/ie/cc288325%28v=vs.85%29.aspx#SetMode)*. 1. April 2011. Abgerufen am 24. Januar 2012.
20. developers.google.com (https://developers.google.com/webmasters/control-crawl-index/docs/robots_meta_tag?hl=de) vom 18. September 2013.

Abgerufen von „https://de.wikipedia.org/w/index.php?title=Liste_der_HTTP-Headerfelder&oldid=161856116“

Diese Seite wurde zuletzt am 21. Januar 2017 um 19:45 Uhr bearbeitet.

Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.
Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.