

HTTP
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

M. Nottingham
Fastly
November 4, 2019

The Cache-Status HTTP Response Header draft-ietf-httpbis-cache-header-02

Abstract

To aid debugging, HTTP caches often append headers to a response detailing how they handled the request. This specification codifies that practice and updates it for HTTP's current caching model.

Note to Readers

RFC EDITOR: please remove this section before publication

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/> [1].

Working Group information can be found at <https://httpwg.org/> [2]; source code and issues list for this draft can be found at <https://github.com/httpwg/http-extensions/labels/cache-header> [3].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Notational Conventions	3
2.	The Cache-Status HTTP Response Header	3
2.1.	The fwd parameter	4
2.2.	The fwd-res parameter	4
2.3.	The fwd-stored parameter	5
2.4.	The res-fresh parameter	5
2.5.	The cache-fresh parameter	5
2.6.	The collapse-hit parameter	5
2.7.	The collapse-wait parameter	5
2.8.	The key parameter	6
3.	Examples	6
4.	Security Considerations	7
5.	References	7
5.1.	Normative References	7
5.2.	Informative References	8
5.3.	URIs	8
	Author's Address	8

[1. Introduction](#)

To aid debugging, HTTP caches often append headers to a response detailing how they handled the request.

Unfortunately, the semantics of these headers are often unclear, and both the semantics and syntax used vary greatly between implementations.

This specification defines a single, new HTTP response header field, "Cache-Status" for this purpose.

Nottingham

Expires May 7, 2020

[Page 2]

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

This document uses ABNF as defined in [[RFC5234](#)], along with the "%" extension for case sensitivity defined in [[RFC7405](#)].

2. The Cache-Status HTTP Response Header

The Cache-Status HTTP response header indicates caches' handling of the request corresponding to the response it occurs within.

Its value is a List [[I-D.ietf-httpbis-header-structure](#)]:

```
Cache-Status    = sh-list
```

Each member of the parameterised list represents a cache that has handled the request. The first member of the list represents the cache closest to the origin server, and the last member of the list represents the cache closest to the user agent (possibly including the user agent's cache itself, if it chooses to append a value).

Caches determine when it is appropriate to add the Cache-Status header field to a response. Some might decide to add it to all responses, whereas others might only do so when specifically configured to, or when the request contains a header that activates a debugging mode.

When adding a value to the Cache-Status header field, caches SHOULD preserve the existing contents of the header, to allow debugging of the entire chain of caches handling the request.

The list members identify the cache that inserted the value, and MUST have a type of either sh-string or sh-token. Depending on the deployment, this might be a product or service name (e.g., ExampleCache or "Example CDN"), a hostname ("cache-3.example.com"), and IP address, or a generated string.

Each member of the list can also have a number of parameters that describe that cache's handling of the request. While all of these parameters are OPTIONAL, caches are encouraged to provide as much information as possible.

fwd	= sh-token
fwd-res	= sh-token
fwd-stored	= sh-boolean
res-fresh	= sh-integer
cache-fresh	= sh-integer
collapse-hit	= sh-boolean
collapse-wait	= sh-integer
key	= sh-string

2.1. The fwd parameter

"fwd" indicates why the request went forward. If it is not present, the value defaults to "none".

It can have one of the following values:

- o none - The request did not go forward; i.e., it was a hit, and was served from the cache.
- o bypass - The cache was configured to not handle this request
- o uri-miss - The cache did not contain any responses that matched the request URI
- o vary-miss - The cache contained a response that matched the request URI, but could not select a response based upon this request's headers.
- o miss - The cache did not contain any responses that could be used to satisfy this request (to be used when an implementation cannot distinguish between uri-miss and vary-miss)
- o res-stale - The cache was able to select a response for the request, but it was stale
- o req-stale - The cache was able to select a fresh response for the request, but client request headers (e.g., Cache-Control request directives) did not allow its use

2.2. The fwd-res parameter

"fwd-res" indicates what the result of the forward request was. It is only valid when fwd is "res-stale" or "req-stale", and defaults to "full" if not present when fwd is one of those values.

It can have one of the following values:

- o full - indicates that the response was a complete response (any status code except 304 Not Modified and 206 Partial Response)
- o partial - indicates that the response was a 206 Partial Response
- o notmod - indicates that the response was a 304 Not Modified

2.3. The fwd-stored parameter

"fwd-stored" indicates whether the cache stored the response; a true value indicates that it did. Only valid when fwd is not "none".

2.4. The res-fresh parameter

"res-fresh" indicates the response's remaining freshness lifetime (as per [[I-D.ietf-httpbis-cache](#)], Section 4.2.1), as an integer number of seconds. This does not include freshness assigned by the cache (see "cache-fresh"). May be negative, to indicate staleness.

2.5. The cache-fresh parameter

"cache-fresh" indicates the response's remaining freshness lifetime as calculated by the cache, as an integer number of seconds. This includes freshness assigned by the cache; e.g., through heuristics, local configuration, or other factors. May be negative, to indicate staleness.

If both cache-fresh and res-fresh appear as parameters on the same value, it implies that the cache freshness overrode the response freshness.

2.6. The collapse-hit parameter

"collapse-hit" indicates whether this request was collapsed together with one or more other forward requests; if true, the response was successfully reused; if not, a new request had to be made. If not present, the request was not collapsed with others.

2.7. The collapse-wait parameter

"collapse-wait" indicates the amount of time that the cache held the request while waiting to see if it could be successfully collapsed, as an integer number of milliseconds.

2.8. The key parameter

"key" conveys a representation of the cache key used for the response. Note that this may be implementation-specific.

3. Examples

The most minimal cache hit:

```
Cache-Status: ExampleCache
```

... but a polite cache will give some more information, e.g.:

```
Cache-Status: ExampleCache; res-fresh=376
```

A "negative" hit (i.e., the cache imposed its own freshness lifetime):

```
Cache-Status: ExampleCache; cache-fresh=415
```

A stale hit just has negative freshness:

```
Cache-Status: ExampleCache; res-fresh=-412
```

Whereas a complete miss is:

```
Cache-Status: ExampleCache; fwd=uri-miss
```

A miss that validated on the back-end server:

```
Cache-Status: ExampleCache; fwd=res-stale; fwd-res=notmod
```

A miss that was collapsed with another request:

```
Cache-Status: ExampleCache; fwd=uri-miss; collapse-hit=?1
```

A miss that the cache attempted to collapse, but couldn't:

```
Cache-Status: ExampleCache; fwd=uri-miss;  
               collapse-hit=?0; collapse-wait=240
```

Going through two layers of caching, both of which were hits, and the second collapsed with other requests:

```
Cache-Status: "CDN Company Here"; res-fresh=545,  
               OriginCache; cache-fresh=1100; collapse-hit=?1
```


4. Security Considerations

Information about a cache's content can be used to infer the activity of those using it. Generally, access to sensitive information in a cache is limited to those who are authorised to access that information (using a variety of techniques), so this does not represent an attack vector in the general sense.

However, if the Cache-Status header is exposed to parties who are not authorised to obtain the response it occurs within, it could expose information about that data.

For example, if an attacker were able to obtain the Cache-Status header from a response containing sensitive information and access were limited to one person (or limited set of people), they could determine whether that information had been accessed before. This is similar to the information exposed by various timing attacks, but is arguably more reliable, since the cache is directly reporting its state.

Mitigations include use of encryption (e.g., TLS [[RFC8446](#)])) to protect the response, and careful controls over access to response headers (as are present in the Web platform). When in doubt, the Cache-Status header field can be omitted.

5. References

5.1. Normative References

[I-D.ietf-httpbis-cache]

Fielding, R., Nottingham, M., and J. Reschke, "HTTP Caching", [draft-ietf-httpbis-cache-05](#) (work in progress), July 2019.

[I-D.ietf-httpbis-header-structure]

Nottingham, M. and P. Kamp, "Structured Headers for HTTP", [draft-ietf-httpbis-header-structure-13](#) (work in progress), August 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.

- [RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", [RFC 7405](#), DOI 10.17487/RFC7405, December 2014, <<https://www.rfc-editor.org/info/rfc7405>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[5.2.](#) Informative References

- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

[5.3.](#) URIs

- [1] <https://lists.w3.org/Archives/Public/ietf-http-wg/>
- [2] <https://httpwg.org/>
- [3] <https://github.com/httpwg/http-extensions/labels/cache-header>

Author's Address

Mark Nottingham
Fastly

Email: mnot@mnot.net
URI: <https://www.mnot.net/>

